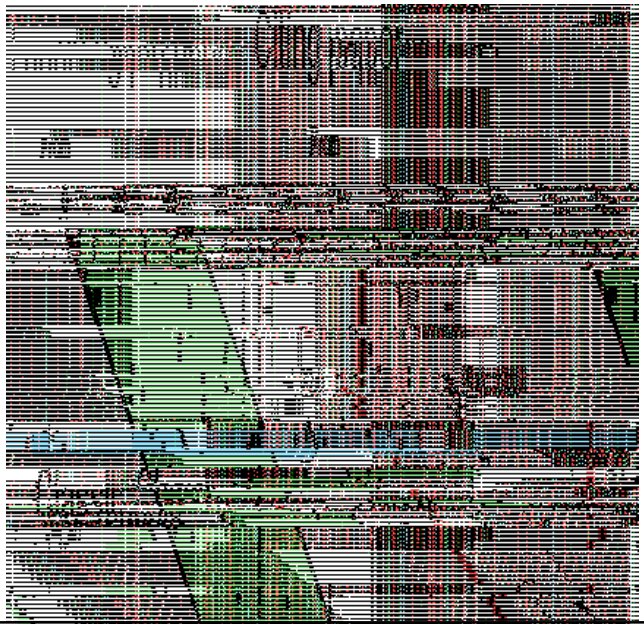
Outline

- search engine basics
- bibliometric and related methods
- information extraction methods
- relevant WWW resources
- emerging issues

Bibliometric Methods

- the methods we have discussed so far do not take into consideration *relationships among documents*, such as:
  - citations in a bibliography
  - hyperlinks
- *bibliometric methods* exploit the structure of such links

Bibliometrics Illustrated

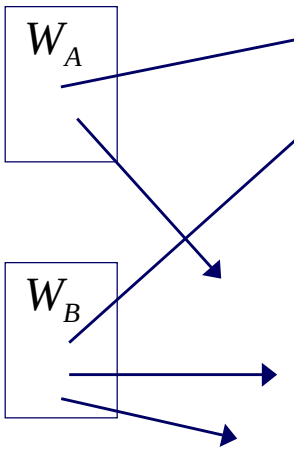


Google™ and the PageRank™ Algorithm

- the Google search engine uses
  - Boolean methods to determine relevant pages
  - * bibliometric methods to rank these relevant pages
- net result: top ranking pages tend to be authoritative sources
- http://www.google.com

The PageRank Idea

- the "weight" of a page is determined by the weights of pages that link to it



$$W_C \propto \frac{W_A}{2} + \frac{W_B}{3}$$

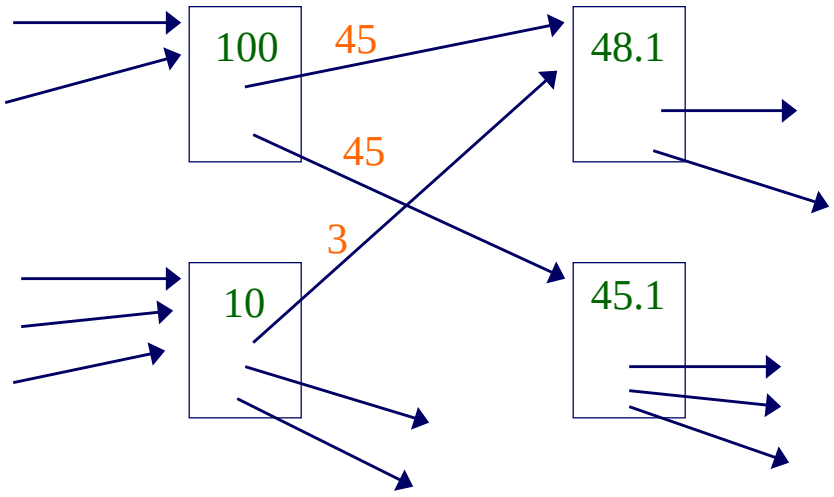
The PageRank Algorithm

- iteratively update weight W_j associated with each page

$$W_j = (1-d) + d \sum_{i=1, i \neq j}^{N} l_{i,j} \frac{W_i}{n_i}$$

d	fraction of weight that gets forwarded
$l_{i,j}$	1 if i links to j, 0 otherwise
n_i	number of links emanating from i

