Machine Learning Summer School 2003

Information Retrieval and Language Technology

Thorsten Joachims

Cornell University
Computer Science Department
tj@cs.cornell.edu
http://www.joachims.org

Abstract

The course will give an overview of how statistical learning can help organize and access information that is represented in textual form. In particular, it will cover tasks like text classification, information retrieval, information extraction, topic detection, and topic tracking. The course will introduce the basic techniques for representing text and analyze their statistical properties. An emphasis of the course will be on giving an overview of interesting learning problems in this area, providing starting points for future research.

Overview

Part I: Information Retrieval Basics

- boolean retrieval / vector space retrieval and TFIDF weighting
- bag-of-words representation / stemming / stopword removal etc.
- evaluation

Part II: Text Classification and the Statistical Properties of Text

- conventional / large-margin methods for text classification
- relationship between margin and statistical properties of text

Part III: Tasks and Research Areas in Language Technology

- topic detection and tracking
- part-of-speech tagging
- information extraction
- named entity recognition
- learning retrieval functions

Part I:
Information Retrieval Basics

Machine Learning Summer School 2003

Thorsten Joachims

Cornell University

Based on slides from Professor Claire Cardie, Cornell, and Professor Jamie Callan, CMU

Overview: IR Basics

- **Task Definition**
  - IR process
  - Ad-hoc retrieval
- **Retrieval Models**
- **Evaluation**
- **Representation and Indexing**
- **Data Structures and Access**

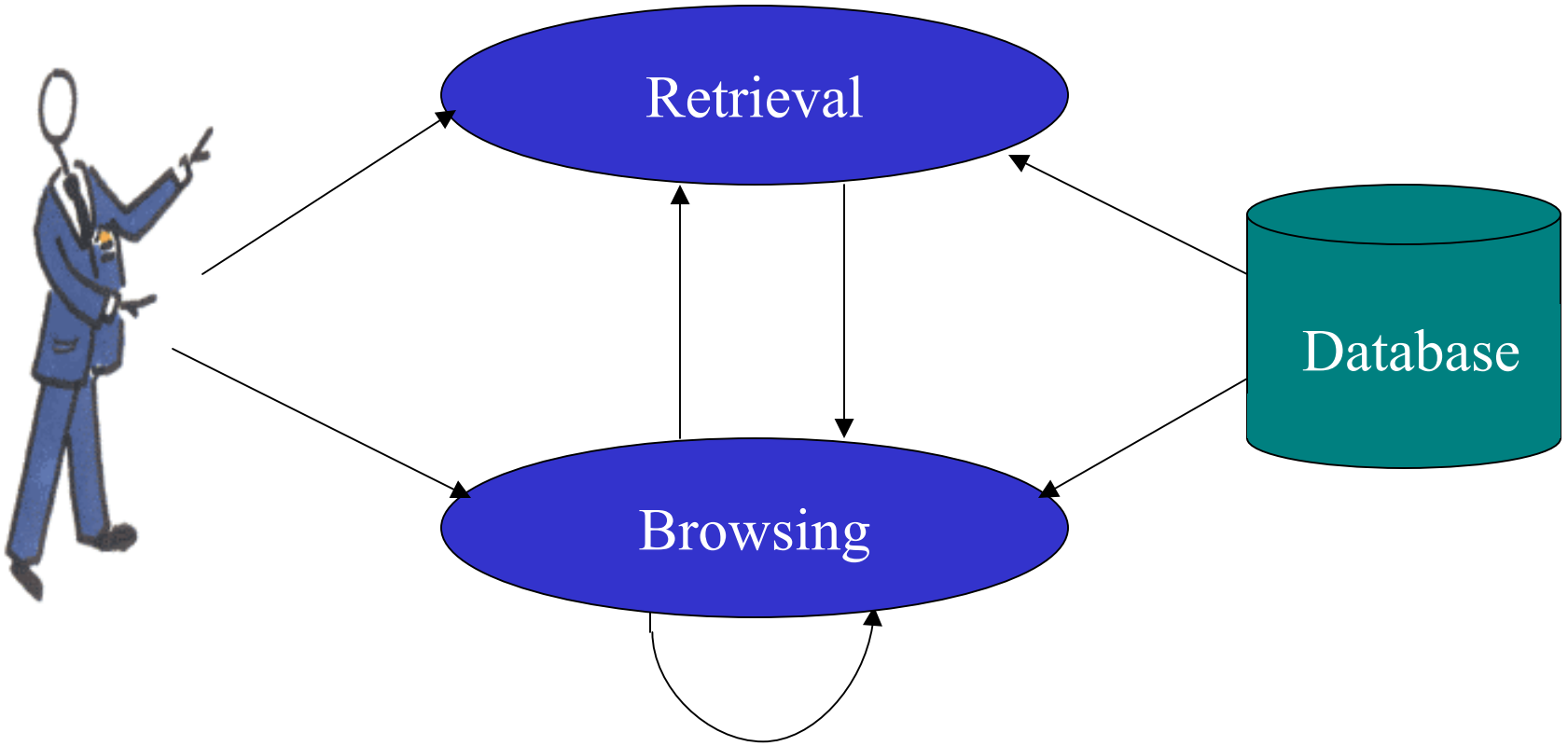
Information Retrieval

The field of *information retrieval* deals with the

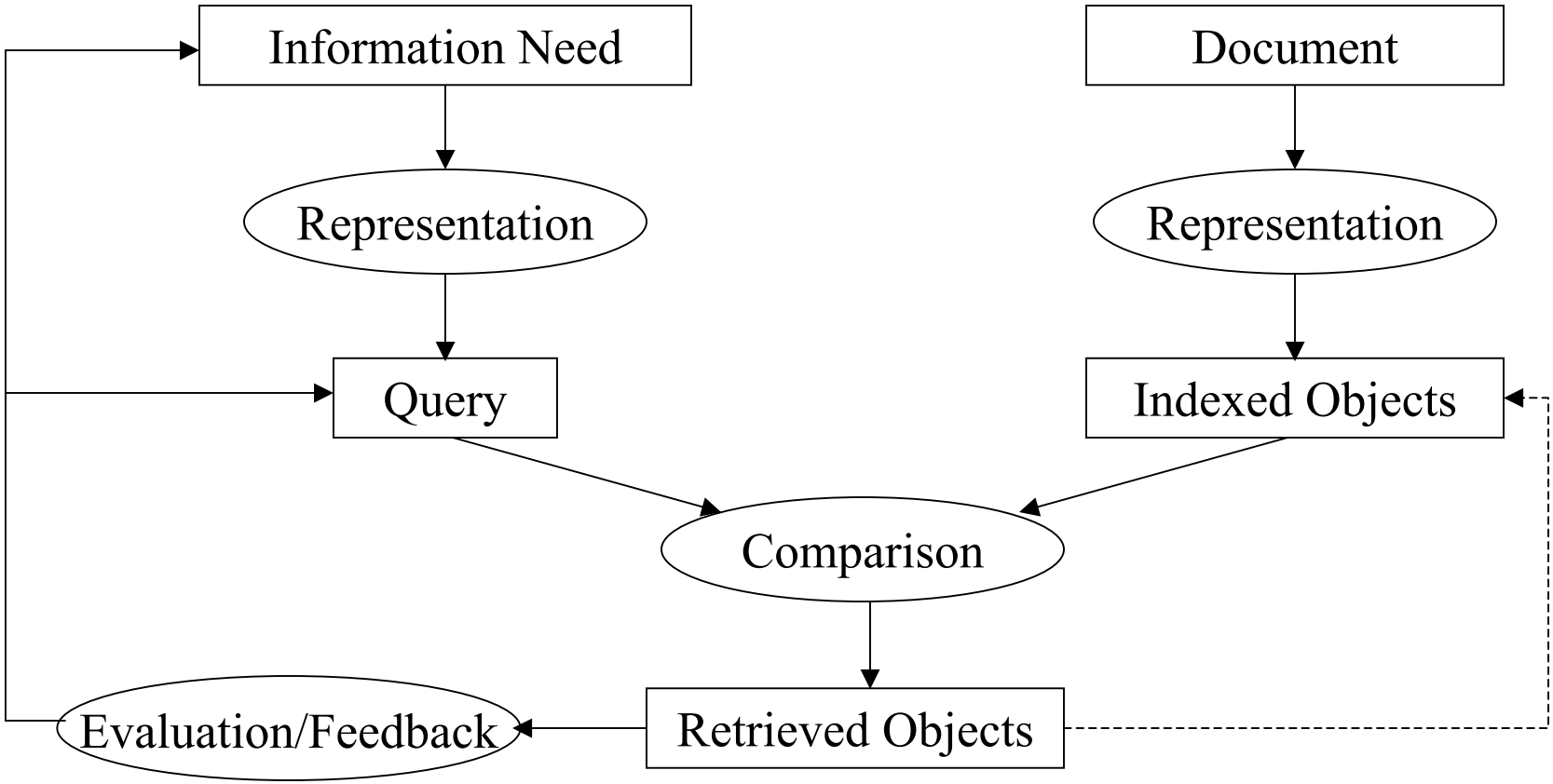
- **representation,**
- **storage,**
- **organization of,**
- **access to**

information items.

User Task

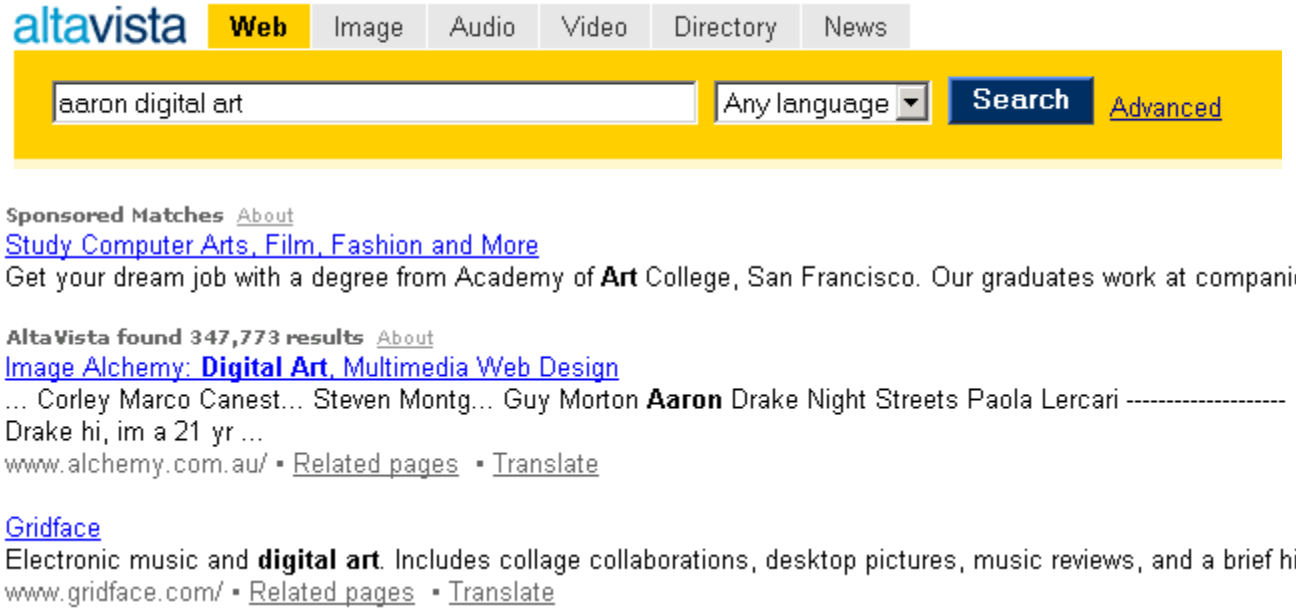


Basic IR Processes



Task Definition: Ad-hoc Retrieval

- **Search a large collection of documents to find the ones that satisfy an information need**
  - I.e., find relevant documents
- **Sometimes called "archival" retrieval**
- **Example: Web search systems**



Overview: IR Basics

- **Task Definition**
- **Retrieval Models**
  - Basic representation
  - Boolean retrieval
  - Vector space retrieval
  - TFIDF term weighting
- **Evaluation**
- **Representation and Indexing**
- **Data Structures and Access**

Text Representation

- **Manual indexing**
  - Indexers decide which keywords to assign to document based on *controlled vocabularies*
    - Examples: libraries, Medline, Yahoo
  - Significant human costs, but no computational costs
- **Automatic indexing**
  - Indexing program assigns (key)words, phrases, or other features
    - Example: words from text of document
    - Example: controlled vocabulary items assigned via automatic text classification
    - Computational cost, but no human cost

Example Document

How aspartame prevents the toxicity of ochratoxin A.

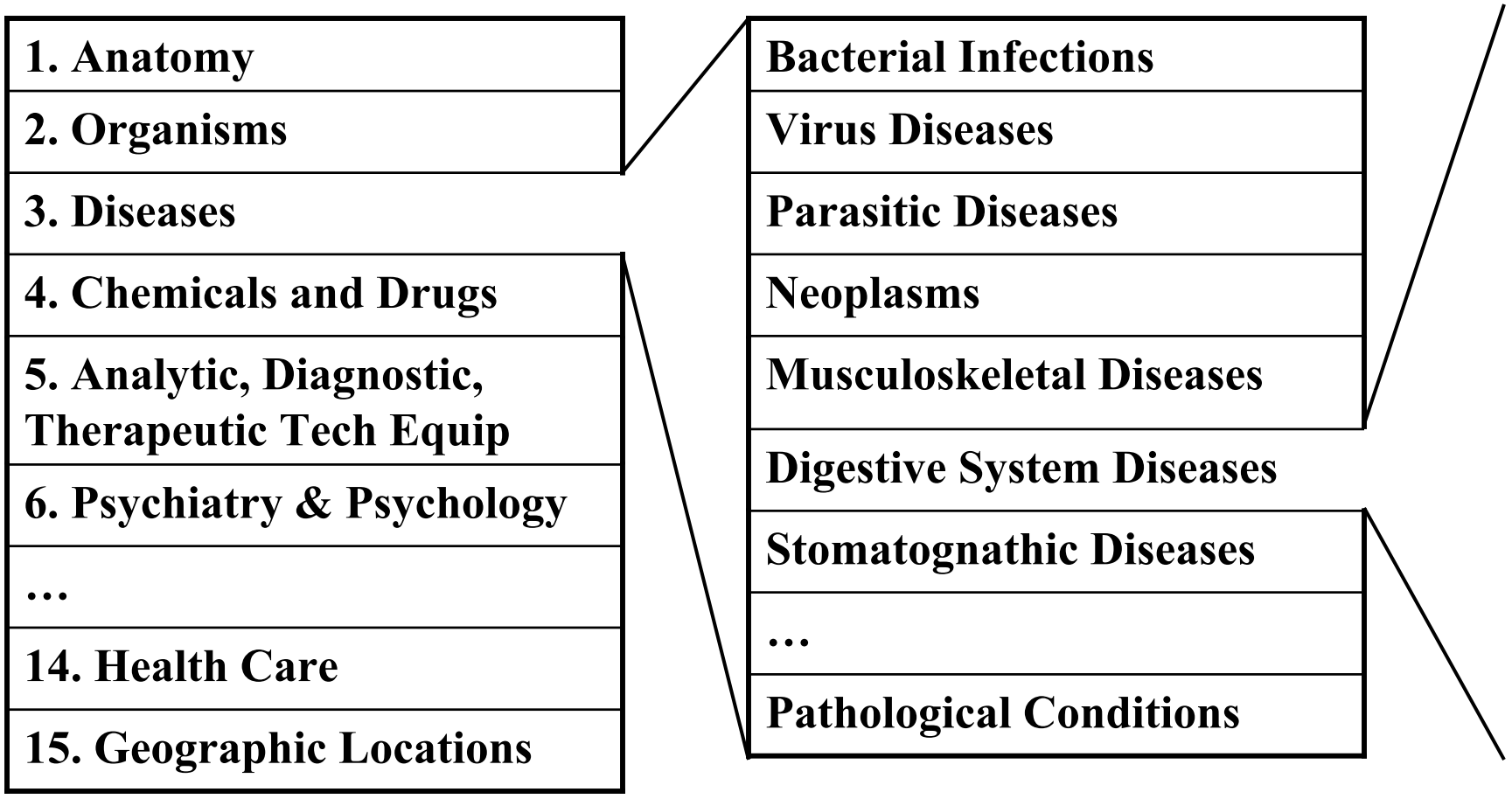
Creppy EE, Baudrimont I, Anne-Marie

Toxicology Department, University of Bordeaux, France.

The ubiquitous mycotoxin ochratoxin A (OTA) is found as a frequent contaminant of a large variety of food and feed and beverage such as beer, coffee and win. It is produced as a secondary metabolite of moulds from Aspergillus and Penicillium genera. Ochratoxin A has been shown experimentally to inhibit protein synthesis by competition with phenylalanine its structural analogue and also to enhance oxygen reactive radicals production. The combination of these basic mechanisms with the unusual long plasma half-life time (35 days in non-human primates and in humans), the metabolisation of OTA into still active derivatives and glutathione conjugate both potentially reactive with cellular macromolecules including DNA could explain the multiple toxic effects, cytotoxicity, teratogenicity, genotoxicity, mutagenicity and carcinogenicity. A relation was first recognised between exposure to OTA in the Balkan geographical

Controlled Vocabularies

Example: Medical Subject Headings (MeSH)



Controlled Vocabulary Indexing: Example

UI	98433165
AU	Creppy EE
AU	Baudrimont I
AU	Anne-Marie
TI	How Aspartame Prevents the Toxicity of Ochratoxin A.
LA	Eng
MH	Animal
MH	Aspartame/*pharmacology
MH	Blood Proteins/metabolism
MH	Cercopithecus aethiops
MH	DNA/drug effects
MH	Human
MH	Mycotoxins/*toxicity

Controlled Vocabulary Indexing

- **There are many controlled vocabularies. None is "best".**
  - Library of Congress Subject Headings (LCSH)
  - Medical Subject Headings (MeSH)
  - …
- **Tradeoffs:** coverage vs. detail
  - Example: LCSH is broad, MeSH is detailed
- **Advantage:** Solves the vocabulary mismatch problem
- **Advantage:** Makes the ontology of a domain explicit
  - Nice for browsing
- **Disadvantage**: Difficult and expensive to create, to use, and to maintain

Full-Text Indexing

Term	TF	Term	TF	Term	TF	Term	TF
the	31	by	6	peptide	4	such	3
of	26	effect	6	several	4	toxic	3
and	22	are	5	toxin	4	vitro	3
in	21	aspartame	5	also	3	when	3
a	15	exposure	5	countries	3	added	2
to	11	human	5	given	3	africa	2
as	9	with	5	it	3	balkan	2
ota	9	animals	4	preventative	3	be	2
for	8	include	4	rate	3	been	2
is	8	ochratoxin	4	shown	3	compound	2

Types of Retrieval Models: Exact Match vs. Best Match Retrieval

- **Exact match**
  - Query specifies precise retrieval criteria
  - Every document either matches or fails to match query
  - Result is a set of documents
    - Usually in no particular order w.r.t. relevance
    - Often in reverse-chronological order
  - Often called “unranked retrieval”
- **Best match**
  - Query describes retrieval criteria for desired documents
  - Every document matches the query to some degree
  - Result is a ranked list of documents, “best” first
  - Often called “ranked retrieval”

Popular Retrieval Models

- **Boolean** — **exact match**
- **Vector space** — **best match**
  - Basic vector space
  - Extended boolean model
  - Latent semantic indexing (LSI)
- **Citation analysis models** — **best match**
  - Hubs & authorities (Kleinberg, IBM Clever)
  - PageRank (Google)
- **Probabilistic models** — **best match**
  - Basic probabilistic model
  - Bayesian inference networks
  - Language models

Exact Match vs. Best Match Retrieval

- **Best-match models are usually more accurate/effective**
  - Good documents appear at the top of the rankings
  - Good documents often don't exactly match the query
    - Query may be too strict
    - Document didn't match user expectations
- **Exact match still prevalent in some markets**
  - Installed base
  - Efficient
  - Sufficient for some tasks
  - Web "advanced search"

Unranked Boolean Retrieval Model

- **Most common Exact Match model**
- **Model**
  - Retrieve documents iff they satisfy a Boolean expression
    - Query specifies precise relevance criteria
  - Documents returned in no particular order
- **Operators**
  - Logical operators: AND, OR, AND-NOT (BUT)
    - Unconstrained NOT is expensive, so it's often not included
  - Distance operators: near/1, sentence/1, paragraph/1, …
  - String matching operators: wildcard
  - Field operators: date, author, title
- **Unranked Boolean model is not the same as Boolean queries**

Example

Boolean Query

((((professional OR elite) NEAR/1 competitive NEAR/1 eating) OR (competit* NEAR/1 eat*)) AND (FIELD date 7/4/2002) AND-NOT (weight NEAR/1 loss))

- **Studies show that people are not good at creating Boolean queries**
  - People overestimate the quality of the queries they create
  - Queries are too strict: few relevant documents found
  - Queries are too loose: too many documents found (but few relevant)

Ranked Vector Space Retrieval Model

- **Best Match retrieval: return a set of documents that satisfies the query ordered by (presumed) relevance**
- **Assumption: any text object can be represented by a *term vector***
  - Examples: documents, queries, …
- **Similarity is determined by distance in a vector space**
- **The SMART system**
  - Developed at Cornell University, 1960-1999
  - Still used widely

Vector Space Representation

- **Document representation in the binary model**

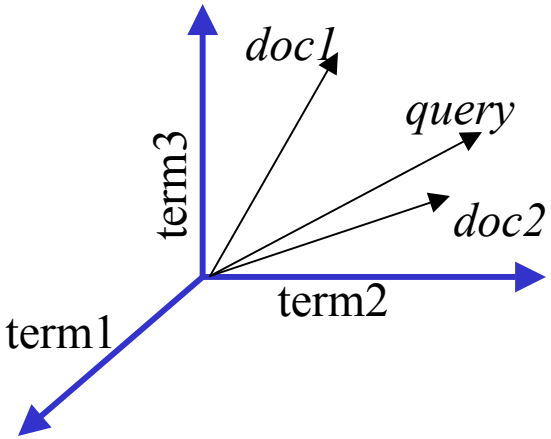
	$Term_1$	$Term_2$	$Term_3$	$Term_4$	…	$Term_n$
Doc_1	**1**	**0**	**0**	**1**	**…**	**1**
Doc_2	**0**	**1**	**1**	**0**	**…**	**0**
Doc_3	**1**	**0**	**1**	**0**	**…**	**0**
…						

- **A document is represented as a vector of binary values**
  - One dimension per term in the corpus vocabulary
- **An unstructured query can also be represented as a vector**

  Query 0 0 1 0 … 1
- **Similarity between query and document vector determines presumed relevance**

Vector Space Similarity

- **Similarity is inversely related to the angle between the vectors**



- **Doc2 is more similar to the query**
- **Rank the documents by their similarity to the query**

Vector Space Similarity

- **Cosine of the angle between the two vectors**
  - Binary term vectors

$$\frac{|X \cap Y|}{\sqrt{|X|}\sqrt{|Y|}}$$

  - Weighted term vectors

$$\frac{\sum x_i y_i}{\sqrt{\sum x_i^2}\sqrt{\sum y_i^2}}$$

What Should be the Basis of the Vector Space?

