Topics for Today

- Text transformation
  - Word occurrence statistics
  - Tokenizing
  - Stopping and stemming
  - Phrases
  - Document structure
  - Link analysis
  - **Information extraction**
  - **Internationalization**

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"; "history book"; "deciduous trees"
  - Less ambiguous
    - » e.g., "big apple" vs. "apple"
- Can be difficult to incorporate into 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 issues
  - Should phrases be indexed?
  - 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

POS Tagging

- POS taggers use 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

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: (a very old POS 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 a Zipf distribution than words alone
- Could index all n-grams up to specified length
  - Much faster than POS tagging
  - Uses a lot of storage
    - » e.g., document containing 1,000 words would contain 3,990 instances of word n-grams of length $2 \le n \le 5$

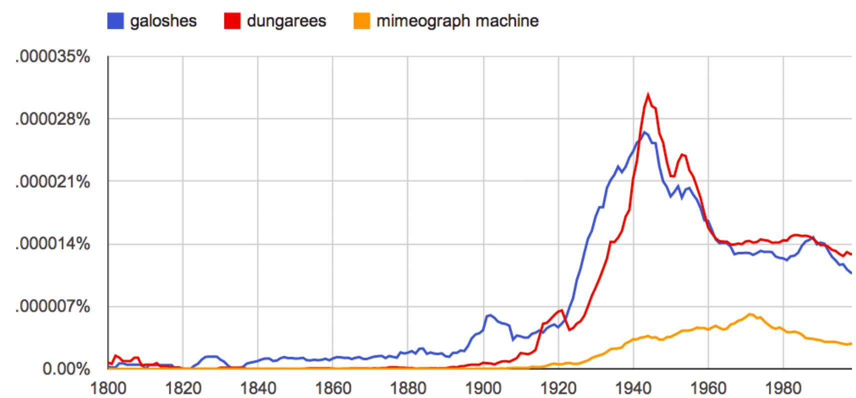
Google N-Grams

- Web search engines index n-grams
- Google n-gram 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”

Google books n-gram corpus



Topics for Today

- Text transformation
  – Word occurrence statistics
  – Tokenizing
  – Stopping and stemming
  – Phrases
  – Document structure
  – Link analysis
  – Information extraction
  – Internationalization

Information Extraction

- Automatically extract structure from text
  – annotate document using tags to identify extracted structure
- ***Named entity recognition***
  – identify phrases that refer to something of interest in a particular application
  – e.g., people, companies, locations, dates, product names, prices, etc.

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

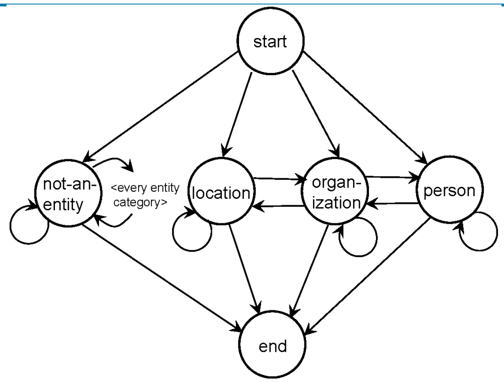
- Rules either developed manually by trial and error or using machine learning techniques
- **Statistical**
  - use a probabilistic model of the words in and around an entity
  - probabilities estimated using ***training data*** (manually annotated text)
  - Hidden Markov Model (HMM) is one approach

Covered in CS4740 NLP

HMM for Extraction

- Resolve ambiguity in a word using *context*
  - e.g., "marathon" is a location or a sporting event, "boston marathon" is a specific sporting event
- Model context using a *generative* model of the sequence of words
  - *Markov property*: the next word in a sequence depends only on a small number of the previous words

Learn a HMM from training data



- Each state is associated with a probability distribution over words (the output)

HMM for Extraction

- To recognize named entities, find sequence of “labels” that give highest probability for the sentence
  - only the outputs (words) are visible or observed
  - states are “hidden”
  - e.g., <start><name><not-an-entity><location><not-an-entity><end>
- *Viterbi* algorithm used for recognition

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

NE recognition

- Not generally found to be helpful during search
- Useful for domain-specific or vertical search engines
- Useful for displaying results, browsing
- Critical for question answering applications

Internationalization

- 2/3 of the Web is in English
- About 50% of Web users do not use English as their primary language
- 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*: search in multiple languages at the same time

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

Chinese “Tokenizing”

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