PageRank Example

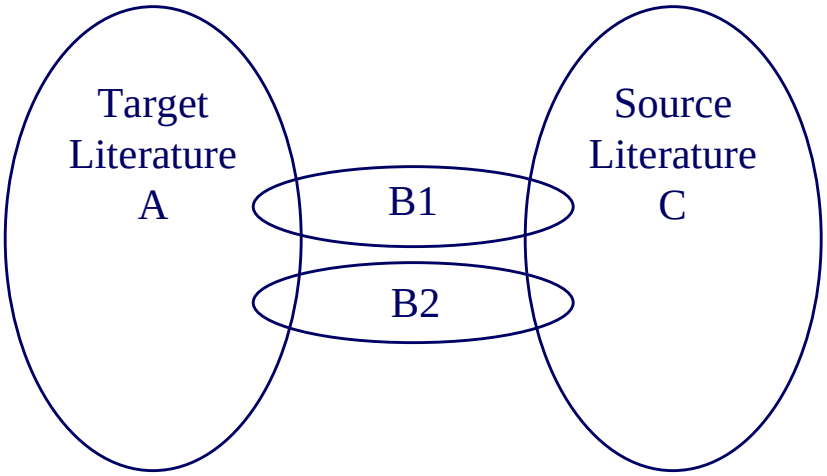


1– =.1

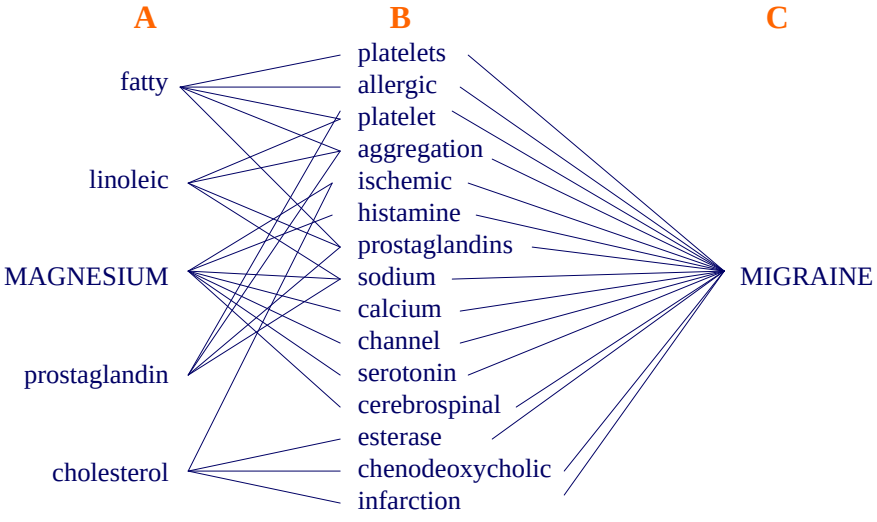
ARROWSMITH: Finding Complementary Literatures

- another sense in which documents can be linked: refer to the same terms/concepts
- ARROWSMITH aids in identifying relationships that are implicit, but not explicitly described, in the literature
- http://kiwi.uchicago.edu/
- Swanson & Smalheiser, *Artificial Intelligence* 91, 1997

ARROWSMITH: Finding Complementary Literatures



ARROWSMITH Example: The Magnesium-Migraine Link



The ARROWSMITH Method

- given: query concept C (e.g. *migraine*)
- do:
  - run MEDLINE search on C
  - derive a set of words (B) from titles of returned articles; retain words
  - run MEDLINE search on each B word to assemble list of A words
  - rank A-C linkages by number of different intermediate B terms

Restricting the Search in ARROWSMITH

- prune B list by
  - using a predefined stop-list ("clinical", "comparative", "drugs",…)
  - having a human expert filter terms
- prune A list using *category restrictions* (e.g. dietary factors, toxins, etc.)
- prune C-B, B-A linkages by requiring:

$$\Pr(B \mid C) > \Pr(B)$$

$$\Pr(A \mid B) > \Pr(A)$$

ARROWSMITH Case Studies

- indomethacin and Alzheimer's disease
- estrogen and Alzheimer's disease
- phospholipases and sleep
- etc.