- **"Basic concepts"**
  - Difficult to determine
  - Orthogonal (by definition)
  - A relatively static vector space
- **Terms (words, word stems):**
  - Easy to determine
  - Not *really* orthogonal (orthogonal enough?)
  - A constantly growing vector space
    - New vocabulary creates new dimensions

Term Weights

- **The words of a text are not equally indicative of its meaning**

  "Most scientists think that butterflies use the position of the sun in the sky as a kind of compass that allows them to determine which way is north. Scientists think that butterflies may use other cues, such as the earth's magnetic field, but we have a lot to learn about monarchs' sense of direction."

- **Important: butterflies, monarchs, scientists, direction, compass**
- **Unimportant: most, think, kind, sky, determine, cues, learn**
- **Term weights reflect the (estimated) importance of each term**

Term Weights (TF)

- **Term frequency (*tf*)**
  - The more often a word occurs in a document, the better that term is in describing what the document is about
  - Has some basis in the 2-Poisson probabilistic model of IR
  - Often normalized, e.g. by the length of the document
  - Sometimes biased to range [0.4..1.0] to represent the fact that even a single occurrence of a term is a significant event

$$T = \frac{tf}{doc\_length} \qquad T = \frac{tf}{\max tf_d}$$

  - But terms that appear in many documents in the collection are not very useful for distinguishing a relevant document from a non-relevant one

Term Weights (IDF)

- **Inverse document frequency (*idf*)**
  - Terms that occur in many docucments in the collection are less useful for discriminating among documents
  - Document frequency (*df*): number of documents containing the term
  - *idf* often calculated as $I = \log\left(\frac{N}{df}\right) + 1$
  - Sometimes scaled to [0..1]

$$I = \frac{\log\left(\frac{N + 0.5}{df}\right)}{\log(N + 1.0)}$$

TFIDF Weights with Cosine

- **There are many variations on how term weights are calculated (see [Salton and Buckley, 1988])**
- **Weight of word i in document j is the product of TF score and IDF score**

$$tf_{i,j} * idf_{i,j}$$

- **Example**

	Term	wts	
Query	**0.0**	**0.2**	**0.0**

	Term	wts	
Doc1	**0.3**	**0.1**	**0.4**
Doc2	**0.8**	**0.5**	**0.6**

$$Sim(D_1, Q) = \frac{(0*0.3)+(0.2*0.1)+(0*0.4)}{\sqrt{0^2+0.2^2+0^2} * \sqrt{0.3^2+0.1^2+0.4^2}} = \frac{0.02}{0.10} = 0.20$$

$$Sim(D_2, Q) = \frac{(0*0.8)+(0.2*0.5)+(0*0.6)}{\sqrt{0^2+0.2^2+0^2} * \sqrt{0.8^2+0.5^2+0.6^2}} = \frac{0.10}{0.22} = 0.45$$

Settings for Ad-hoc Retrieval

- **Cross-lingual retrieval (CLIR)**
  - Query in one language (e.g. English)
  - Return documents in other languages (e.g. Korean, Greek, Tamil)
  - Sometimes called “translingual” retrieval

Settings for Ad-hoc Retrieval

- **Distributed retrieval**
  - Ad-hoc retrieval in a distributed computing environment
    - many text collections
    - reside on different machines
    - possibly different IR system for each machine
  - Issues to address include
    - Database selection
    - Merging results from different databases

Overview: IR Basics

- **Task Definition**
- **Retrieval Models**
- **Evaluation**
  - Issues
  - Test collections
  - Metrics
- **Representation and Indexing**
- **Data Structures and Access**

Evaluating Ad-hoc Retrieval Effectiveness

- **Query:** ***ski areas in New York***
- **Results:**
  - GoSki New York – New York ski areas, snow …
  - NY ski areas on "I Love NY" tourism guide
  - Ski areas in the Adirondack region
  - Press Releases
  - Lake Placid
  - Ski areas in Central NY
  - Ski areas in Cortland County
  - Ski areas in the United States
  - Nordic skiing ski areas wrap up season
  - Greek Peek
  - AYH near ski areas

Relevance

- **Relevance is difficult to define satisfactorily**
- **A relevant document is one judged useful in the context of a query**
  - Who judges?
  - What is "useful"?
  - Issue of serendipitous utility
  - Humans aren't consistent in their judgments
  - Judgment depends on more than the document and query
- **With real collections, the full set of relevant documents is never known**
- **All retrieval models include an implicit definition of relevance**

Test Collections

- **Retrieval performance is compared using a test collection**
  - Set of documents, set of queries, set of relevance judgments
- **To compare two techniques**
  - Each technique is used to evaluate queries
  - Results (set or ranked list) compared using some metric
  - Most common measures: precision, recall
- **Usually use multiple measures to get different perspectives**
- **Usually test with multiple test collections because performance is collection-dependent to some extent**

Sample Test Collections

	Cranfield	CACM	ISI	TREC2
Size (documents)	1,400	3,204	1,460	742,611
Size (MB)	1.5	2.3	2.2	2,162
Year created	1968	1983	1983	1991
Word stems	8,226	5.493	5,448	1,040,415
Stem occurrences	123,200	117,578	98,304	243,800,000
Avg DocLen (words)	88	37	67	328
Queries	225	50	35	100

Finding Relevant Documents

- **For small test collections, can review all documents for a query**
- **Not practical for large collections**
- **Pooling**
  - Retrieve documents using several techniques
  - Judge top n documents for each technique
  - Relevant set is union of relevant documents from each technique
  - Relevant set is a subset of the true relevant set
- **Possible to estimate size of true relevant set by sampling**
- **When testing:**
  - How should unjudged documents be treated?
  - How might this affect the results?

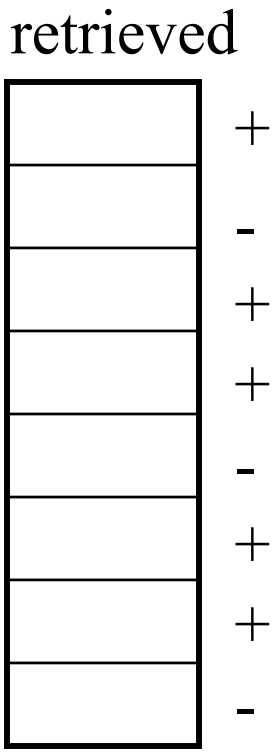
Evaluation Metrics: Precision and Recall

- **Recall**
  - Percentage of all relevant documents that are found by a search

$$R = \frac{\#\ of\ relevant\ items\ retrieved}{\#\ of\ relevant\ items\ in\ collection}$$

- **Precision**
  - Percentage of retrieved documents that are relevant

$$P = \frac{\#\ of\ relevant\ items\ retrieved}{\#\ of\ items\ retrieved}$$



$R = 5/10 = 50\%$

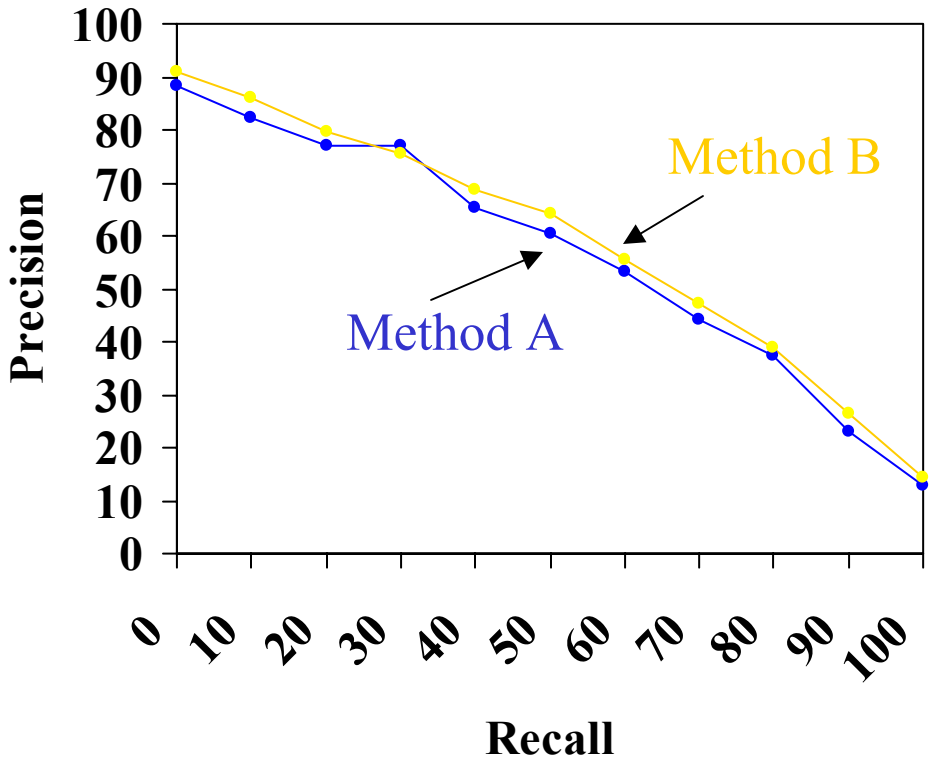
$P = 5/8 = 62.5\%$

Evaluation Metrics: Precision and Recall

- **Precision and recall are well-defined for unranked retrieval**
  - Unranked retrieval produces a set of documents
- **For ranked retrieval**
  - The entire collection is ranked (in theory)
    - Compute P at fixed recall points (e.g. precision at 20% recall)
    - Compute P at fixed rank cutoffs (e.g. precision at rank 20)

Recall Precision Tables

Recall	Method A	Method B
0	88.20	90.8 (+2.9)
10	82.40	86.1 (+4.5)
20	77.00	79.8 (+3.6)
30	77.10	75.6 (+5.4)
40	65.10	68.7 (+5.4)
50	60.30	64.1 (+6.2)
60	53.30	55.6 (+4.4)
70	44.00	47.3 (+7.5)
80	37.20	39.0 (+4.6)
90	23.10	26.6 (+15.1)
100	12.70	14.2 (+11.4)
Average	55.90	58.9 (+5.3)



Precision at Fixed Rank Cutoffs

Precision	Method A	Method B
at 5 docs	84.3	88.2
at 10 docs	79.3	84.5
at 15 docs	75.1	77.3
at 20 docs	68.2	70.5
at 30 docs	59.3	60.1
at 100 docs	35.4	34.2

F-measure

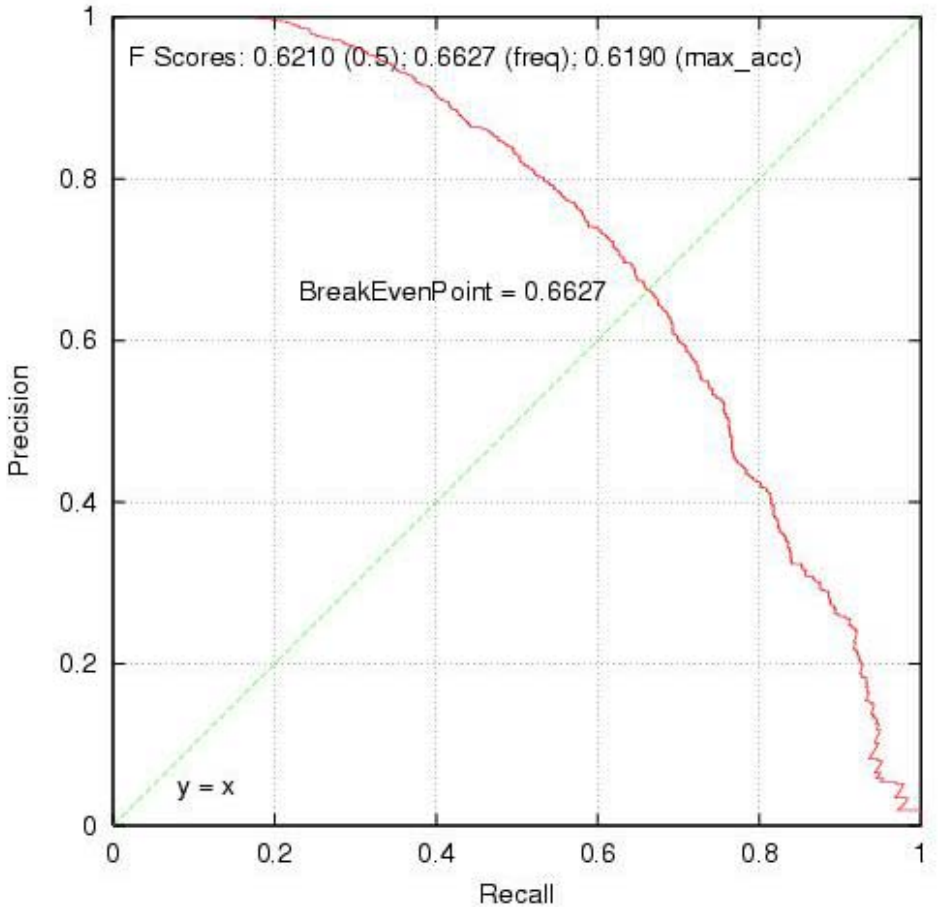
harmonic average of precision and recall

$$F = \frac{2 * (PRECISION \times RECALL)}{(PRECISION + RECALL)}$$

- **rewards results that keep recall and precision close together**
  - R=40, P=60. R/P average = 50. F-measure= 48
  - R=45, P=55. R/P average = 50. F-measure= 49.5

BreakEvenPoint

- **break even point is the point at which recall equals precision**



Overview: IR Basics

- **Task Definition**
- **Retrieval Models**
- **Evaluation**
- **Representation and Indexing**
  - Stemming
  - Stopword removal
  - Phrases
  - N-grams
- **Data Structures and Access**

Example Document

How aspartame prevents the toxicity of ochratoxin A.

Creppy EE, Baudrimont I, Anne-Marie

Toxicology Department, University of Bordeaux, France.

The ubiquitous mycotoxin ochratoxin A (OTA) is found as a frequent contaminant of a large variety of food and feed and beverage such as beer, coffee and win. It is produced as a secondary metabolite of moulds from Aspergillus and Penicillium genera. Ochratoxin A has been shown experimentally to inhibit protein synthesis by competition with phenylalanine its structural analogue and also to enhance oxygen reactive radicals production. The combination of these basic mechanisms with the unusual long plasma half-life time (35 days in non-human primates and in humans), the metabolisation of OTA into still active derivatives and glutathione conjugate both potentially reactive with cellular macromolecules including DNA could explain the multiple toxic effects, cytotoxicity, teratogenicity, genotoxicity, mutagenicity and carcinogenicity. A relation was first recognised between exposure to OTA in the Balkan geographical

Full-Text Indexing

Term	TF	Term	TF	Term	TF	Term	TF
the	31	by	6	peptide	4	such	3
of	26	effect	6	several	4	toxic	3
and	22	are	5	toxin	4	vitro	3
in	21	aspartame	5	also	3	when	3
a	15	exposure	5	countries	3	added	2
to	11	human	5	given	3	africa	2
as	9	with	5	it	3	balkan	2
ota	9	animals	4	preventative	3	be	2
for	8	include	4	rate	3	been	2
is	8	ochratoxin	4	shown	3	compound	2

Full-Text Representation Overview

- **Parse documents to recognize structure**
  - E.g., titles, dates, authors, hyperlinks
- **Scan for word tokens**
- **Stopword removal**
- **Word stemming**
  - Conflate all morphological variants of a word into a single form
- **Phrase recognition**
- **Concept / feature recognition**

Tokenization

- **Design decisions**
  - Numbers
  - Hyphenation
  - Capitalization
  - Punctuation
  - Special characters
- **Languages such as Chinese and Japanese need segmentation**
- **Record positional information for proximity operators**

Stopword Removal

- **Stopwords: words that are discarded from a document representation**
  - Function words: a, an, and, as, for, in, of, the, to, …
  - About 400 words in English.
  - Other frequent words: “Lotus” in a Lotus Support db
- **Why remove stopwords?**
  - Reduces the size of the representation
  - May also improve effectiveness of the retrieval algorithm
  - This implies a weakness in the retrieval algorithm
- **Removing stopwords makes some queries difficult to satisfy**
  - Few queries affected, so little effect on *experimental* results
    - But, very **annoying to people**

Full-Text Indexing without Stopwords

Term	TF	Term	TF	Term	TF	Term	TF
ota	9	toxin	4	compound	2	medium	2
effect	6	countries	3	culture	2	mould	2
aspartame	5	given	3	days	2	northern	2
exposure	5	preventative	3	dna	2	phenylalanine	2
humans	5	rate	3	endemic	2	prevent	2
animals	4	toxic	3	food	2	protein	2
include	4	vitro	3	genotoxicity	2	reactive	2
ochratoxin	4	added	2	incidence	2	synthesis	2
peptide	4	africa	2	induce	2	vivo	2
several	4	balkan	2	large	2	weeks	2

Words vs. Phrases vs. Concepts

- **Indexing Term:** General name for any indexing feature
- **Words**
  - word stems
  - N-grams
- **Phrases**
  - Part-of-speech
  - Statistical recognition
  - Examples: “information retrieval”, “home run”
- **Concept**
  - Example: “about medicine”, “is billing statement”
  - Manual or automatic recognition rules

Stemming

- **Group morphological variants**
  - Plural: “streets” ⇔ “street”
  - Adverbs: “fully” ⇔ “full”
  - Other inflected word forms: “goes” ⇔ “go”
  - Grouping process is called “conflation”
- **More accurate than string matching**
- **Current stemming algorithms make mistakes**
  - Conflating terms manually is difficult, time-consuming
  - Automatic conflation using rules
    - Porter Stemmer
  - Porter stemming example: “police”, “policy” => “polic”

Porter Stemming Algorithm

- **Algorithm is based on a set of condition/action rules**
  - `old_suffix->new_suffix`
- **Rules are divided into steps and are examined in sequence**
  - Step 1a: `sses->ss, ies->i, s->NULL`
    - caresses -> caress, ponies -> poni, cats -> cat
  - Step 1b: `if m>0, eed -> ee`
    - Agreed -> agree
- **Many implementations available**
  - http://www.tartarus.org/~martin/PorterStemmer/
- **Good performance on average**

Porter Stemming Example and Problems

- **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.
- **After Porter stemming and stopwords removal**
  market strateg carr compan agricultur chemic report predict market share chemic report market statist agrochem
- **Problems**
  - Sometimes too aggressive in conflation
    - e.g., policy/police, execute/executive, university/universe
  - Sometimes miss good conflations
    - e.g., European/Europe, matrices/matrix, machine/machinery

Phrases

- **Why use phrases?**
  - "Home run" more precise than "home AND run", "home, run"
  - Sometimes difficult to incorporate in retrieval model
- **Pre-coordinate phrase recognition methods**
  - recognize when document is parsed & indexed
  - Statistically, part-of-speech
  - Recognition costs incurred just once, but not flexible
- **Post-coordinate phrase recognition methods**
  - Recognize when query is evaluated
  - Query operators
  - Recognition costs incurred repeatedly, but very flexible

Phrases: Statistical Recognition

- **Consider all word bigrams**
  - Example: “hit a”, “a home”, “home run”, “run yesterday”, …
- **Select by corpus term frequency (*ctf*) or document frequency (*df*)**
  - Remove stopwords
  - Example: “home run” (54), “run yesterday” (1)
- **Reasonably accurate, but makes mistakes**
  - If a pattern occurs often, it is probably a phrase
  - Counter-example: “announced yesterday”
- **Very fast**

Phrases: Part-of-Speech Tagging

- **Assign part of speech tags**
  - Usually with a probabilistic or rule-based part of speech tagger
  - Example: "…hit/v a/art home/n run/n …."
- **Match phrases by POS patterns**
  - Example: N+, AN+
- **More accurate (maybe)**
  - N+: "home run"
  - AN+: "white house"
  - AN+: "big home run" (is this a good phrase?)
- **Reasonably fast, but slower than statistical recognition**

Part-of-Speech Phrases
TREC Example

- 65,824 United States
- 61,327 Article Type
- 33,864 Los Angeles
- 18,062 Hong Kong
- 17,788 North Korea
- 17,308 New York
- 15,513 San Diego
- 15,009 Orange County
- 12,869 prime minister
- 12,067 Soviet Union
- 10,811 Russian Federation
- 9,912 United Nations
- 8,127 Southern California
- 7,640 South Korea
- 7,620 end recording
- 7,524 European Union
- 7,086 news conference
- 6,792 City Council
- 6,348 Middle East
- 6,157 peace process
- 5,955 human rights
- 5,837 White House
- 5,778 long time
- 5,776 Armed Forces
- 5,636 Santa Ana
- 5,619 Foreign Ministry
- 5,527 Bosnia-Herzegovina
- 5,458 words indistinct
- 5,452 international community
- 5,443 vice president
- 5,247 Security Council
- 5,098 North Korean

Thesaurus

- **Resolve synonyms**
  - Query for “computer science”, document uses “informatics”
  - Standardized terms => controlled vocabulary
  - Index documents by synset
- **Resolve other relationship**
  - More general, more special terms
  - Broadening/narrowing the search
- **Query expansion**
  - Interactive by user
  - Automatic addition of terms
- **Wordnet**
  - http://www.cogsci.princeton.edu/~wn/

Document Indexing
Summary

- **Task: convert the document into a set of indexing terms**
- **Issues and Design Decisions:**
  - **Tokens:** AT&T, drive-in, 527-4701, $1,110,427, …
  - **Stopwords:** Why remove stopwords? How are stopwords defined?
  - **Stemming:** Why stem? How do stemming algorithms work?
  - **Phrases:** Why index phrases? How are phrases recognized?
  - **NLP:** What role can/should it play?
  - **Features:** Why index features? How are features recognized?
  - **Structure:** Why index structure? How is it stored and used?
  - **Efficiency:** Memory, disk space, *speed.*

Overview: IR Basics

- **Task Definition**
- **Retrieval Models**
- **Evaluation**
- **Representation and Indexing**
- **Data Structures and Access**
  - Inverted index
  - Exploiting sparsity

Why Create Index Datastructures?

- **Sequential scan of the entire collection**
  - Very flexible (e.g. search for complex patterns)
  - Available in hardware form (e.G., Fast data finder)
  - Computational and I/O costs are *O*(characters in collection)
  - Practical for only “small” collections
- **Use index for direct access**
  - An index associates a document with one or more *keys*
  - Present a key, get back the document
  - Evaluation time *O*(query term occurrences in collection)
  - Practical for “large” collections
  - Many opportunities for optimization

Inverted Index

- **Problem:** Can't predict the keys that people will use in queries
  - Every word in a document is a potential search term
  - Solution: **Index by *all* keys (terms) =>** full-text indexing
- **Inverted Index:**
  - Source file: **collection, organized by document**
    - one record per document, listing the terms that occur in this document
  - Inverted file: **collection organized by term**
    - one record per term, listing the documents the term occurs in
  - Inverted lists are today the most common indexing technique

Inverted Index Example

Source

DocID	Terms
1	machine learning
2	human learning
3	learning systems
4	database theory
5	operating systems
6	computer systems

Inverted

Term	DocIDs
computer	6
database	4
human	2
learning	1, 2, 3
operating	5
systems	3, 5, 6
theory	4

Boolean Query Evaluation

- **Operators**
  - AND: intersection of inverted document list
  - OR: union of inverted document list
  - NOT: complement of inverted document list
- **Order**
  - Equivalent queries with different evaluation order
  - Difference in efficiency

Term	DocIDs
computer	**6**
database	**4**
human	**2**
learning	**1, 2, 3**
operating	**5**
systems	**3, 5, 6**
theory	**4**

Sparse Vectors

- **Many vectors in IR are high dimensional, but sparse**
  - Store non-zero entries as sorted list
  - (0,0,0,0,4,0,0,3,0,0,1,0,0,0,0,0) => 5:4, 8:3, 11:1
- **More memory efficient**
  - O(non-zero elements)
- **Efficient set operations**
  - Merge by going through both lists in parallel
    - Intersection: keep only those where both non-zero
    - Union: keep those where at least one non-zero
    - Dot-Product: multiply and sum for those where both non-zero
    - Etc.
  - O(non-zero elements)
  - Result is again a sparse vector

Inverted Index with Positions

Source

DocID	Terms
1	machine learning
2	human learning
3	learning systems
4	database theory
5	operating systems
6	computer systems

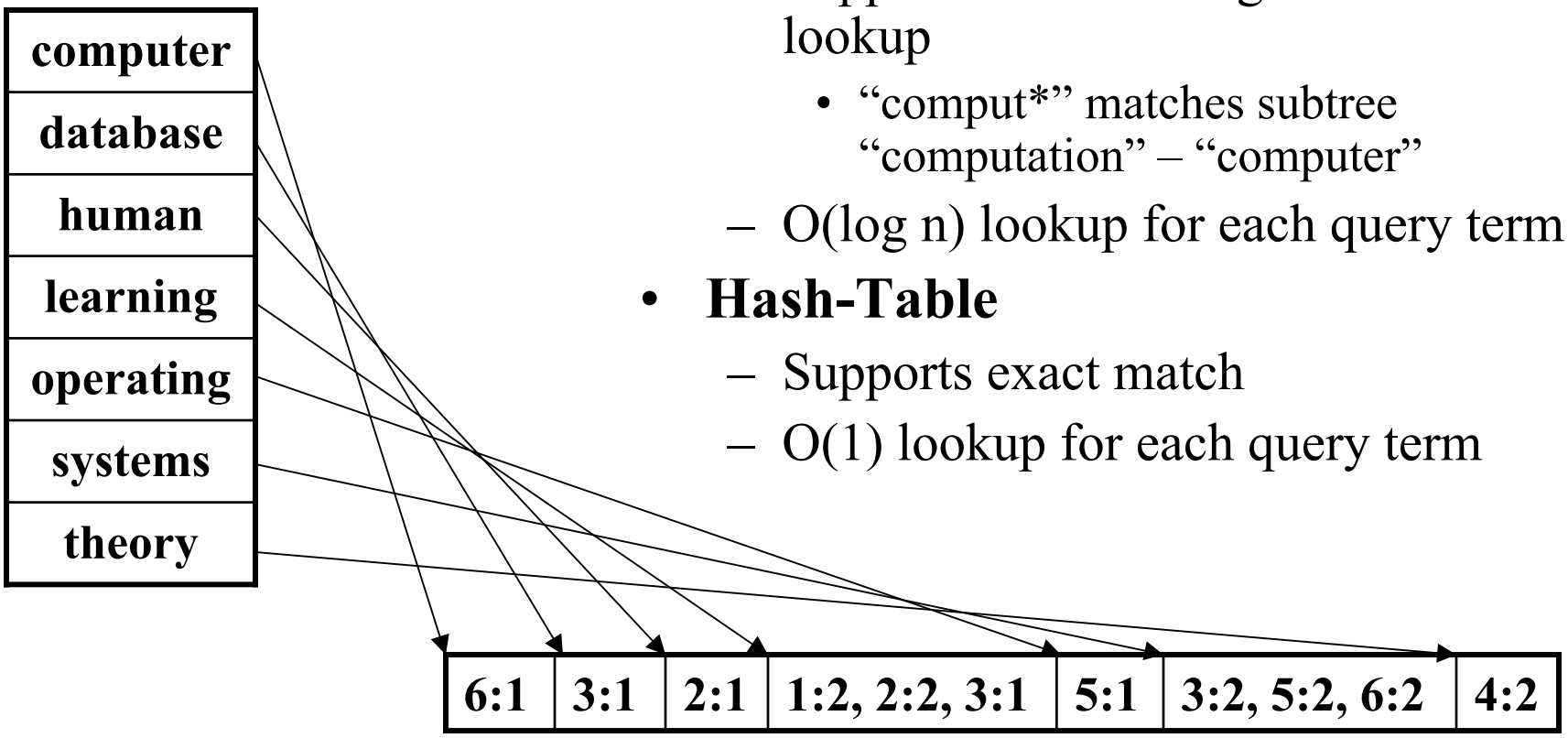
Inverted

Term	DocIDs
computer	6:1
database	3:1
human	2:1
learning	1:2, 2:2, 3:1
operating	5:1
systems	3:2, 5:2, 6:2
theory	4:2

- "…": phrases of adjacent words
- NEAR: match words within certain distance

Accessing the Inverted Lists

Hash-Table / B-Tree



- **B-Tree**
  - Supports exact & range based lookup
    - "comput*" matches subtree "computation" – "computer"
  - O(log n) lookup for each query term
- **Hash-Table**
  - Supports exact match
  - O(1) lookup for each query term

Ranking Documents

- **Simple algorithm**
  - "word1 OR word2 OR …"
  - Keep track of partial scores in accumulator
  - Might rank 100.000 document just to get the top 10 documents
  - Large memory overhead for high frequency words
- **Refinements to improve efficiency**
  - Compute only the top k documents accurately
  - Process high-weight terms first (e.g. sort inverted lists by decreasing score)
  - Limit number of accumulators (e.g. introduce accumulator only for documents with high-weigh term)

Top-Docs Ranking

- **Example:**
  - Find top 1 document only
  - Equal query weights of 1 for both query terms
- **Pruning criteria**
  - Bound on score of single document
  - Remaining maximum weight
- **Relax conditions**
  - Not necessarily optimal
  - Trade time/space for accuray

Term	DocIDs:weight
computer	**6:0.7**
database	**3:0.3**
human	**2:0.8**
learning	**2:0.9, 1:0.5, 3:0.1**
operating	**5:0.7**
systems	**6:0.3, 5:0.2, 3:0.2**
theory	**4:0.2**

Building the Inverted Index

- **Step 1: Build partial index in memory**
  - Sequentially read words from sorted documents (by DocID)
  - Look up word in current index structure (O(length of word))
    - If already contained: add DocID to end of inverted list (O(1))
    - If not contained: add word to index with new inverted list
  - If memory exhausted, write partial index to disk sorted by term
- **Step 2: Merge partial indexes**
  - Union of sparse vectors
  - Append lists if in both partial indexes
  - Each merge requires O(size of indexes)
  - O(log n) mergers

Compressing the Inverted Index

- **Inverted lists are usually compressed**
  - Uncompressed, the inverted index with word locations is about the size of the raw data
  - Compressed without position: about 10% of original text
  - Compressed with position: about 20-30% of original text
- **Distribution of numbers is skewed**
  - Most numbers are small (e.g., word locations, term frequency)
  - Distribution easily can be made more skewed
    Delta encoding: 5, 8, 10, 17 --> 5, 3, 2, 7
- **Simple compression techniques are often the best choice**
  - Goal: Time saved by reduced I/O > Time required to uncompress

Part II:
Text Classification
and the
Statistical Properties of Text

Machine Learning Summer School 2003

Thorsten Joachims
Cornell University

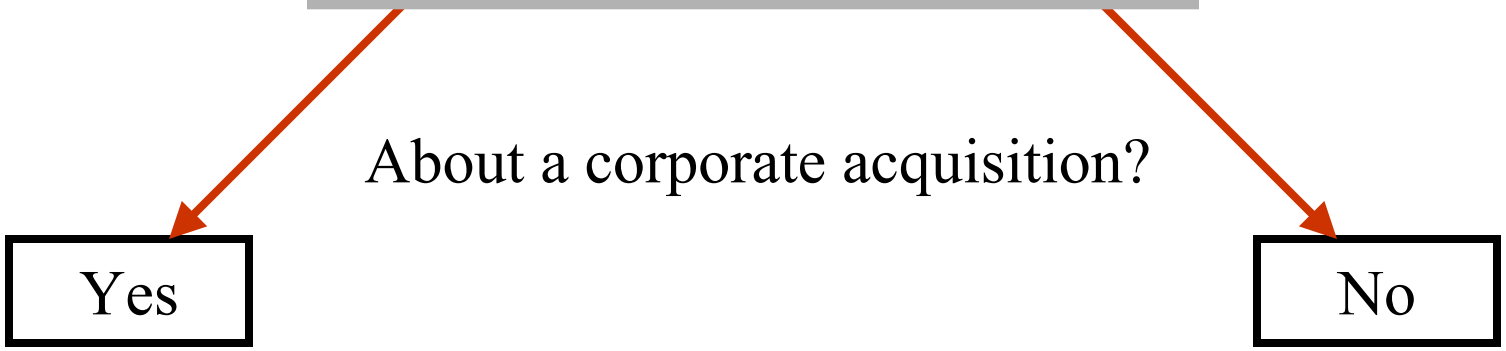
Overview: Text Classification

- **Task Definition**
  - Applications
  - Why learning?
- **Text Classification Methods**
- **Evaluation**
- **Statistical Properties of Text**

Text Classification Example

E.D. And F. MAN TO BUY INTO HONG KONG FIRM

The U.K. Based commodity house E.D. And F. Man Ltd and Singapore's Yeo Hiap Seng Ltd jointly announced that Man will buy a substantial stake in Yeo's 71.1 pct held unit, Yeo Hiap Seng Enterprises Ltd. Man will develop the locally listed soft drinks manufacturer into a securities and commodities brokerage arm and will rename the firm Man Pacific (Holdings) Ltd.



Text Classification

- **Assign pieces of text to predefined categories based on content**
- **Types of text**
  - Documents (typical)
  - Paragraphs
  - Sentences
  - WWW-Sites
- **Different types of categories**
  - By topic
  - By function
  - By author
  - By style

Text Classification Applications

- **Help-Desk Support**
  - Who is an appropriate expert for a particular problem?
- **Information Filtering Agent**
  - Which news articles are interesting to a particular person?
- **Relevance Feedback**
  - What are other documents relevant for a particular query?
- **Knowledge Management**
  - Organizing a document database by semantic categories.
- **Focused Crawling**
  - Find all the WWW pages on a particular topic.

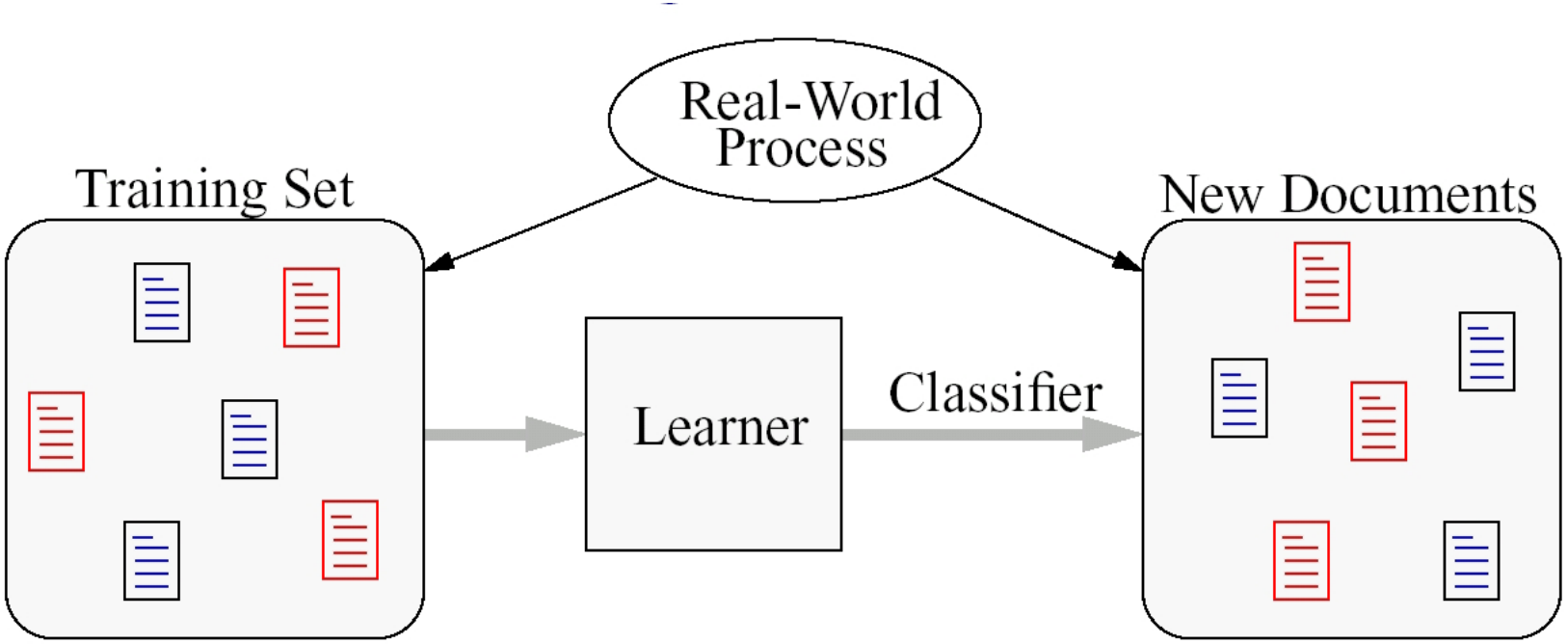
Why Learn Text Classifiers

- **Classifying documents by hand is costly and does not scale well**
  - e.g. browse all WWW pages to filter out those about job announcements
- **Humans are not really good at constructing text classification rules**
  - It is hard to write good queries
- **Sometimes there is no expert available**
  - e.g. rules for routing email
- **Often training data is cheap and plenty**
  - e.g. clickthrough from users, existing databases

Overview: Text Classification

- **Task Definition**
- **Text Classification Methods**
  - Multinomial Naïve Bayes
  - Rocchio
  - K Nearest Neighbors
- **Evaluation**
- **Statistical Properties of Text**

Learning Setting



Goal:

- Learner uses training set to find classifier with low prediction error.

Learning Setting

Process:

- Generator: Generate descriptions according to distribution *P(X)*.
- Teacher: Assigns a value to each description based on *P(Y|X)*.