Bibliometric Methods: Prospectus

- currently there are good bibliometric-related tools for
  - Web searching (Google)
  - MEDLINE discovery (ARROWSMITH)
- future methods will perhaps
  - integrate diverse sources (MEDLINE, Web pages, sequence databases, etc.)
  - try to characterize the relationships encoded by links

Outline

- search engine basics
- bibliometric and related methods
- information extraction methods
- relevant WWW resources
- emerging issues

Information Extraction

- *information extraction* involves automatically extracting key fragments from documents
- we'll consider three types of IE tasks
  - *named entity recognition*: identify instances of a specified set of classes
  - *keyword extraction*: extract a set of keywords that characterizes a given set of documents
  - *relation extraction*: extract instances of a specified set of relations

Named Entity Recognition

- in addition to controlled vocabularies (e.g. MeSH), it would be useful to have methods for recognizing general classes of terms
- protein names, for example, can be accurately recognized using *morphological, lexical,* and *syntactic* information
- Fukuda et al., *Pacific Symposium on Biocomputing*, 1998

Recognizing Protein Names

- morphological analysis is used to identify "core terms" (e.g. Src, SH3, p54, SAP) and "feature terms" (e.g. receptor, protein)

  The focal adhesion kinase (FAK) is...

- lexical and syntactic analysis is used to extend terms into protein names

  The focal adhesion kinase (FAK) is...

Recognizing Protein Names: Morphological Analysis

- make list of candidate terms: words that include upper-case letters, digits, and non-alphanumeric characters
- exclude words with length > 9 consisting of lower-case letters and -'s (e.g. full-length)
- exclude words that indicate units (e.g. aa, bp, nM)
- exclude words that are composed mostly of non-alphanumeric characters (e.g. +/-)

Recognizing Protein Names: Lexical/Syntactic Analysis

- merge adjacent terms

  Src SH3 domain → Src SH3 domain

- merge non-adjacent terms separated only by nouns, adjectives and numerals

  Ras guanine nucleotide exchange factor Sos
  →
  Ras guanine nucleotide exchange factor Sos

Recognizing Protein Names: Lexical/Syntactic Analysis

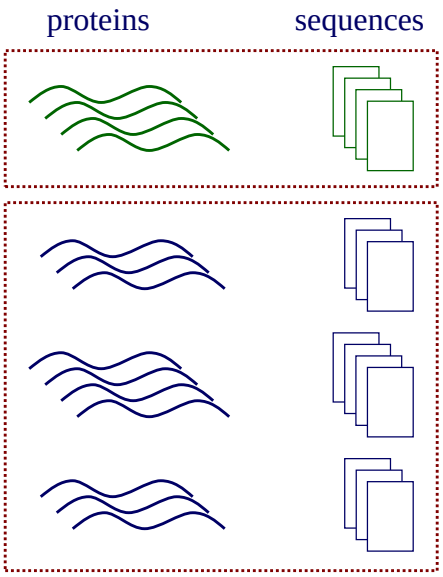
- extend term to include a succeeding upper-case letter or a Greek-letter word

p85 alpha ⟶ p85 alpha

Keyword Extraction

- example: given a protein family, extract keywords from associated abstracts that have a high degree of specificity to family
- can be thought of as statistical annotation
- Andrade & Valencia, *Bioinformatics* 14(7), 1998

Keyword Extraction



find keywords that distinguish the set of query abstracts from the background abstracts

Keyword Extraction: Annotating Protein Families

- to evaluate a candidate keyword for a query family, want to consider
  - how frequently word occurs in abstracts associated with family
  - how frequently word occurs in abstracts for other families on average
  - how much variance there is in word frequency for other families

Keyword Extraction: Calculating Word Frequencies

- for each word and each family, determine the frequency of the word in the family as:

$$= \frac{\text{[illegible]}}{\text{[illegible]}}$$

Keyword Extraction: Word Frequency Statistics

- determine how frequently word *w* occurs in abstracts for families in background on average

$$\overline{\quad} = \frac{\sum_{i=1}}{\quad}$$

Keyword Extraction: Word Frequency Statistics

- determine how much variance is there is in word frequency for families in background

$$\sigma^w = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(F_i^w - \overline{F^w})^2}$$

Keyword Extraction: Ranking Keywords

- a word *w* that has a high *z*-score is a candidate keyword

frequency of *w* in query family

avg. frequency of *w* in background

$$= \frac{\quad - \quad}{\sigma}$$

deviation of frequency of *w* in background

Keyword Extraction: Example

- keywords have z-score > 0.1 and appear in abstracts for > 50% of proteins in family