Training Examples $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n) \sim P(X, Y)$

Goal:

- Find a classification rule *h* with low prediction error on new examples from distribution *P(X,Y)*

$$Err_P(h) = P(h(\vec{x}) \neq y) = \int \Delta(h(\vec{x}), y) P(\vec{x}, y) dx dy$$

Prediction Error and Loss Function

- **Prediction error**
  - Also generalization error or true error
  - Probability of making an error on a new example drawn from the same distribution *P(X,Y)*
  - Equivalent: Expected value of loss function

$$Err_P(h) = P(h(\vec{x}) \neq y) = \int \Delta(h(\vec{x}), y) P(\vec{x}, y) dx dy$$

- **Loss function**
  - Assigns amount of "penality" when making a mistake
  - Zero/One-Loss:

$$\Delta(h(\vec{x}), y) = \begin{cases} 0 & \text{if } h(\vec{x}) = y \\ 1 & \text{else} \end{cases}$$

Generative vs. Discriminative Training

Process:

- Generator: Generate descriptions according to distribution *P(X)*.
- Teacher: Assigns a value to each description based on *P(Y|X)*.

Training Examples $(\vec{x}_1, y_1), ..., (\vec{x}_n, y_n) \sim P(X, Y)$

Discriminative Training

- make assumptions about the set *H* of classifiers
- estimate error of classifiers in *H* from the training data
- select classifier with lowest error rate
- example: SVM, decision tree

Generative Training

- make assumptions about the parametric form of *P(X,Y)*.
- estimate the parameters of *P(X,Y)* from the training data
- derive optimal classifier using Bayes' rule
- example: naive Bayes

Unigram Model for Text

- **What is the probability of seeing a document in class +1 vs. class –1**
  - Need to estimate *P(X=x | Y=1)P(Y=1)* and *P(X=x | Y=-1) P(Y=-1)*
- **Assume that words are drawn randomly from class dependent lexicons (with replacement)**
- **Result**
  - l_x is the total number of words in the document x
  - w_i is the i-th word in the document

$$P(X = \vec{x}|Y = 1) = \prod_{i=1}^{l_x} P(W = w_i|Y = 1)$$

$$P(X = \vec{x}|Y = -1) = \prod_{i=1}^{l_x} P(W = w_i|Y = -1)$$

Naïve Bayes' Classifier for Text

- **Multinomial model for each class**

$$P(X = \vec{x}|Y) = \prod_{i=1}^{l_x} P(W = w_i|Y)$$

- **Prior probabilities**

$$P(Y)$$

- **Classification rule:**
  - predict class +1 if

$$P(Y = 1) \prod_{i=1}^{l_x} P(W = w_i|Y = 1) > P(Y = -1) \prod_{i=1}^{l_x} P(W = w_i|Y = -1)$$

  - else, predict class -1

Estimating the Parameters

- **Count frequencies in training data**
  - *n*: number of training examples
  - *pos/neg*: number of positive/negative training examples
  - *TF(w,y)*: number of times word w occurs in class y
  - l_y: number of words occurring in documents in class y
- **Estimating P(Y)**
  - Fraction of positive / negative examples in training data

$$\hat{P}(Y=1)=\frac{pos}{n} \qquad \hat{P}(Y=-1)=\frac{neg}{n}$$

- **Estimating P(W|Y)**
  - Smoothing with Laplace estimate

$$\hat{P}(W=w|Y=y)=\frac{TF(w,y)+1}{l_y+2}$$

Pros and Cons for Naïve Bayes

- **Pros:**
  - Explicit theoretical foundation
  - Relatively effective
  - Very simple
  - Fast in learning and classification
- **Cons:**
  - Multinomial model / independence assumption clearly wrong for text
  - Performs worse than other methods in practice
  - on some datasets it really fails badly

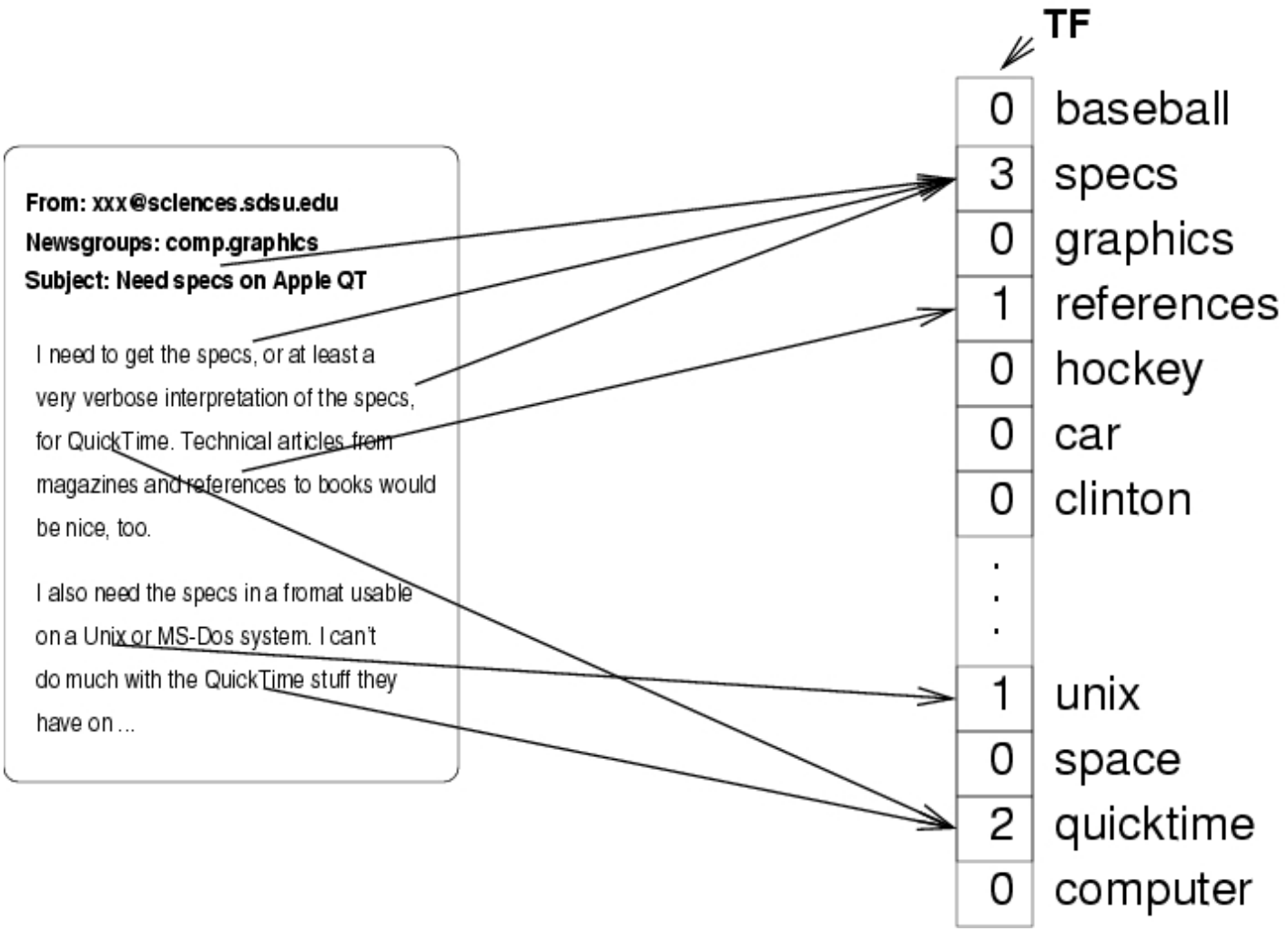
Rocchio Algorithm (Learning)

- **Given:** $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n) \sim P(X, Y)$
- **Preprocessing:**
  - Bring into vector space model representation (e.g. TFIDF)
  - Vectors normalized to Euclidian length 1
  - Split into set of positive / negative examples (ie. D_+ / D_-)
- **Training:**
  - Build prototype vector for each class
  - Compute weight vector as weighted difference between prototypes

$$\vec{w} = \frac{1}{|D_+|} \sum_{\vec{x}_i \in D_+} \vec{x}_i - \beta \frac{1}{|D_-|} \sum_{\vec{x}_i \in D_-} \vec{x}_i$$

  - Often: set negative elements of w vector to zero

Representing Text as Attribute Vectors



=> Ignore ordering of words

Rocchio Algorithm (Prediction)

- **Compute cosine between weight vector w and new example x'**
- **Prediction rule**

$$h(\vec{x}') = \begin{cases} 1 & \text{if } cos(\vec{w}, \vec{x}') > \theta \\ -1 & \text{else} \end{cases}$$

- **Threshold is a parameter, or often the cosine is just used to get a ranking**

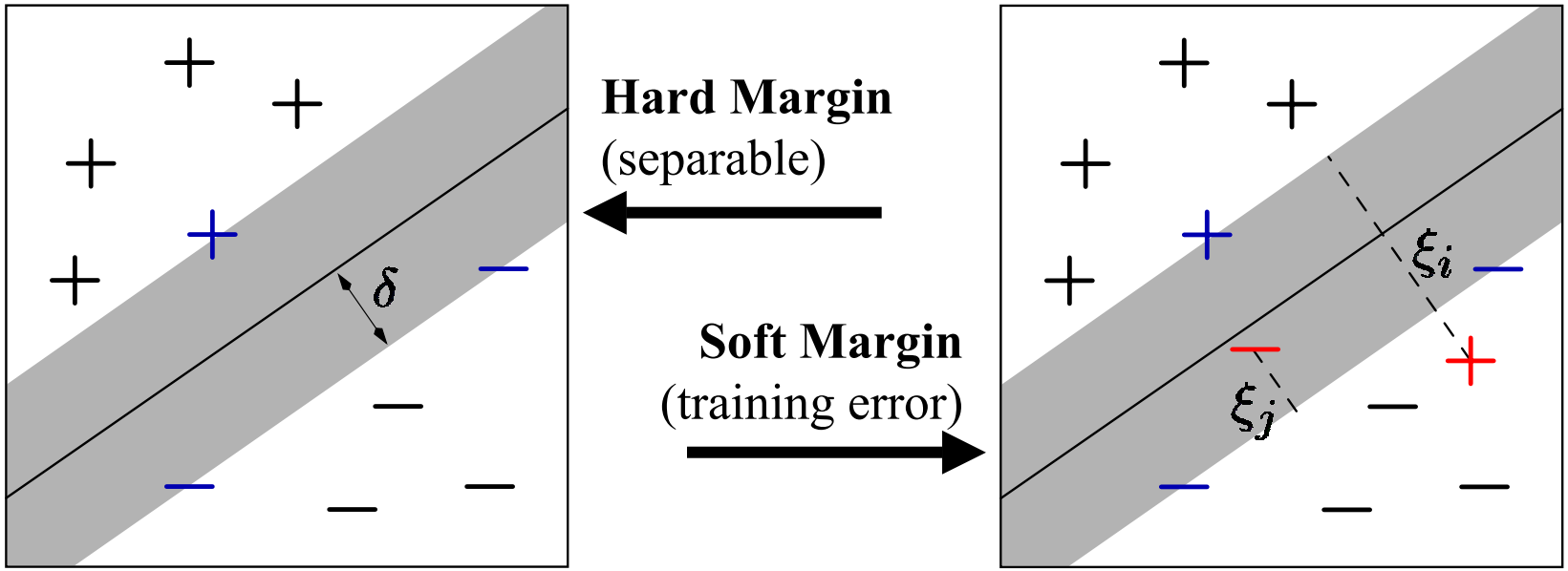
K-Nearest Neighbor

- **Given:** $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n) \sim P(X, Y)$
- **Preprocessing:**
  - Bring into vector space model representation (e.g. TFIDF)
- **Learning:**
  - None
- **Prediction rule**

$$h(\vec{x}') = sign\left(\sum_{i \in knn(\vec{x}')} y_i cos(\vec{x}_i, \vec{x}')\right)$$

Support Vector Machine [Vapnik, 1998]

- **Training Examples:** $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$ $\vec{x} \in \Re^N$ $y \in \{+1, -1\}$
- **Hypothesis Space:** $h(\vec{x}) = sgn[\vec{x}\vec{w} + b]$ with $\vec{w} = \sum_{i=1}^{n} \alpha_i y_i \vec{x}_i$
- **Training:** Find hyperplane $<\vec{w}, b>$ with minimal $\frac{1}{\delta^2} + C \sum_{i=1}^{n} \xi_i$



Feature (Subset) Selection

- **Some classifiers perform worse when using all features**
  - E.g. K-NN, Rocchio, C4.5, sometimes Naïve Bayes
- **Some classifiers are too inefficient to use all features**
  - E.g. C4.5
- **Methods**
  - Document Frequency Thresholding: Use only those words that occur at least *m* times in the training documents
  - Empirical Mutual Information: Pick words *w* with largest

$$I(w,y) = \sum_{y \in \{+1,-1\}} \sum_{w \in \{0,1\}} P(y,w) \log\left(\frac{P(y,w)}{P(y)P(w)}\right)$$

  - Also odds-ratio, chi-square score, stopword-removal, stemming

Beyond the Bag-of-Words Representation

- **Syntactic Phrases**
  - see [Leopold and Kindermann, 2002]
- **String, Sequence, and other Kernels**
  - [Lodhi et al, 2002] [Cristianini and Shawe-Taylor, 2000] [Jaakkola and Haussler, 1998], [Collins and Duffy, 2001]
- **Semantic Features and Thesaurus**
  - [Siolas and d'Alche-Buc, 2000]
- **Latent Semantic Indexing**
  - [Deerwester et al., 1990] [Cristianini et al., 2002]

Overview: Text Classification

- **Task Definition**
- **Text Classification Methods**
- **Evaluation**
  - Multi class / multi label
  - Micro / macro averaging
  - Experimental results
- **Statistical Properties of Text**

Multi-Class / Multi-Label

- **Cannot learn multi-label rules directly**
  - Most classifiers assume that each document is in exactly one class
  - Many classifiers can only learn binary classification rules
- **Most common solution: Multi-Label**
  - Learn one binary classifier for each label
  - Attach all labels, for which some classifier says positive
- **Most common solutions: Multi-Class**
  - One-against-rest: Learn one binary classifier for each label and put example into the class with the highest probability (see e.g. [Joachims, 1998, 2002] [Dumais et al., 1998]
  - Pairwise: Learn one binary classifier for each pair of labels and classify example by voting (see e.g. [Platt et al., 2000])
  - Direct SVM Approach: [Crammer and Singer, 2001]

Test Collections

- **Reuters-21578**
  - Reuters newswire articles classified by topic
  - 90 categories (multi-label)
  - 9603 training documents / 3299 test documents (ModApte)
  - ~27,000 features
  - http://www.research.att.com/ lewis/reuters21578.html
- **WebKB Collection**
  - WWW pages classified by function (e.g. personal HP, project HP)
  - 4 categories (multi-class)
  - 4183 training documents / 226 test documents
  - ~38,000 features
  - http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data
- **Ohsumed MeSH**
  - Medical abstracts classified by subject heading
  - 20 categories from “disease” subtree (multi-label)
  - 10,000 training documents/ 10,000 test documents
  - ~38,000 features
  - ftp://medir.ohsu.edu/pub/ohsumed

Example: Reuters Article (Multi-Label)

Categories: COFFEE, CRUDE

KENYAN ECONOMY FACES PROBLEMS, PRESIDENT SAYS

The Kenyan economy is heading for difficult times after a boom last year, and the country must tighten its belt to prevent the balance of payments swinging too far into deficit, President Daniel Arap Moi said.

In a speech at the state opening of parliament, Moi said high coffee prices and cheap oil in 1986 led to economic growth of five pct, compared with 4.1 pct in 1985. The same factors produced a two billion shilling balance of payments surplus and inflation fell to 5.6 pct from 10.7 pct in 1985, he added.

"But both these factors are no longer in our favour ... As a result, we cannot expect an increase in foreign exchange reserves during the year," he said.

…

Example: Ohsumed Abstract

Categories: Animal, Blood_Proteins/Metabolism, DNA/Drug_Effects, Mycotoxins/Toxicity, …

How aspartame prevents the toxicity of ochratoxin A.

Creppy EE, Baudrimont I, Anne-Marie

Toxicology Department, University of Bordeaux, France.

The ubiquitous mycotoxin ochratoxin A (OTA) is found as a frequent contaminant of a large variety of food and feed and beverage such as beer, coffee and win. It is produced as a secondary metabolite of moulds from Aspergillus and Penicillium genera. Ochratoxin A has been shown experimentally to inhibit protein synthesis by competition with phenylalanine its structural analogue and also to enhance oxygen reactive radicals production. The combination of these basic mechanisms with the unusual long plasma half-life time (35 days in non-human primates and in humans), the metabolisation of OTA into still active derivatives and glutathione conjugate both potentially reactive with cellular macromolecules including DNA could explain the multiple toxic effects, cytotoxicity, teratogenicity, genotoxicity, mutagenicity and carcinogenicity. A relation was first recognised between exposure to OTA in the Balkan geographical

Performance Measures

- **Precision/Recall Break-Even Point**
  - Intersection of PR-curve with the identity line
- **Macro-averaging**
  - First compute the measure, then compute average
  - Results in average over tasks
- **Micro-averaging**
  - First average the elements of the contingency table, then compute the measure
  - Results in average over each individual classification decision

Experimental Results

Reuters Newswir e
- 90 cate gories
- 9603 training doc.
- 3299 test doc.
- ~27000 features

WebKB Collection
- 4 cate gories
- 4183 training doc.
- 226 test doc.
- ~38000 features

Ohsumed MeSH
- 20 cate gories
- 10000 training doc.
- 10000 test doc.
- ~38000 features

microaveraged precision/recall breakeven-point [0..100]	**Reuters**	**WebKB**	**Ohsumed**
Naive Bayes	72.3	82.0	62.4
Rocchio Algorithm	79.9	74.1	61.5
C4.5 Decision Tree	79.4	79.1	56.7
k-Nearest Neighbors	82.6	80.5	63.4
SVM	**87.5**	**90.3**	**71.6**

from [Joachims, 2002]

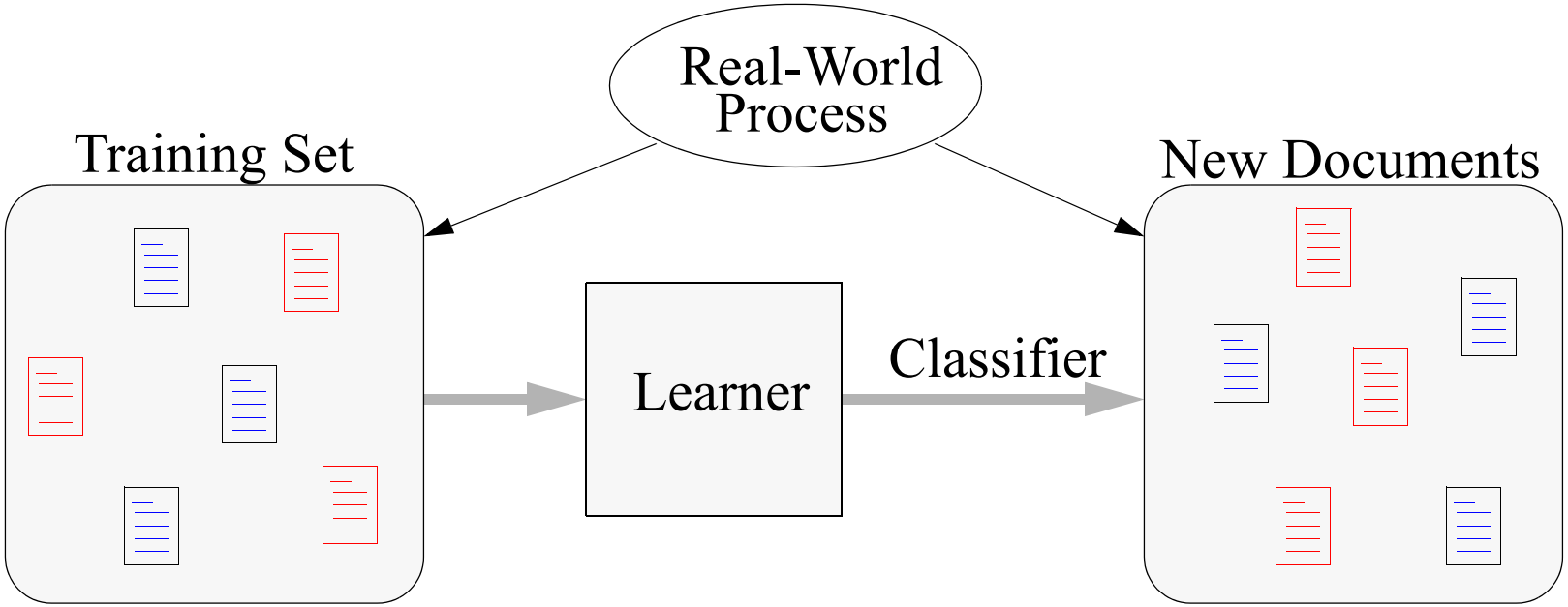
Comparison of Methods

	Naïve Bayes	Rocchio	C4.5	K-NN	SVM
Simplicity (conceptual)	+	++	-	++	-
Efficiency at training	+	+	--	++	-
Efficiency at prediction	++	++	+	--	++
Handling many classes	+	+	--	++	-
Theoretical understanding	o	--	-	o	+
Prediction accuracy	-	o	-	+	++
Stability and robustness	-	-	--	+	++

Overview: Text Classification

- **Task Definition**
- **Text Classification Methods**
- **Evaluation**
- **Statistical Properties of Text**
  - Zipf's law etc.
  - Margins for text

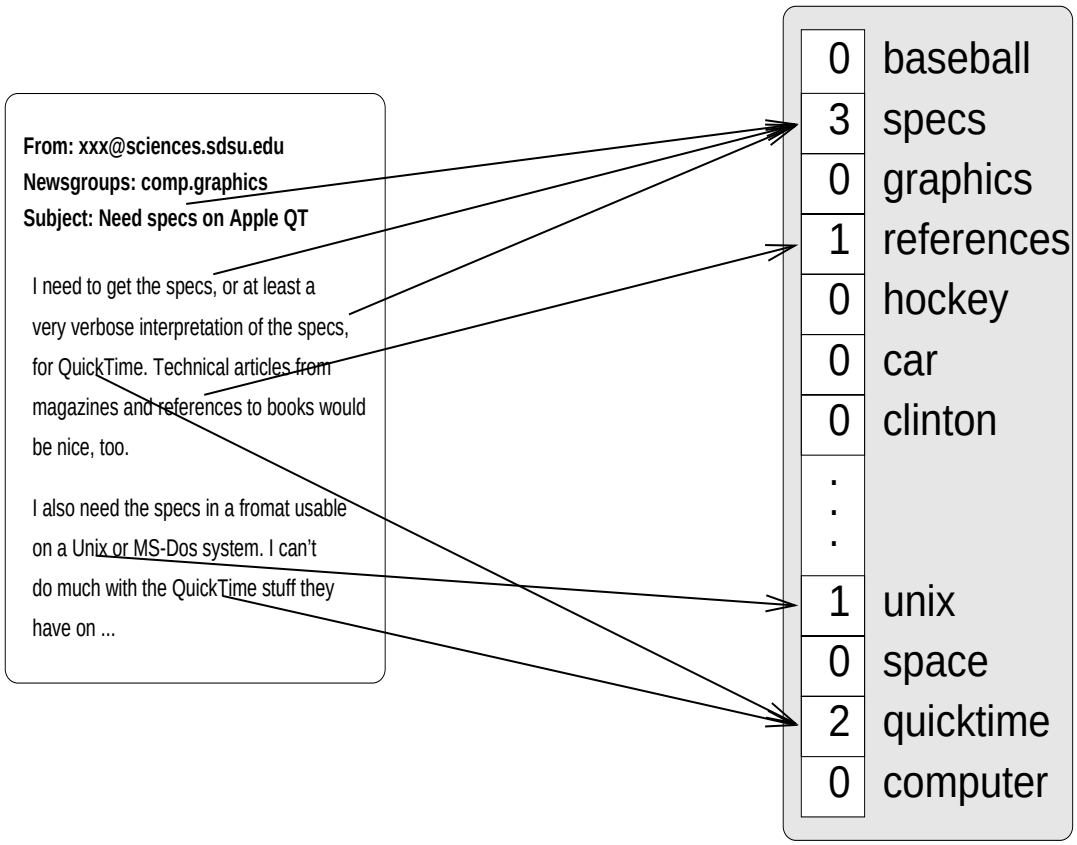
Learning Text Classifiers



Goal:

- Learner uses training set to find classifier with low prediction error.

Representing Text As Feature Vectors: Bag-of-Words

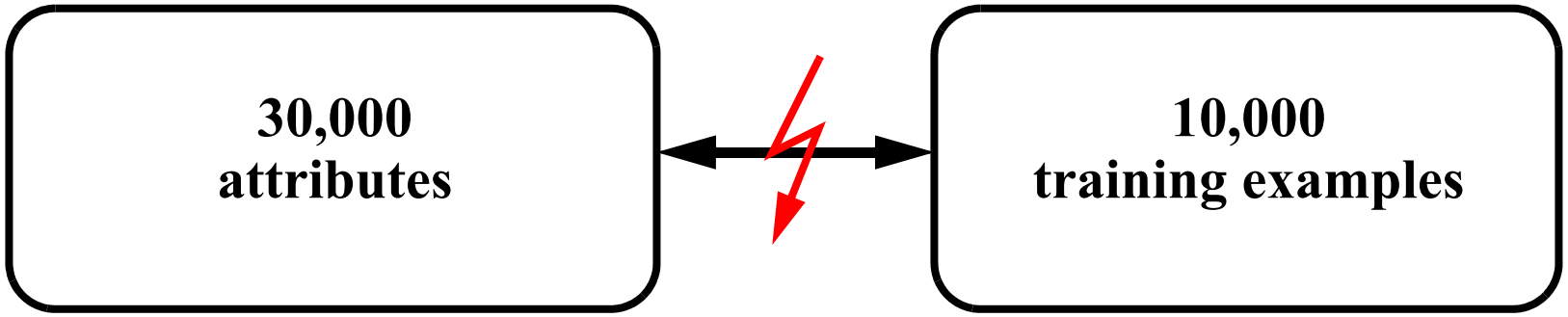


Features: words (wordstems)

Values: occurrence frequency

==> Ignore ordering of words.

Paradox of Text Classification



Experimental Results

Reuters Newswire
- 90 categories
- 9603 training doc.
- 3299 test doc.
- ~27000 features

WebKB Collection
- 4 categories
- 4183 training doc.
- 226 test doc.
- ~38000 features

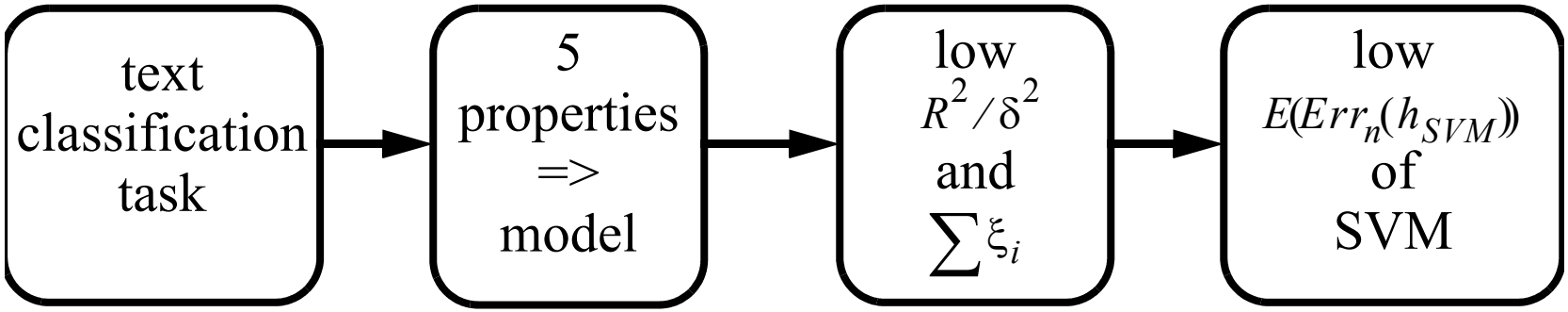
Ohsumed MeSH
- 20 categories
- 10000 training doc.
- 10000 test doc.
- ~38000 features

microaveraged precision/recall breakeven-point [0..100]	**Reuters**	**WebKB**	**Ohsumed**
Naive Bayes	72.3	82.0	62.4
Rocchio Algorithm	79.9	74.1	61.5
C4.5 Decision Tree	79.4	79.1	56.7
k-Nearest Neighbors	82.6	80.5	63.4
SVM	**87.5**	**90.3**	**71.6**

[Joachims, 2002]

Why Do SVMs Work Well for Text Classification?

A statistical learning model of text classification with SVMs:



Margin/Loss Based Bound on the Expected Error

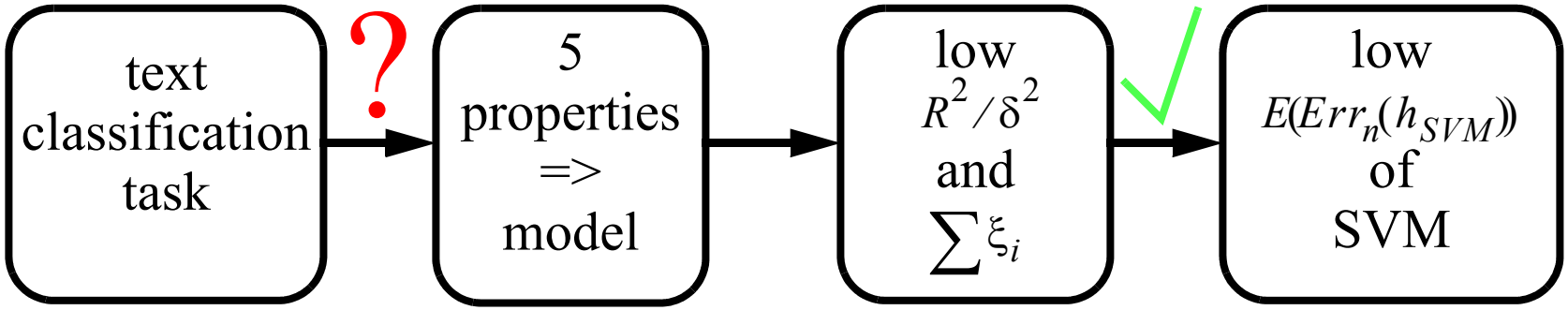
Theorem: The expected error of a soft margin SVM is bounded by

$$E(Err_n(h_{SVM})) \le \frac{\rho E\left(\frac{R^2}{\delta^2}\right) + \rho C R^2 E\left(\sum_{n=1}^{n+1} \xi_i\right)}{n+1} \qquad C \ge \frac{1}{\rho R^2}$$

$$E(Err_n(h_{SVM})) \le \frac{\rho E\left(\frac{R^2}{\delta^2}\right) + \rho (C R^2 + 1) E\left(\sum_{n=1}^{n+1} \xi_i\right)}{n+1} \qquad C < \frac{1}{\rho R^2}$$

Where $E\left(\frac{R^2}{\delta^2}\right)$ is the expected soft margin and $E\left(\sum_{n=1}^{n+1} \xi_i\right)$ is the expected training loss on training sets of size $n+1$.

First Step Completed



Properties 1+2: Sparse Examples in High Dimension

- High dimensional feature vectors (30,000 features)
- Sparse document vectors: only a few words of the whole language occur in each document

	Training Examples	Number of Features	Distinct Words (Sparsity)
Reuters Newswire Articles	9,603	27,658	74 (0.27%)
Ohsumed MeSH Abstracts	10,000	38,679	100 (0.26%)
WebKB WWW-Pages	3,957	38,359	130 (0.34%)

Property 3: Heterogeneous Use Of Words

MODULAIRE BUYS BOISE HOMES PROPERTY

Modulaire Industries said it acquired the design library and manufacturing rights of privately-owned Boise Homes for an undisclosed amount of cash. Boise Homes sold commercial and residential prefabricated structures, Modulaire said.

JUSTICE ASKS U.S. DISMISSAL OF TWA FILING

The Justice Department told the Transportation Department it supported a request by USAir Group that the DOT dismiss an application by Trans World Airlines Inc for approval to take control of USAir. ``Our rationale is that we reviewed the application for control filed by TWA with the DOT and ascertained that it did not contain sufficient information upon which to base a competitive review,'' James Weiss, an official in Justice's Antitrust Division, told Reuters.

USX, CONS. NATURAL END TALKS

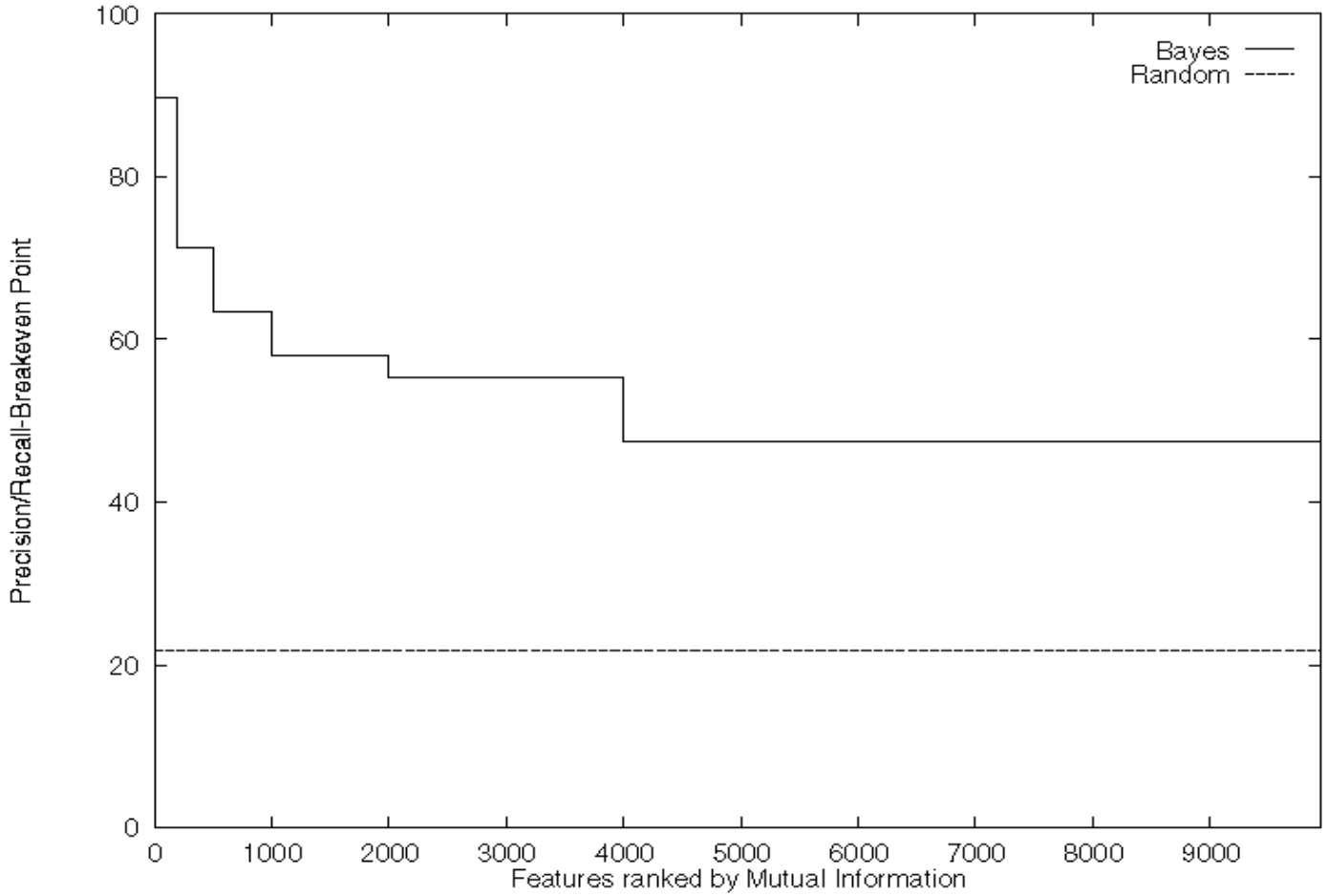
USX Corp's Texas Oil and Gas Corp subsidiary and Consolidated Natural Gas Co have mutually agreed not to pursue further their talks on Consolidated's possible purchase of Apollo Gas Co from Texas Oil. No details were given.

E.D. And F. MAN TO BUY INTO HONG KONG FIRM

The U.K. Based commodity house E.D. And F. Man Ltd and Singapore's Yeo Hiap Seng Ltd jointly announced that Man will buy a substantial stake in Yeo's 71.1 pct held unit, Yeo Hiap Seng Enterprises Ltd. Man will develop the locally listed soft drinks manufacturer into a securities and commodities brokerage arm and will rename the firm Man Pacific (Holdings) Ltd.

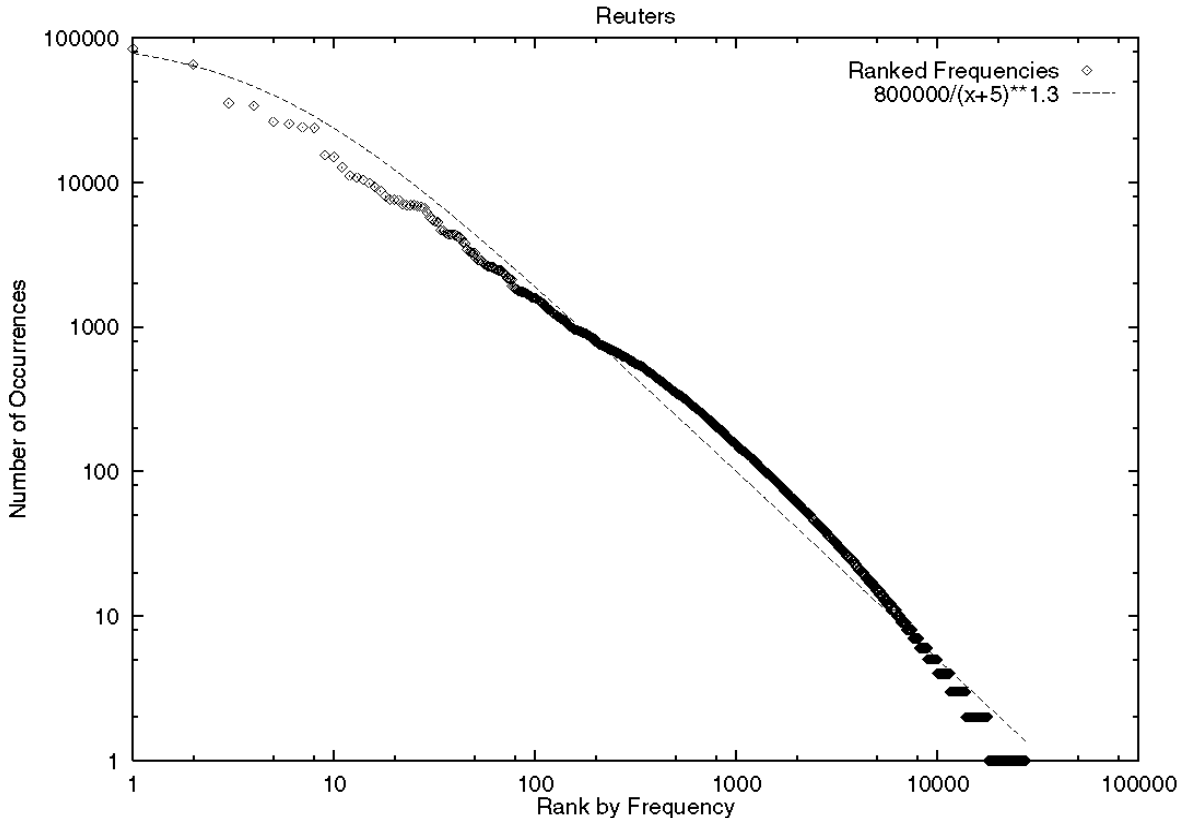
No pair of documents shares any words, but "it", "the", "and", "of", "for", "an", "a", "not", "that", "in".

Property 4: High Level Of Redundancy



=> Few features are irrelevant!

Property 5: "Zipf's Law"



Zipf's Law: In text, the *i*-th frequent word occurs $f_i = \frac{k}{(c+i)^{\Theta}}$ times.

=> Most words occur very infrequently!