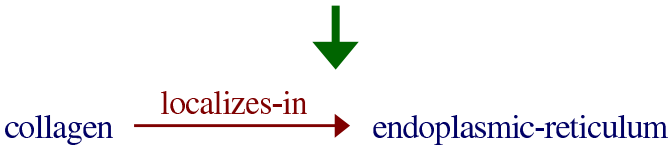
family	SwissProt annotation	extracted keywords
lndk	diphosphate	awd
	ndk	nm23
	ndp	ndp
	nucleoside	diphosphate
	atp	nucleoside
	kinase	drosophila
	transferase	k
		kinase
lppn	proteinase	papaya
	thiol	papain
	protease	thiol
		proteinase
		cysteine

Relation Extraction

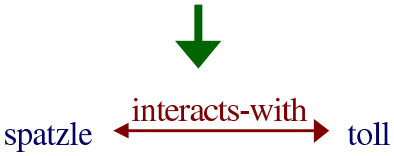
- given predefined relations of interest, extract *instances* of these relations
- example relations
  - subcellular localization of proteins [Craven & Kumlien, ISMB 1999]
  - chromosome locations of genes [Leek, M.S. thesis 1997]
  - protein-protein interactions [Blaschke et al., ISMB 1999; Thomas et al., PSB 2000]

Relation Extraction: Examples

... which plays a pivotal role in during the biosynthesis and secretion of collagen molecules in the endoplasmic reticulum.



…where they lead to the localized activation of a serine protease cascade required to produce the active spatzle ligand to activate the toll receptor.



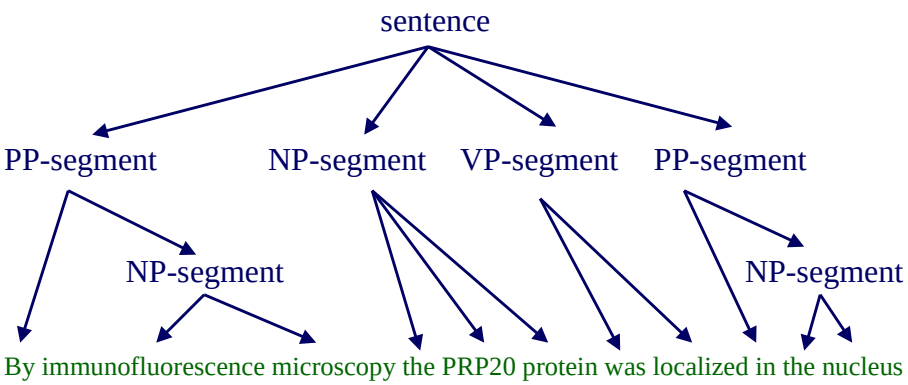
Relation Extraction: Hand-Coded Extractors

- [Blaschke et al., ISMB 1999]
- extract protein-protein interactions by looking for sentence fragments that conform to pattern:
  <protein A> … <action> … <protein B>
- where <action> is verb from specified list:
  acetylat-, activat-, associated with, bind-, destabilize-, inhibit-, interact-, is conjugated to, modulat-, phosphorylat-, regulat-, stabiliz-, suppress-, target

Relation Extraction: Learned Extractors

- [Craven & Kumlien, ISMB 1999]
- use machine-learning methods to induce regularities in sentences expressing a given relation
- approach:
  - parse sentences using a shallow parser
  - learn rules that can characterize positive examples in terms of (i) relationships among phrases (ii) word statistics in phrases

Relation Extraction: A Shallow Parse



Relation Extraction: A Learned Rule (Simplified)

- look for sentences that match the pattern:

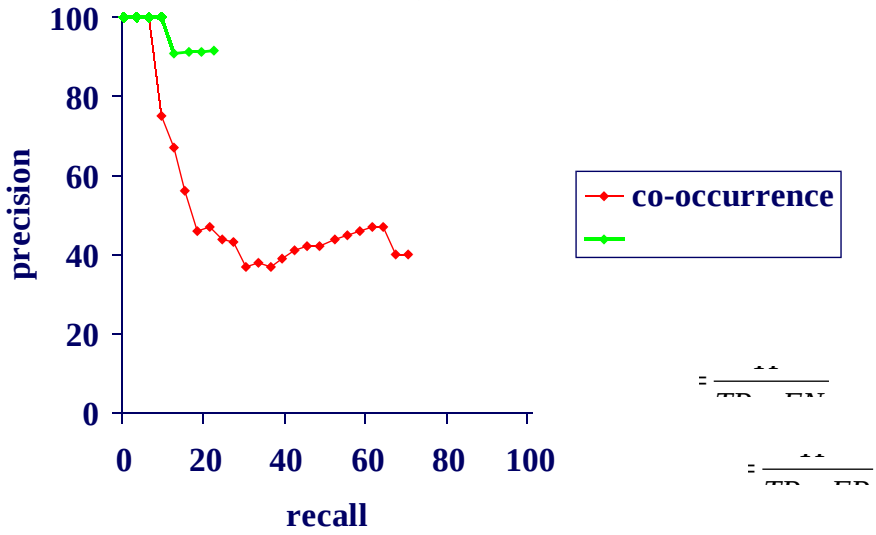
  <protein NP > <VP> <location NP >

- where <protein NP> satisfies a naïve Bayes classifier that highly weights the words: *protein, beta, galactosidas, gene, alpha, mannosidas, bifunct, product, …*
- and <location NP> satisfies a naïve Bayes classifier that highly weights the words: *nucleu, nuclei, mitochondria, vacuol, plasma, insid, membran, in, to, with,...*

Relation Extraction: One Experiment

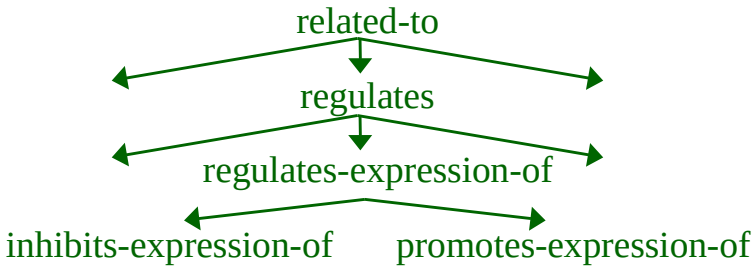
- *training data*: 336 protein-location pairs from Yeast Protein Database and associated abstracts from MEDLINE
- *test data*: 2,889 abstracts characterizing 6 proteins
- *baseline for comparison*: predicting a protein is found in a subcellular location if the protein/location are referenced in the same sentence

Recognizing Sentences about Protein Localization



Relation Extraction

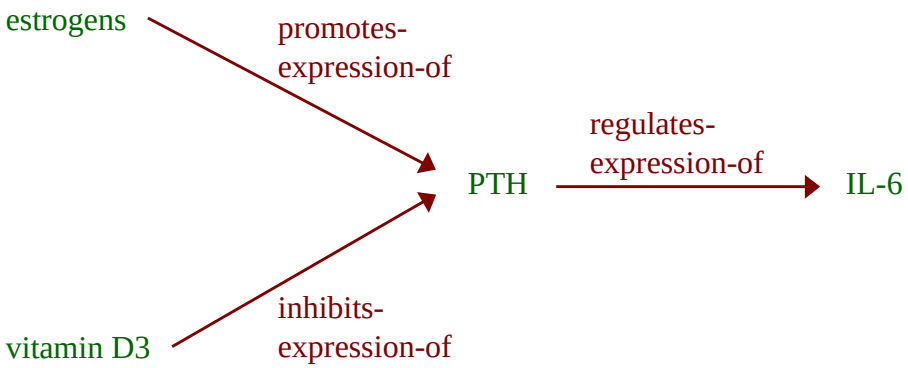
- suppose we have a hierarchy of relations



- even if we can't accurately do extraction for most specific relations, there is value in extracting more abstract relations (consider ARROWSMITH)

Relation Extraction

- Given a collection of extracted instances and some simple background rules, we can do automatic inference (e.g. *what indirectly regulates IL-6?*).



Relation Extraction: Prospectus

- current methods
  - require large amount of human effort to use
  - consider sentences in isolation
  - handle only simple sentences
- best future methods will probably
  - involve both hand-coding and learning
  - take better advantage of controlled vocabularies, ontologies, etc.