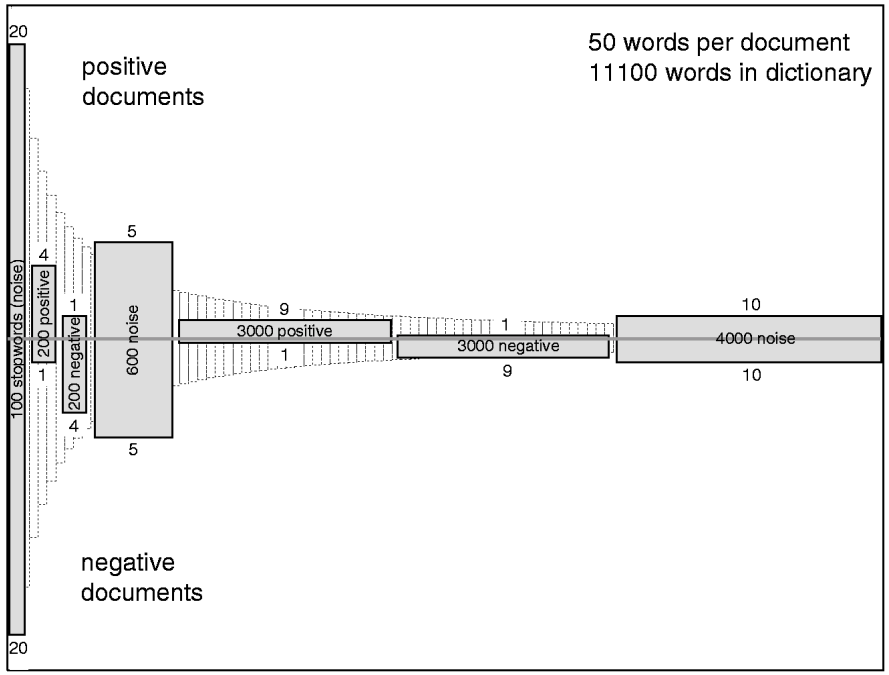
Text Classification Model

Definition: For the TCat-concept there are s disjoint sets of features.

$$TCat([p_1|n_1|f_1], \ldots, [p_s|n_s|f_s])$$

Each positive (negative) example contains p_i (n_i) occurrences from the f_i features in set i.

Example: $TCat\begin{pmatrix}[20|20|100]\\ [4|1|200]\\ [1|4|200]\\ [5|5|600]\\ [9|1|3000]\\ [1|9|3000]\\ [10|10|4000]\end{pmatrix}$



TCat-Concept for WebKB "Course"

$$TCat\begin{pmatrix}[77|29|98],\ [4|21|52] \\ [16|2|431],\ [1|12|341] \\ [9|1|5045],\ [1|21|24276] \\ [169|191|8116]\end{pmatrix}$$

high frequency
medium frequency
low frequency
rest

	high frequency	medium frequency	low frequency
pos	**98 words** all any assignment assignments available be book c chapter class code course cse description discussion document due each eecs exam exams fall final ... section set should solution solutions spring structures students syllabus ta text textbook there thursday topics tuesday unix use wednesday week will you your	**431 words** account acrobat adapted addison adt ahead aho allowed alternate announced announcement announcements answers appointment approximately ... tuesdays turing turn turned tuth txt uidaho uiowa ullman understand ungraded units unless upenn usr vectors vi walter weaver wed wednesdays weekly weeks weights wesley yurttas	**5045 words** 002cc 009a 00a 00om 01oct 01pm 02pm 03oct 03pm 03sep 04dec ... gradable gradebook gradebooks gradefreq1 gradefreq2 gradefreq3 graders gradesheet gradients grafica grafik ... zimmermann zinc zipi zipser zj zlocate znol zoran zp zwatch zwhere zwiener zyda
neg	**52 words** acm address am austin ca california center college computational conference contact current currently d department dr faculty fax graduate group he ... me member my our parallel performance ph pp proceedings professor publications recent research sciences support technical technology university vision was working	**341 words** aaai academy accesses accurate adaptation advisor advisory affiliated affiliations agent agents alberta album alumni amanda america amherst annual ... victoria virginia visiting visitors visualization vita vitae voice wa watson weather webster went west wi wife wireless wisconsin worked workshop workshops wrote yale york	**24276 words** 0a 0b 0b1 0e 0f 0r 0software 0x82d4ff 100k 100mhz 100th 1020x620 102k 103k ... lunar lunches lunchtime lund lundberg lunedi lung luniewski luo luong lupin lupton lure lurker lus ... zuo zuowei zurich zvi zw zwaenepoel zwarico zwickau zwilling zygmunt zzhen00
	high frequency	medium frequency	low frequency

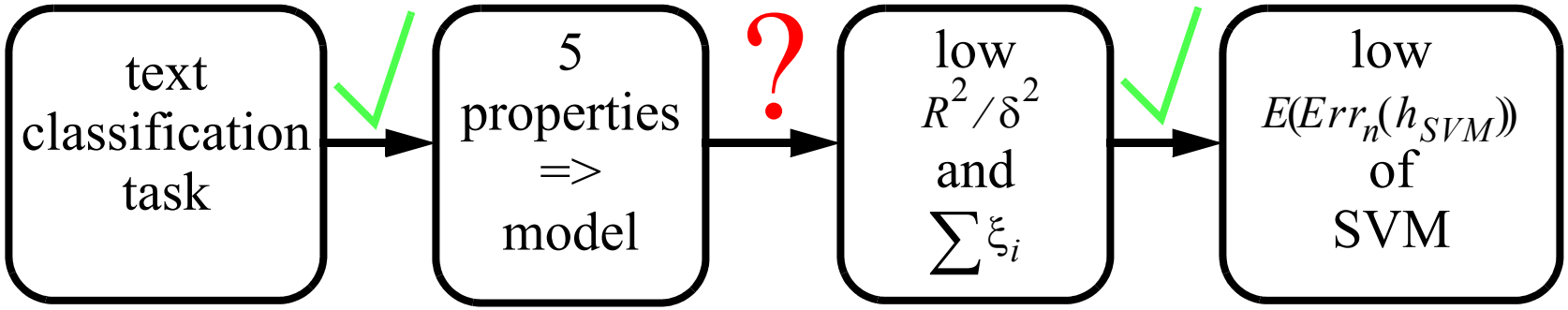
Real Text Classification Tasks as TCat-Concepts

Reuters "Earn": $TCat\begin{pmatrix} [33|2|65], [32|65|152] \\ [2|1|171], [3|21|974] \\ [3|1|3455], [1|10|17020] \\ [78|52|5821] \end{pmatrix}$ *high frequency* / *medium frequency* / *low frequency* / *rest*

Webkb "Course": $TCat\begin{pmatrix} [77|29|98], [4|21|52] \\ [16|2|431], [1|12|341] \\ [9|1|5045], [1|21|24276] \\ [169|191|8116] \end{pmatrix}$ *high frequency* / *medium frequency* / *low frequency* / *rest*

Ohsumed "Pathology": $TCat\begin{pmatrix} [2|1|10], [1|4|22] \\ [2|1|92], [1|2|94] \\ [5|1|4080], [1|10|20922] \\ [197|190|13459] \end{pmatrix}$ *high frequency* / *medium frequency* / *low frequency* / *rest*

Second Step Completed



The Margin δ^2 of TCat-Concepts

Lemma 1: For $TCat([p_1|n_1|f_1], \ldots, [p_s|n_s|f_s])$ -concepts there is always a hyperplane passing through the origin with margin δ^2 at least

$$\delta^2 \geq \frac{ad - b^2}{a + 2b + d} \qquad with \qquad a = \sum_{i=1}^{s} \frac{p_i^2}{f_i}$$

$$d = \sum_{i=1}^{s} \frac{n_i^2}{f_i}$$

$$b = \sum_{i=1}^{s} \frac{n_i p_i}{f_i}$$

Example: The previous example WebKB "course" has a margin of at least

$$\delta^2 \geq 0.23$$

The Length R^2 of Document Vectors

Lemma 2: If the ranked term frequencies f_i in a document with l words have the form of the generalized Zipf's Law

$$f_i = \frac{k}{(c+i)^{\Theta}}$$

based on their frequency rank i, then the Euclidean length of the document vector $\vec{x}$ is bounded by

$$\|\vec{x}\| \leq \sqrt{\sum_{i=1}^{d} \left(\frac{k}{(c+i)^{\Theta}}\right)^2} \qquad with \qquad \sum_{i=1}^{d} \frac{k}{(c+i)^{\Theta}} = l$$

Example: For WebKB "course" with

$$f_i = \frac{470000}{(5+i)^{1.25}}$$

follows that $R^2 \leq 1900$.

R^2, δ^2, and $\sum \xi_i$ for Text Classification

Reuters Newswire Stories

- 10 most frequent categories
- 9603 training examples
- 27658 attributes

$$E(Err_P(h_{SVM})) \leq \frac{E\left(\frac{R^2}{\delta^2}\right) + CR^2 E\left(\sum_{i=1}^{n+1} \xi_i\right)}{n+1}$$

	R^2/δ^2	$\sum \xi_i$
earn	1143	0
acq	1848	0
money-fx	1489	27
grain	585	0
crude	810	4

	R^2/δ^2	$\sum \xi_i$
trade	869	9
interest	2082	33
ship	458	0
wheat	405	2
corn	378	0

Sensitivity Analysis

What makes a text classification problem suitable for a linear SVM?

High Redundancy:

$$TCat\begin{pmatrix} [40|40|50] \\ [25|5|1000], [5|25|1000] \\ [30|30|30000] \end{pmatrix} \begin{matrix} high\ frequency \\ medium\ frequency \\ low\ frequency \end{matrix}$$

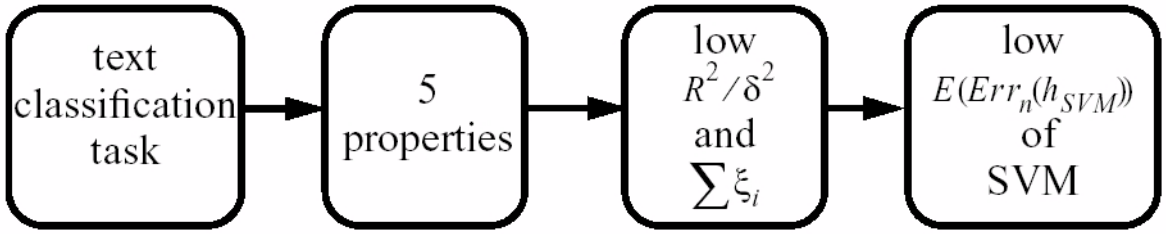
High Discriminatory Power:

$$TCat\begin{pmatrix} [40|40|50] \\ [15|0|500], [0|15|500], [15|15|1000] \\ [30|30|30000] \end{pmatrix} \begin{matrix} high\ frequency \\ medium\ frequency \\ low\ frequency \end{matrix}$$

High Frequency:

$$TCat\begin{pmatrix} [16|4|10], [4|16|10], [20|20|30] \\ [30|30|2000] \\ [30|30|30000] \end{pmatrix} \begin{matrix} high\ frequency \\ medium\ frequency \\ low\ frequency \end{matrix}$$

Summary
SVMs and the Stat. Properties of Text Classification



Intuition: If the problem can be cast as a TCat-concept with

- high redundancy,
- strongly discriminating features
- particularly in the high frequency region

then linear SVMs achieve a low generalization error [Joachims, 2002].

Assumptions and Restrictions:

- no noise (attribute and classification)
- no variance (only "average" examples)
- only upper bounds, no lower bounds

Part III:
Tasks and Research Areas
in
Language Technology

Machine Learning Summer School 2003

Thorsten Joachims

Cornell University

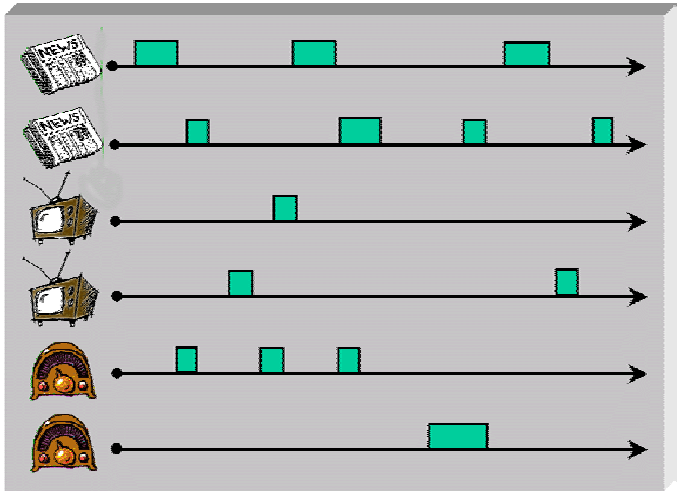
Overview: Tasks in Lang Tech

- **Topic Detection and Tracking**
- **Part-of-Speech Tagging**
- **Information Extraction**
- **Named Entity Recognition**
- **Learning Retrieval Functions**

Topic Detection and Tracking

Challenge:

Develop applications that organize and locate relevant stories from a continuous feed of news stories from various media



TDT Evaluations:

- Annual evaluation 1998 to 2002
- Promote research on basic components for such a system
- Training and test data from broadcast news and newswire text
- Info: http://www.nist.gov/speech/tests/tdt/tdt2001

TDT Tasks

- **Story Segmentation**:
  - Detect changes between topically cohesive sections
- **Topic Tracking**:
  - Keep track of stories similar to a set of example stories
- **Topic Detection**:
  - Build clusters of stories that discuss the same topic
- **First Story Detection:**
  - Detect if a story is the first story of a new, unknown topic
- **Link Detection:**
  - Detect whether or not two stories are topically linked

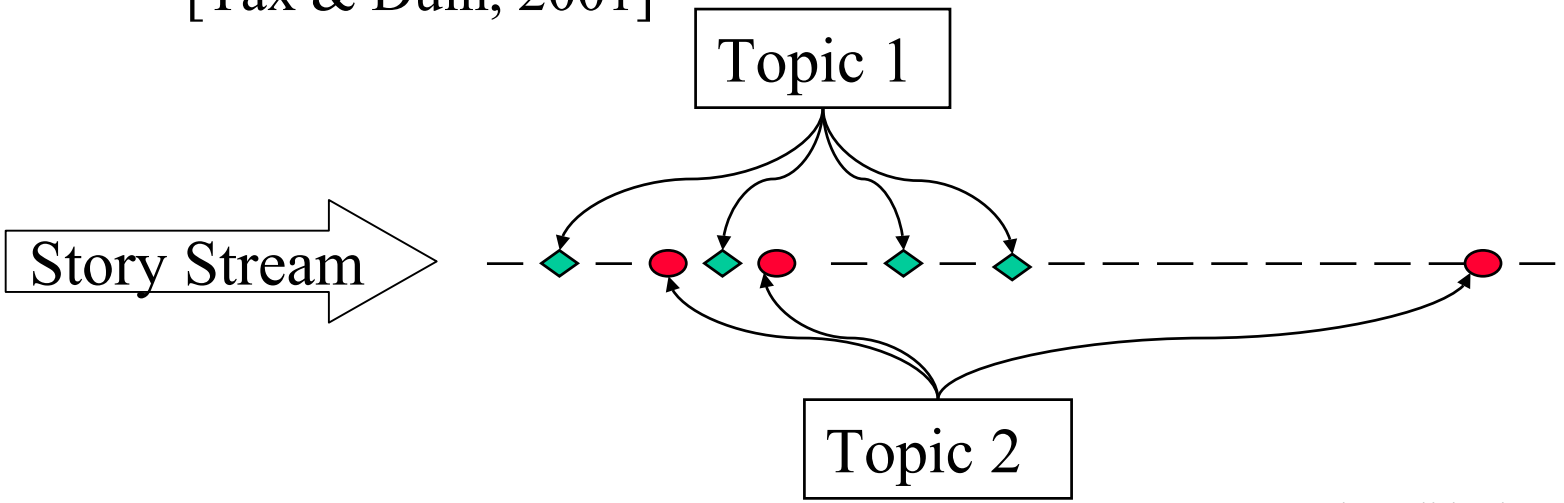
Topic Detection

Task:

- Detect topics in terms of the (clusters of) stories that discuss them.

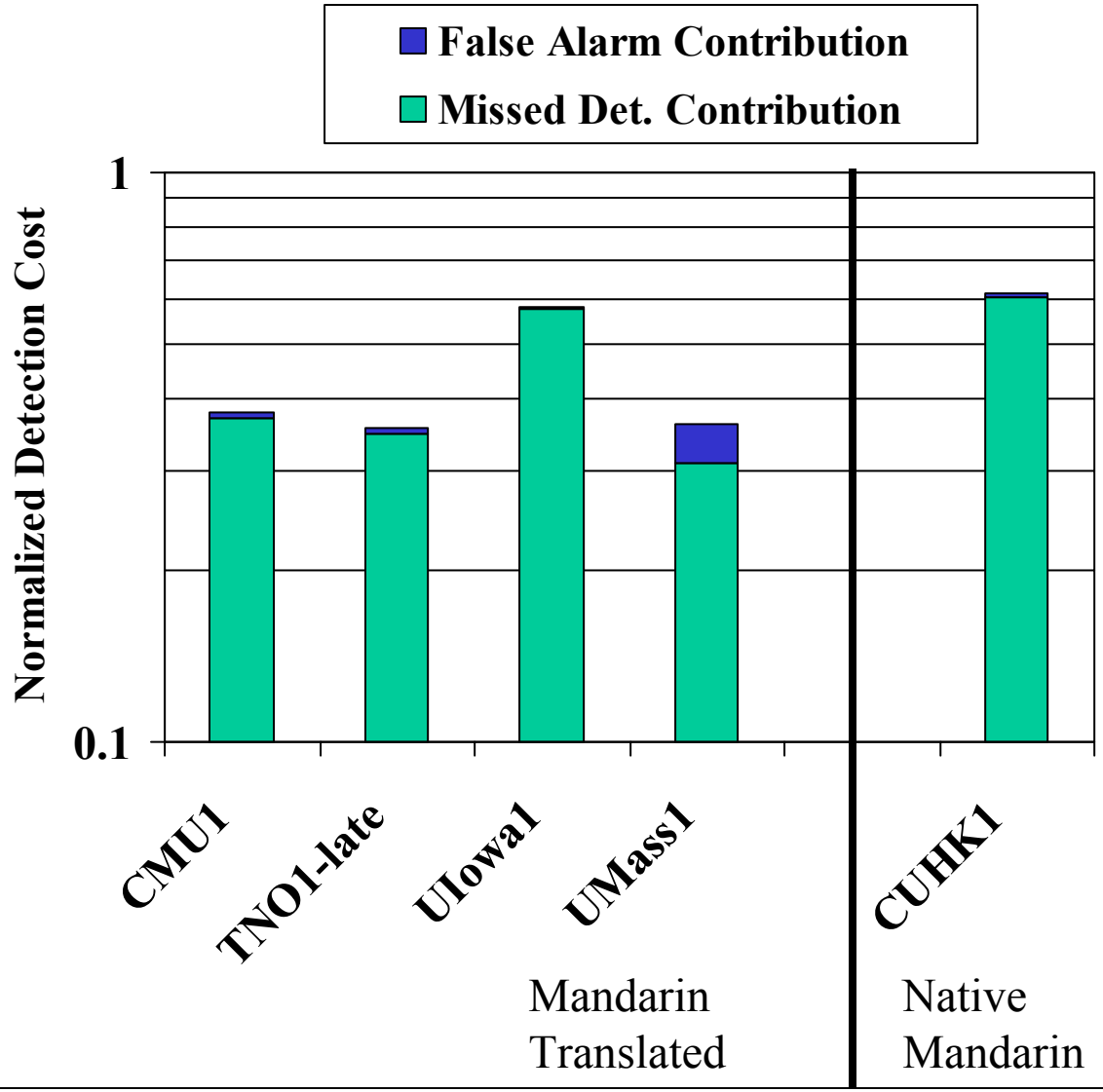
Properties:

- "Unsupervised" topic training
- New topics must be detected as the incoming stories are processed.
- Connection to novelty / outlier detection [Schölkopf et al., 1995] [Tax & Duin, 2001]



Based on slides by Jonathan Fiscus

2001 Topic Detection Results



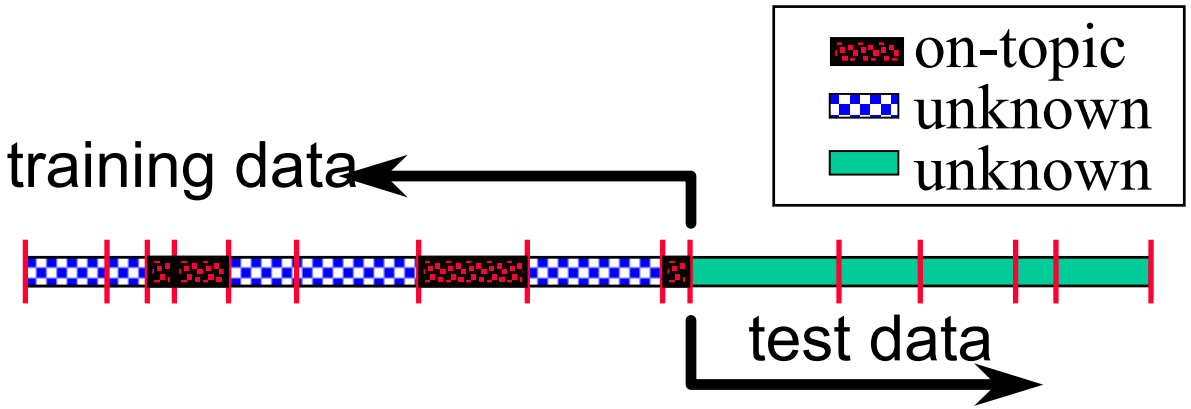
Topic Tracking

Task:

- Detect stories that discuss the target topic in multiple source streams.

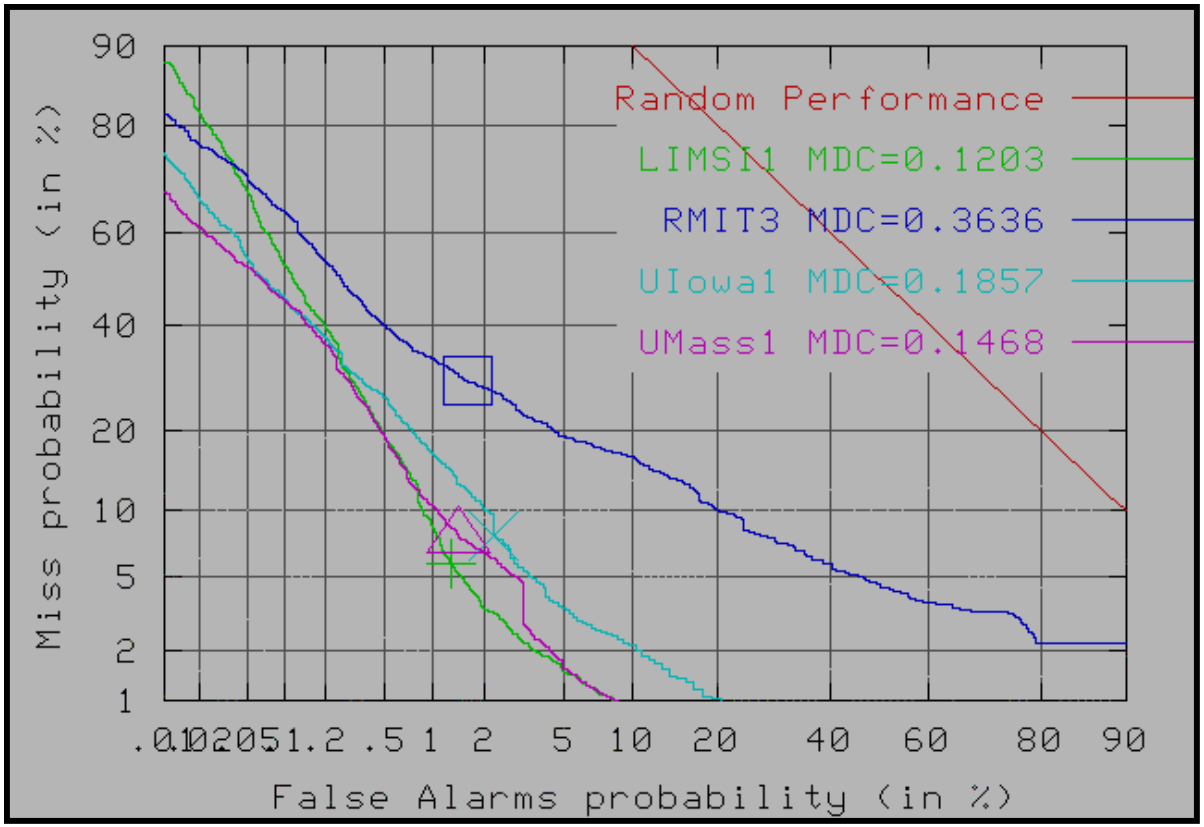
Properties:

- Training data: given N_t sample stories that discuss a given target topic
- Testing: find all subsequent stories that discuss the target topic



Based on slides by Jonathan Fiscus

Topic Tracking Results 2001



Based on slides by Jonathan Fiscus

TREC Evaluations

- **Also organized by NIST**
- **Annual evaluation since 1992**
- **Provide**
  - Annotated datasets
  - Unbiased evaluation
  - Changing tasks
    - Ad-hoc Retrieval
    - Interactive Retrieval
    - Text Filtering
    - Text Routing
    - Etc.
- **Info:** http://trec.nist.gov/

Overview: Tasks in Lang Tech

- **Topic Detection and Tracking**
- **Part-of-Speech Tagging**
- **Information Extraction**
- **Named Entity Recognition**
- **Learning Retrieval Functions**

Part-of-Speech Tagging

- **Assign the correct part of speech (word class) to each word in a document**

  "The/DT planet/NN Jupiter/NNP and/CC its/PRP moons/NNS are/VBP in/IN effect/NN a/DT mini-solar/JJ system/NN ,/, and/CC Jupiter/NNP itself/PRP is/VBZ often/RB called/VBN a/DT star/NN that/IN never/RB caught/VBN fire/NN ./."

- **Needed as an initial processing step for a number of language technology applications**
  - Answer extraction in Question Answering
  - Base step in identifying syntactic phrases for IR systems
  - Critical for word-sense disambiguation (WordNet apps)
  - Information extraction
  - …

Why is POS Tagging Difficult?

- **Ambiguity:**
  - He will race/VB the car.
  - When will the race/NOUN end?
  - The boat floated/VBD down the river bank.

  => Average of ~2 parts of speech for each word
- **Typical number of tags used by different systems beween less than 20 to more than 400.**
- **Approaches:**
  - Transformation-Based Learning [Brill, 1993]
  - Hidden Markov Models (e.g. [Altun et al. 2003])

Transformation-Based Learning [Brill, 93]

Given

- Annotated training corpus

Learning

- Label each word with most frequent tag for that word
- Greedily search for "exception" rules
- If rule applies, the predicted tag is changed

Examples

- **NN→VB if the previous tag is TO**

  I wanted to/TO win/NN→VB a Subaru WRX…

- **VBP→VB if one of the prev-3 tags is MD**

  The food might/MD vanish/VBP→VB from sight.

Empirical Results

- **Simple heuristics can go a long way**
  - ~90% accuracy by choosing the most frequent tag for a word
- **Performance**
  - Not lexicalized
    - Transformations are entirely tag-based; no specific words were used in the rules.
    - Tagger achieves 97.0% accuracy
    - Learns 378 rules
  - Lexicalized
    - Certain phrases and lexicalized expressions can yield idiosyncratic tag sequences => allow rules to look for specific words
    - Tagger achieves 97.2% accuracy
      - First 200 rules achieved 97.0%
      - First 100 rules achieved 96.8%
    - Learns 447 rules

Overview: Tasks in Lang Tech

- **Topic Detection and Tracking**
- **Part-of-Speech Tagging**
- **Information Extraction**
- **Named Entity Recognition**
- **Learning Retrieval Functions**

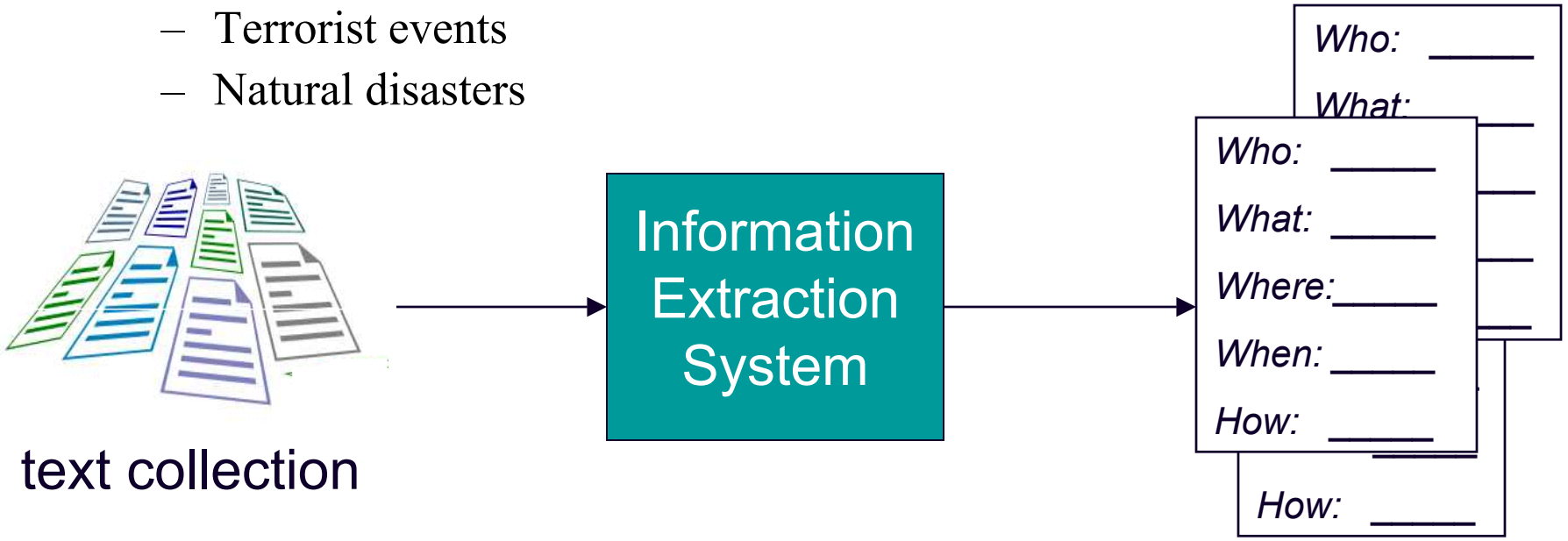
Information Extraction

Task

- Identify specific pieces of information (data) in a unstructured or semi-structured textual document.
- Transform unstructured information in a corpus of documents or web pages into a structured database.

Examples

- Job postings from WWW / Newsgroups: Flipdog, Rapier
- Terrorist events
- Natural disasters



from Claire Cardie

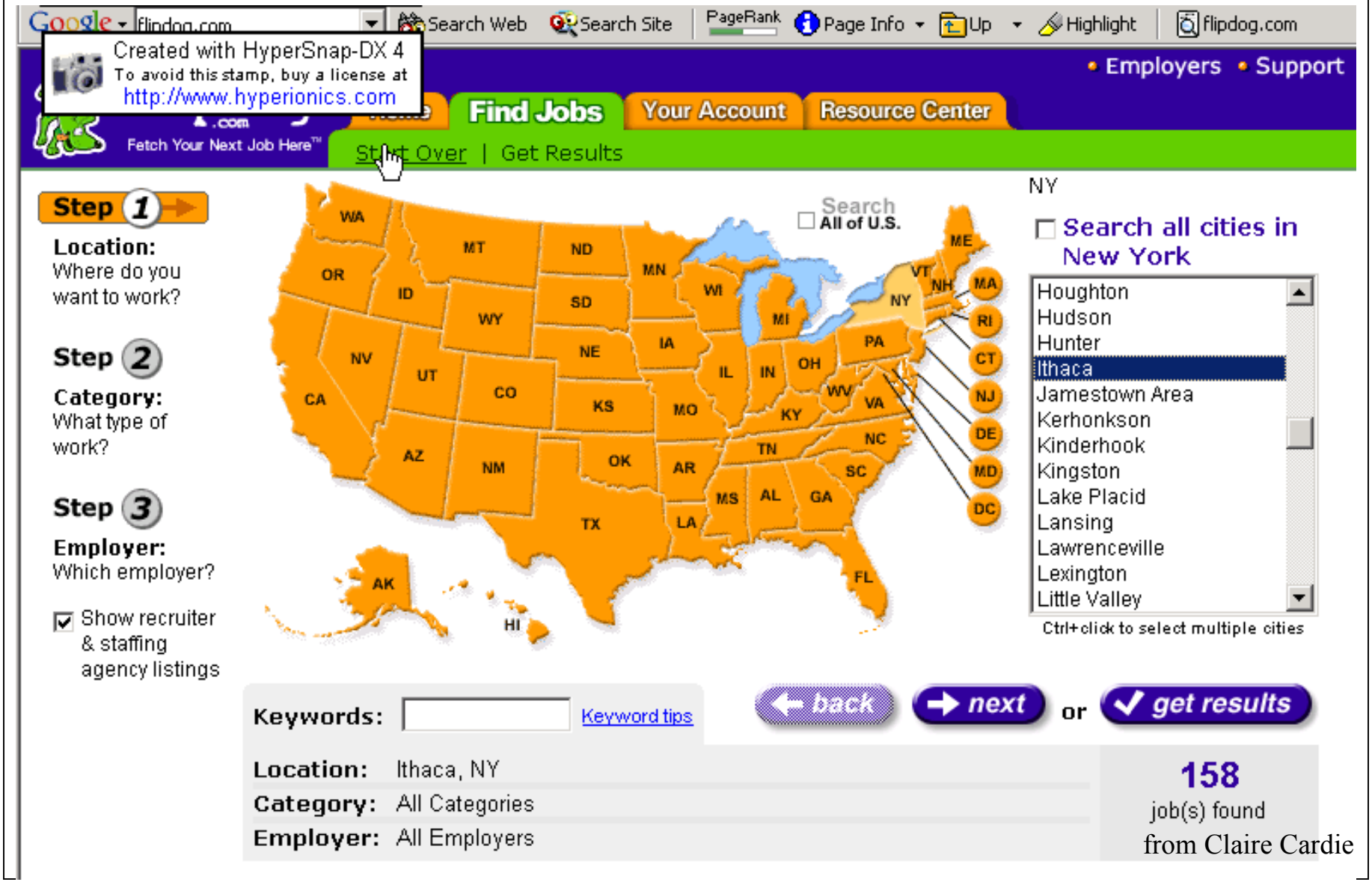
Sample Job Posting / Extraction

Subject: US-TN-SOFTWARE PROGRAMMER
Date: 17 Nov 1996 17:37:29 GMT
Organization: Reference.Com Posting Service
Message-ID: <56nigp$mrs@bilbo.reference.com>

SOFTWARE PROGRAMMER
Position available for Software Programmer experienced in generating software for PC-Based Voice Mail systems. Experienced in C Programming. Must be familiar with communicating with and controlling voice cards; preferable Dialogic, however, experience with others such as Rhetorix and Natural Microsystems is okay. Prefer 5 years or more experience with PC Based Voice Mail, but will consider as little as 2 years. Need to find a Senior level person who can come on board and pick up code with very little training. Present Operating System is DOS. May go to OS-2 or UNIX in future.
Please reply to:
Kim Anderson
AdNET
(901) 458-2888 fax
kimander@memphisonline.com

computer_science_job
id: 56nigp$mrs@bilbo.reference.com
title: SOFTWARE PROGRAMMER
salary:
company:
recruiter:
state: TN
city:
country: US
language: C
platform: PC \ DOS \ OS-2 \ UNIX
application:
area: Voice Mail
req_years_experience: 2
desired_years_experience: 5
req_degree:
desired_degree:
post_date: 17 Nov 1996

Flipdog.com



from Claire Cardie

Flipdog.com

from Claire Cardie

Approaches to Information Extraction

Goal

- Given a training set of documents paired with human-produced filled extraction templates [answer keys],
- Learn extraction patterns for each slot (different types)

Approaches

- Manual rule construction
- Extraction patterns (e.g. [Riloff, 1993] [Freitag, 1998] [Soderland, 1999])
  - <victim> was murdered
  - <perpetrator> attempted to kill
- Regular Expressions (e.g. [Califf and Mooney, 1999])
- Hidden Markov Models (e.g. [Bikel et al., 1999], [McCallum et al., 2000])

State-of-the-Art in Information Extraction

Unrestricted text:
60-70% R; 65-75% P

Semi-structured text:
90% R/P

MUC
[1991-94]

- **terrorist activities**
- **business joint ventures**
- **microelectronic chip fabrication**
- **changes in corporate management**
- **natural disasters**
- **summarize medical patient records**
- **support automatic classification of legal documents**
- **build knowledge bases from web pages**
- **create job-listing databases from newsgroups**

from Claire Cardie

Overview: Tasks in Lang Tech

- **Topic Detection and Tracking**
- **Part-of-Speech Tagging**
- **Information Extraction**
- **Named Entity Recognition**
- **Learning Retrieval Functions**

Named Entity Recognition

Task:

- Identify all named locations, named persons, named organizations, dates, times, monetary amounts, and percentages.

The delegation, which included the commander of the U.N. troops in Bosnia, Lt. Gen. Sir Michael Rose, went to the Serb stronghold of Pale, near Sarajevo, for talks with Bosnian Serb leader Radovan Karadzic.

Este ha sido el primer comentario publico del presidente Clinton respecto a la crisis de Oriente Medio desde que el secretario de Estado, Warren Christopher, decidiera regresar precipitadamente a Washington para impedir la ruptura del proceso de paz tras la violencia desatada en el sur de Libano.

1. Locations
2. Persons
3. Organizations

Figure 1.1 Examples. Examples of correct labels for English text and for Spanish text.

from Claire Cardie

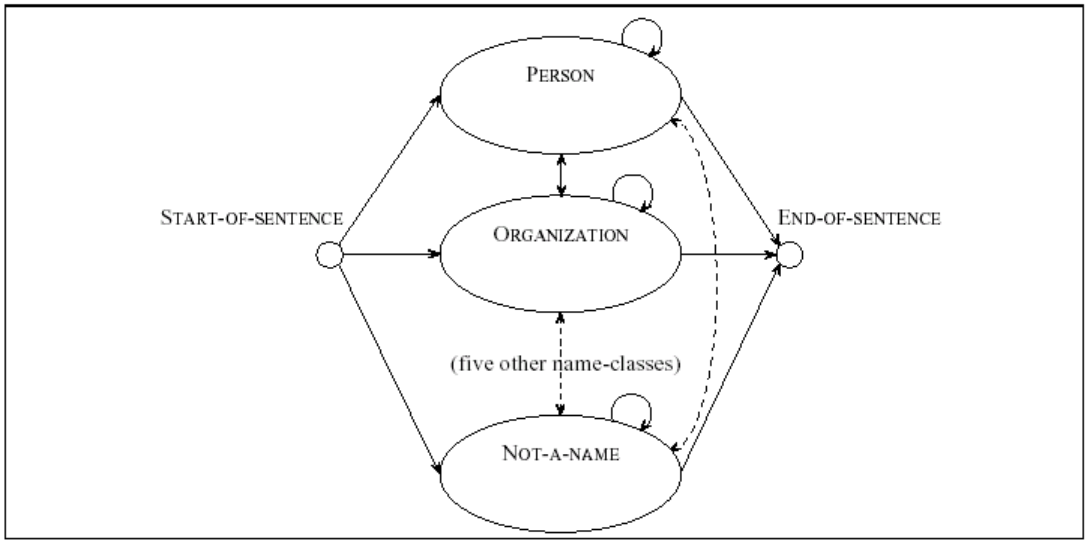
HMM for NE Recognition

Model the process that generates a sentence

- States (hidden)
  - One for each name class
  - Two special start and end states
- Links have transition probabilities between name classes
- Each state produces a word based on emission probability P(w|s)

Performance

- 90%-95% F-Score



Identifinder [Bikel et al., 1999]

Overview: Tasks in Lang Tech

- **Topic Detection and Tracking**
- **Part-of-Speech Tagging**
- **Information Extraction**
- **Named Entity Recognition**
- **Learning Retrieval Functions**

Why Learn Retrieval Functions?

Query:

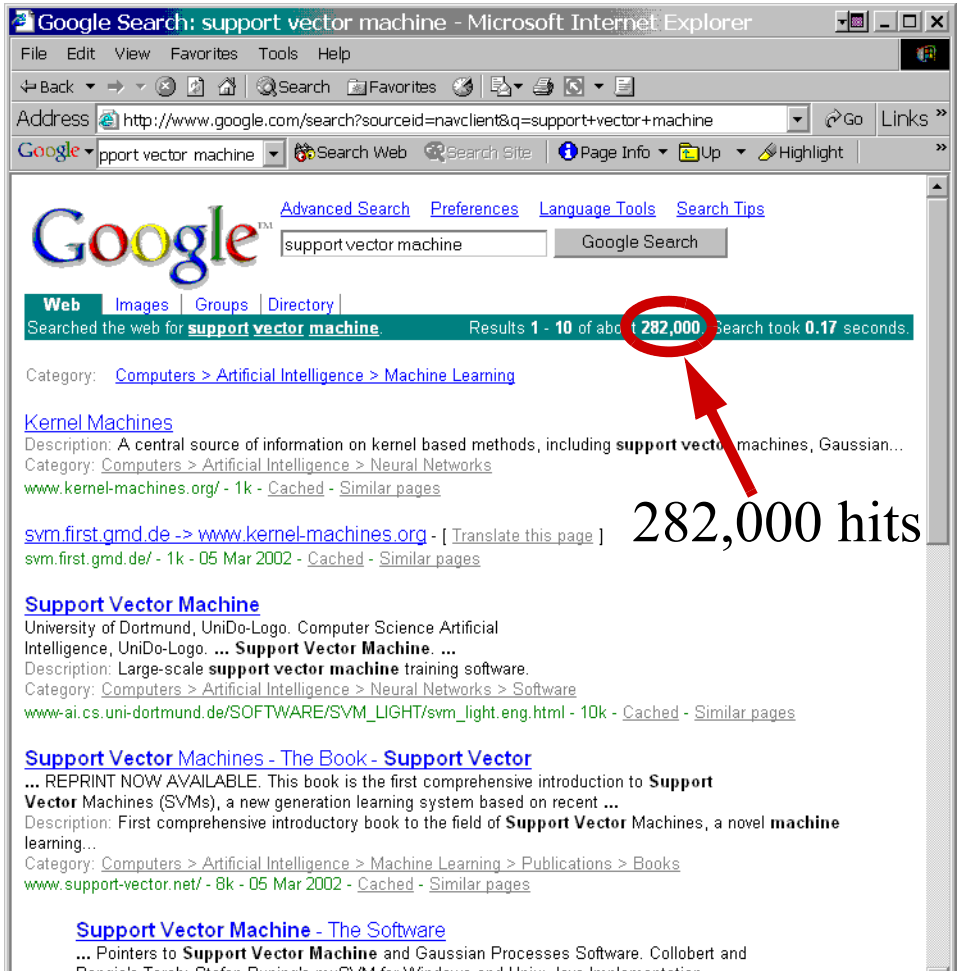
- "Support Vector Machine"

Goal:

- "rank the document I want high in the list"

Applications:

- personalizing search engines
- tuning retrieval functions in XML intranets
- recommender systems



see [Cohen et al., 1999] [Crammer & Singer, 01] [Joachims, 2002c]

Training Examples from Clickthrough

Assumption: If a user skips a link *a* and clicks on a link *b* ranked lower, then the user preference reflects *rank(b) < rank(a)*.

Example: *(3 < 2) and (7 < 2), (7 < 4), (7 < 5), (7 < 6)*

Ranking Presented to User:

1. Kernel Machines
   http://svm.first.gmd.de/
2. Support Vector Machine
   http://jbolivar.freeservers.com/
3. SVM-Light Support Vector Machine
   http://ais.gmd.de/~thorsten/svm light/
4. An Introduction to Support Vector Machines
   http://www.support-vector.net/
5. Support Vector Machine and Kernel ... References
   http://svm.research.bell-labs.com/SVMrefs.html
6. Archives of SUPPORT-VECTOR-MACHINES ...
   http://www.jiscmail.ac.uk/lists/SUPPORT...
7. Lucent Technologies: SVM demo applet
   http://svm.research.bell-labs.com/SVT/SVMsvt.html
8. Royal Holloway Support Vector Machine
   http://svm.dcs.rhbnc.ac.uk/

Learning to Rank

Assume:

- distribution of queries P(Q)
- distribution of target rankings for query P(R | Q)

Given:

- collection D of m documents
- i.i.d. training sample $(q_1, r_1), \ldots, (q_n, r_n)$

Design:

- set of ranking functions F, with elements $f: Q \rightarrow P^{D \times D}$ (weak ordering)
- loss function $l(r_a, r_b)$
- learning algorithm

Goal:

- find $f^\circ \in F$ with minimal $R_P(f) = \int l(f(q), r) dP(q, r)$

A Loss Function for Rankings

For two orderings r_a and r_b, a pair $d_i \neq d_j$ is

- *concordant*, if r_a and r_b agree in their ordering
  P = number of concordant pairs
- *discordant*, if r_a and r_b disagree in their ordering
  Q = number of discordant pairs

Loss function: [Wong et al., 88], [Cohen et al., 1999], [Crammer & Singer, 01], [Herbrich et al., 98] ...

$$l(r_a, r_b) = Q$$

Example:

$$r_a = (a, c, d, b, e, f, g, h)$$
$$r_b = (a, b, c, d, e, f, g, h)$$

=> discordant pairs *(c,b), (d,b)* => $l(r_a, r_b) = 2$

What does the Ranking Function Look Like?

Sort documents d_i by their "retrieval status value" rsv(q,d_i) with query q [Fuhr, 89]:

$$\begin{aligned}\text{rsv}(q,d_i) = \quad & w_1 * \#(\text{of query words in title of } d_i) \\ & + w_2 * \#(\text{of query words in H1 headlines of } d_i) \\ & \ldots \\ & + w_N * \text{PageRank}(d_i) \\ = \vec{w}\ & \Phi(q,d_i).\end{aligned}$$

Select F as:

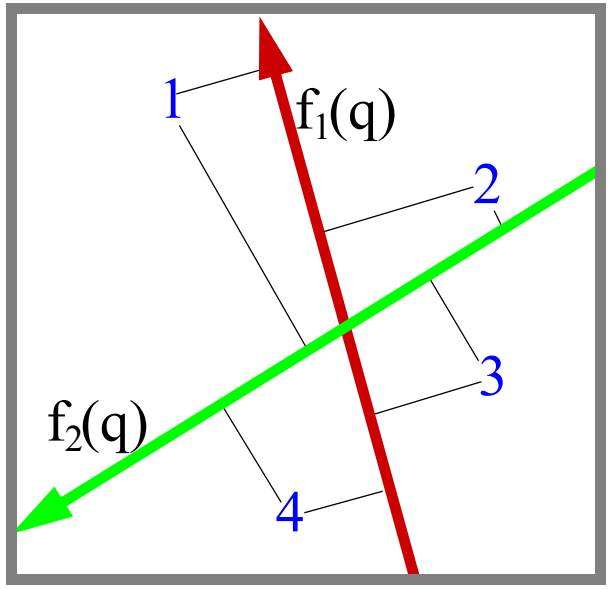
$$d_i > d_j$$

$$\Leftrightarrow$$

$$(d_i, d_j) \in f_{\vec{w}}(q)$$

$$\Leftrightarrow$$

$$\vec{w}\Phi(q, d_i) > \vec{w}\Phi(q, d_j)$$



Ranking Support Vector Machine

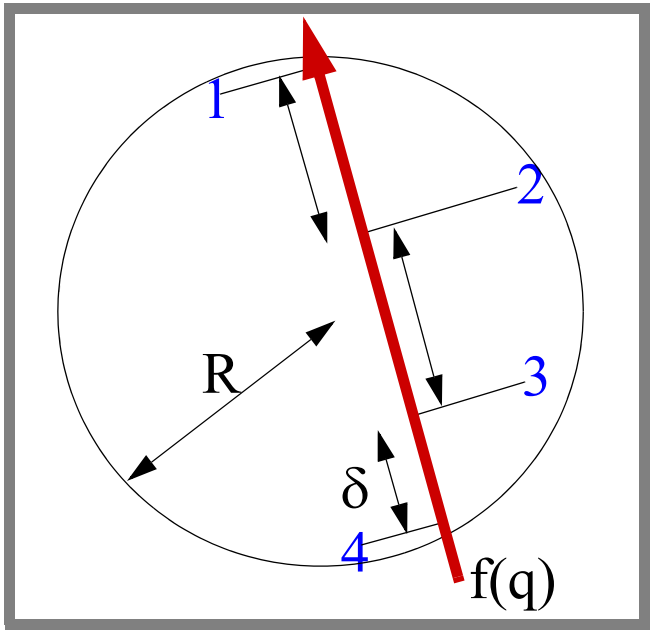
Optimization Problem (primal):

$$min \ \frac{1}{2} w \cdot w + C \sum \xi_{l,i,j}$$

$$\forall (d_i, d_j) \in r_1; (\vec{w}\Phi(q_1, d_i) \geq \vec{w}\Phi(q_1, d_j) + 1 - \xi_{1,i,j})$$
$$\dots$$
$$\forall (d_i, d_j) \in r_n; (\vec{w}\Phi(q_n, d_i) \geq \vec{w}\Phi(q_n, d_j) + 1 - \xi_{n,i,j})$$

Properties:

- minimize trade-off between training loss and margin size $\delta = 1 / \|w\|$
- quadratic program, similar to classification SVM (=> SVMlight)
- convex => unique global optimum
- radius of ball containing the training points R



How is this different from ...

... classification?

$f_1(q)$: - - + -

$f_2(q)$: - +

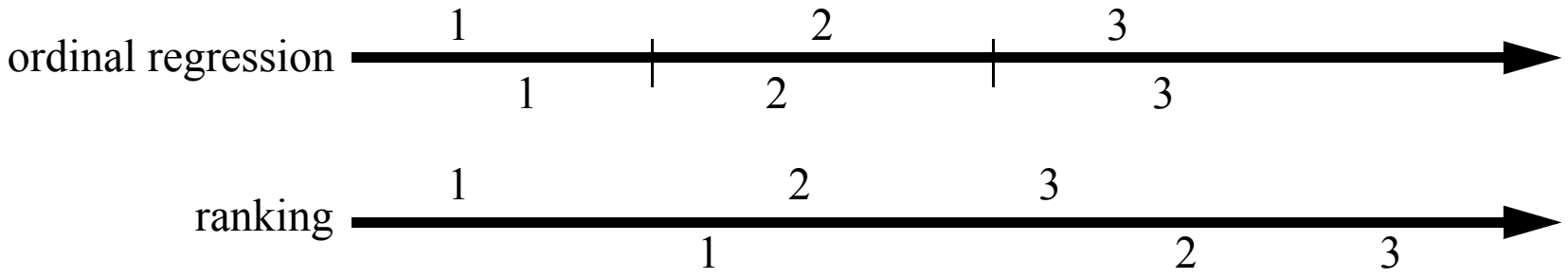
=> both have same error rate (always classify as non-relevant)

=> very different rank loss

... ordinal regression?

Training set $S = (x_1, y_1), \ldots, (x_n, y_n)$, with Y ordinal (and finite)

=> ranks need to be comparable between examples



Experiment: Learning vs. Google/MSNSearch

Ranking A	Ranking B	A better	B better	Tie	Total
Learned	Google	29	13	27	69
Learned	MSNSearch	18	4	7	29
Learned	Toprank	21	9	11	41

~20 users, as of 2nd of December

Toprank: rank by increasing mimium rank over all 5 search engines

=> **Result**: Learned > Google
Learned > MSNSearch
Learned > Toprank

Learned Weights

weight	feature
0.60	cosine between query and abstract
0.48	ranked in top 10 from Google
0.24	cosine between query and the words in the URL
0.24	document was ranked at rank 1 by exactly one of the 5 search engines
...	
0.17	country code of URL is ".de"
0.16	ranked top 1 by HotBot
...	
-0.15	country code of URL is ".fi"
-0.17	length of URL in characters
-0.32	not ranked in top 10 by any of the 5 search engines
-0.38	not ranked top 1 by any of the 5 search engines

Reading

- **Information Retrieval Concepts**: [Baeza-Yates & Ribeiro-Neto, 99]
  Modern Information Retrieval.
- **Information Retrieval Systems**: [Witten et al., 1999]
  Managing Gigabytes: Compressing and Indexing Documents and Images.
- **Text Classification**: [Joachims, 2002]
  Learning to Classify Text Using Support Vector Machines.
- **Natural Language Processing**: [Manning and Schuetze, 1999]
  Foundations of Statistical Natural Language Processing.
- **Support Vector Machines**: [Schölkopf & Smola, 2002]
  Learning with Kernels.
- **Software**: [Joachims, 1999c] `http://svmlight.joachims.org/`
  SVM^{light} for Classification, Regression, Ranking, and Sequence Alignment

Bibliography

- [Altun et al, 2003] Y. Altun, I. Tsochantaridis, and T. Hofmann (2003). Hidden Markov Support Vector Machines, International Conference on Machine Learning (ICML).
- [Baeza-Yates and Ribeiro-Neto, 1999] Baeza-Yates, R. and Ribeiro-Neto, B. (1999). Modern Information Retrieval. Addison-Wesley-Longman, Harlow, UK.
- [Ben-Hur et al., 2001] Ben-Hur, A., Horn, D., Siegelmann, H., Vapnik, V. (2001). Support Vector Clustering. Journal of Machine Learning Research, 2:125-137.
- [Bikel et al, 1999] D. Bikel, R. Schwartz and R. Weischedel (1999). An Algorithm that Learns What's in a Name. Machine Learning Journal, Special Issue on Natural Language Learning.
- [Boser et al., 1992] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A traininig algorithm for optimal margin classifiers. In

Haussler, D., editor, Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, pages 144-152.

- [Brill, 1993] E. Brill (1993). A corpus-based approach to language learning. Ph.D. Dissertation, Department of Computer and Information Science, University of Pennsylvania.
- [Burges, 1998] C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. Knowledge Discovery and Data Mining, 2(2), 1998.
- [Califf and Mooney, 1999] M. Califf and R. Mooney (1999). Relational Learning of Pattern-Match Rules for Information Extraction. Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99), Orlando, FL, pp. 328-334.
- [Chapelle et al., 2002] Chapelle, O., V. Vapnik, O. Bousquet and S. Mukherjee (2002). Choosing Multiple Parameters for Support Vector Machines. Machine Learning 46(1), pages 131-159.

- [Cohen et al., 1999] W. Cohen, R. Shapire, and Y. Singer (1999). Learning to order things. Journal of Artificial Intelligence Research, 10:243-270.
- [Collins & Duffy, 2001] Collins, M., and Duffy, N. (2001). Convolution kernels for natural language. Proceedings of Advances in Neural Information Processing Systems (NIPS).
- [Cortes & Vapnik, 1995] Cortes, C. and Vapnik, V. N. (1995). Support-vector networks. Machine Learning Journal, 20:273-297.
- [Crammer & Singer, 2001] K. Crammer and Y. Singer (2001). On the algorithmic implementation of multiclass kernel-based vector machines, Journal of Machine Learning Research, 2:265-292.
- [Cristianini & Shawe-Taylor, 2000] Cristianini, N. and Shawe-Taylor, J. (2000). An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press.
- [Cristianini et al., 2002] N. Cristianini, J. Shawe-Taylor, and H. Lodhi (2002). Latent Semantic Kernels. Journal of Intelligent Information Systems, 18(2-3):127-152.

- [Deerwester et al., 1990] Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the AmericanSociety of Information Science, 41(6):391–407.
- [Dumais et al., 1998] Dumais, S., Platt, J., Heckerman, D., and Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In Proceedings of ACM-CIKM98.
- [Freitag, 1998] D. Freitag (1998). Machine Learning for Information Extraction in Informal Domains. PhD. dissertation, Carnegie Mellon University, November, 1998.
- [Herbrich et al., 2000] R. Herbrich, T. Graepel, K. Obermayer (2000). Large Margin Rank Boundaries for Ordinal Regression, In Advances in Large Margin Classifiers, pages 115-132, MIT Press.
- [Jaakkola, and Haussler, 1998] Jaakkola, T. and Haussler, D. (1998). Exploiting generative models in discriminative classifiers. In Advances in Neural Information Processing Systems 11, MIT Press.

- [Jaakkola and Haussler, 1999] Jaakkola, T. and Haussler, D. (1999). Probabilistic kernel regression models. In Conference on AI and Statistics.
- [Joachims, 1998] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In Proceedings of the European Conference on Machine Learning, pages 137 - 142, Berlin. Springer.
- [Joachims, 1999b] Joachims, T. (1999b). Making Large-Scale SVM Learning Practical. In Schölkopf, B., Burges, C., and Smola, A., editors, Advances in Kernel Methods - Support Vector Learning, pages 169 - 184. MIT Press, Cambridge.
- [Joachims, 1999c] Joachims, T. (1999c) Transductive Inference for Text Classification using Support Vector Machines. International Conference on Machine Learning (ICML).
- [Joachims, 2002] Joachims, T. (2002). Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms, Kluwer.

- [Joachims, 2002c] Joachims, T. (2002). Optimizing Search Engines using Clickthrough Data, Conference on Knowledge Discovery and Data Mining (KDD), ACM.
- [Leopold & Kindermann, 2002] E. Leopold, J. Kindermann (2002). Text Categorization with Support Vector Machines. How to Represent Texts in Input Space? Machine Learning 46(1-3):423-444.
- [Lodhi et al., 2002] Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., & Watkins, C. (2002). Text Classification using String Kernels. Journal of Machine Learning Research, 2:419-444.
- [Manevitz & Yousef, 2001] L. Manevitz and M. Yousef (2001). One-Class SVMs for Document Classification, Journal of Machine Learning Research, 2:139-154.
- [Manning and Schuetze, 1999] C. Manning and H. Schuetze (1999). Foundations of Statistical Natural Language Processing. MIT Press.
- [McCallum et al, 2000] A. McCallum, D. Freitag, and F. Perreira (2000). Maximum Entropy Markov Models for Information

Extraction and Segmentation. International Conference on Machine Learning (ICML).

- [Platt et al., 2000] Platt, J., Cristianini, N., and Shawe-Taylor, J. (2000). Large margin dags for multiclass classification. In Advances in Neural Information Processing Systems 12.
- [Riloff, 1993] E. Riloff (1993). Automatically Constructing a Dictionary for Information Extraction Tasks. Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93) , AAAI Press/The MIT Press, pp. 811-816.
- [Salton and Buckley, 1988] Salton, G. and Buckley, C. (1988). Term weighting approaches in automatic text retrieval. Information Processing and Management, 24(5):513–523.
- [Schölkopf et al., 1995] Schölkopf, B., Burges, C., Vapnik, V. (1995). Extracting support data for a given task. Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD).

- [Schölkopf et al., 1998] Schölkopf, B., Smola, A., and Mueller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, 10:1299-1319.
- [Schölkopf et al., 2000] Schölkopf, B., R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt (2000). Support vector method for novelty detection. In S. Solla, T. Leen, and K.-R. Müller (Eds.), Advances in Neural Information Processing Systems 12, pp. 582-588. MIT Press.
- [Schölkopf et al., 2001] Schölkopf, B., J. Platt, J. Shawe-Taylor, A.J. Smola and R.C. Williamson (2001). Estimating the support of a high-dimensional distribution. Neural Computation 13(7);1443-1471.
- [Schölkopf & Smola, 2002] Schölkopf, B., Smola, A. (2002). Learning with Kernels, MIT Press.
- [Siolas & d'Alche-Buc, 2000] G. Siolas and F. d'Alché-Buc (2000). Support vectors machines based on a semantic kernel for Text Categorization, International Conference on Artificial Neural Networks (ICANN).

- [Smola & Schölkopf, 1998] A. J. Smola and B. Schölkopf. A Tutorial on Support Vector Regression. NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College of London, UK, 1998.
- [Soderland, 1999] S. Soderland (1999). Learning Information Extraction Rules for Semi-Structured and Free Text. Machine Learning 34(1-3):233-272.
- [Tax & Duin, 1999] D. Tax and R. Duin (1999). Support vector domain description. Patter Recognition Letters, 20:1991-1999.
- [Tax & Duin, 2001] D. Tax and R. Duin (2001). Uniform Object Generation for Optimizing One-Class Classifiers, Journal of Machine Learning Research, 2:155-173.
- [Vapnik, 1998] Vapnik, V. (1998). Statistical Learning Theory. Wiley, Chichester, GB.
- [Vapnik & Chapelle, 2000] V. Vapnik and O. Chapelle. Bounds on Error Expectation for Support Vector Machines. In Neural Computation, 2000, vol 12, 9.

- [Weston & Watkins, 1998] Weston, J. and Watkins, C. (1998). Multi-class support vector machines. Technical Report CSD-TR-98-04, Royal Holloway University of London.
- [Witten et al., 1999] I. Witten, A. Moffat, and T. Bell (1999). Managing Gigabytes: Compressing and Indexing Documents and Images. 2nd edition, Morgan Kaufmann.