



# COMP3420: Advanced Databases and Data Mining

Text data mining

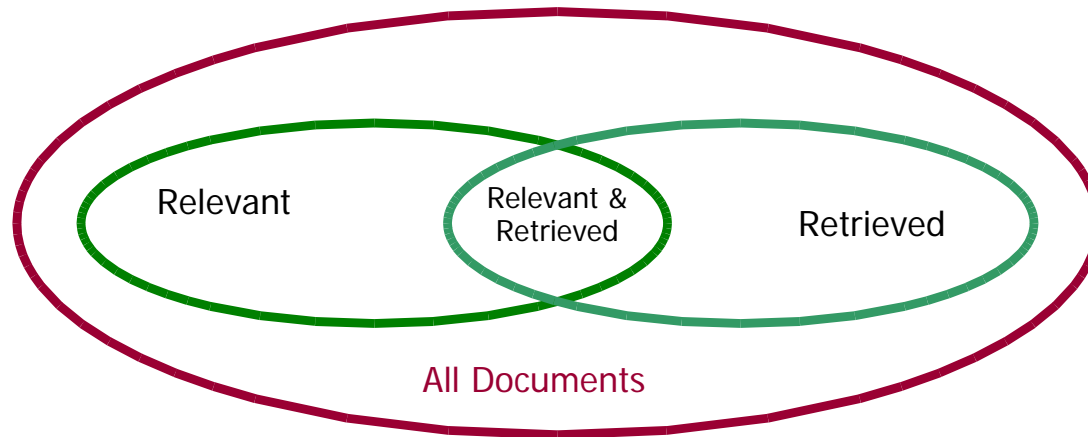
# Lecture outline

- Text data analysis and text/information retrieval
  - Basic measures for text/information retrieval
  - Information retrieval techniques
  - Boolean and vector space model
  - Similarity-based retrieval in text data
  - TF-IDF weighting
  - Vector space model
- Types of text data mining
  - Keyword based association analysis
  - Text classification and categorisation
  - Document clustering

# Text data analysis and information retrieval

- Typical information retrieval systems
  - Online library catalogs
  - Online document management systems
  - Internet search engines
- Information retrieval (IR) versus database (DB) systems
  - Some DB problems are not present in IR, such as: updates, transaction management, complex structured objects
  - Some IR problems are not addressed well in DBMS, for example: unstructured documents, approximate search using keywords and relevance

# Basic measures for text retrieval



- Precision: the percentage of retrieved documents that are in fact relevant to the query (i.e., the “correct” responses)

$$precision = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Retrieved\}|}$$

- Recall: the percentage of documents that are relevant to the query and were, in fact, retrieved

$$recall = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Relevant\}|}$$

# Information retrieval techniques

- Basic concepts

- A document can be described by a set of representative keywords called *index terms*
- Different index terms have varying relevance when used to describe document contents
- This effect is captured through the *assignment of numerical weights* to each index term of a document (for example, frequency, or TF-IDF)

- DBMS analogy

- Index terms → Attributes
- Weights → Attribute values

# Information retrieval techniques (2)

- Index terms (attribute) selection
  - Stop word list
  - Word stem
  - Index terms weighting methods
- Term and document frequency matrices
- Information retrieval models
  - Boolean model
  - Vector model
  - Probabilistic model (categories modeled by probability distributions, find likelihood a document belongs to a certain category, similar to Bayesian classification)

# Boolean model

- Consider that index terms are either present or absent in a document
  - For example: *1=present, 0=absent*
  - As a result, the index term weights are assumed to be all binaries
- A query is composed of index terms linked by three connectives: *not, and, and or*
  - For example: *“car and repair”, “plane or airplane”*
- The Boolean model predicts that each document is either relevant or non-relevant based on the match of a document to the query

# Similarity-based retrieval in text data

- Finds similar documents based on a set of common keywords
- Answer should be based on the degree of relevance based on the nearness of the keywords, relative frequency of the keywords, etc.
- Use of stop lists
  - Set of words that are deemed “irrelevant”, even though they may appear frequently
  - For example: *a, the, of, for, to, with*, etc.
  - Stop lists may vary when document set varies

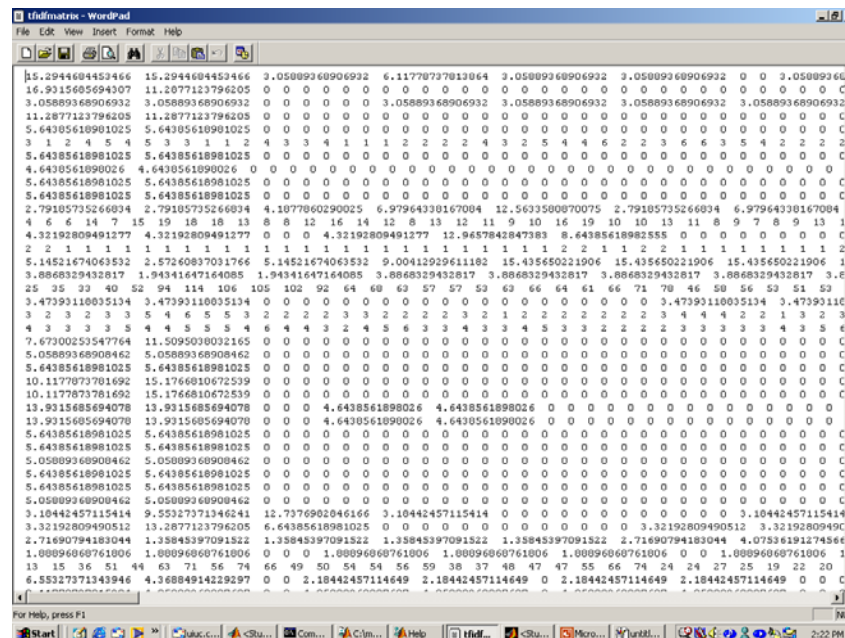
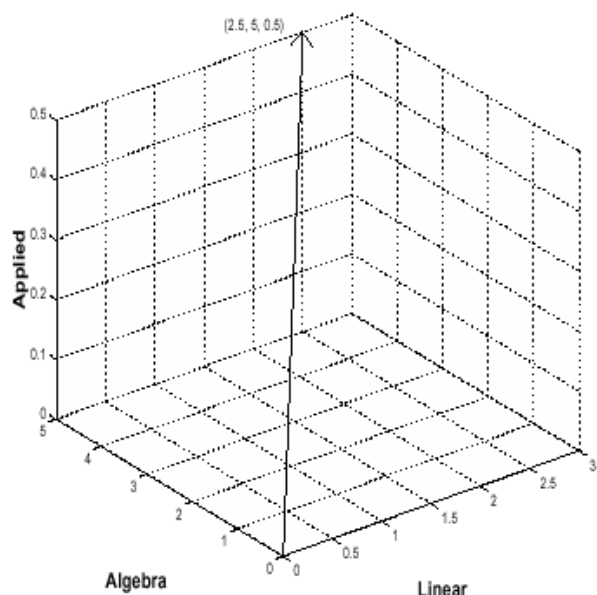
# Similarity-based retrieval in text data (2)

- Apply word stemming
  - Several words are small syntactic variants of each other since they share a common word stem
  - For example, *drug*, *drugs*, *drugged* → *drug*
- A term and document frequency matrix (or table)
  - Each entry  $frequent\_table(i, j)$  = number of occurrences of the word  $t_i$  in document  $d_j$
  - Usually, the *ratio* instead of the absolute number of occurrences is used
- Similarity metrics: measure the closeness of a document to a query (a set of keywords)
  - Relative term occurrences
  - Cosine distance:

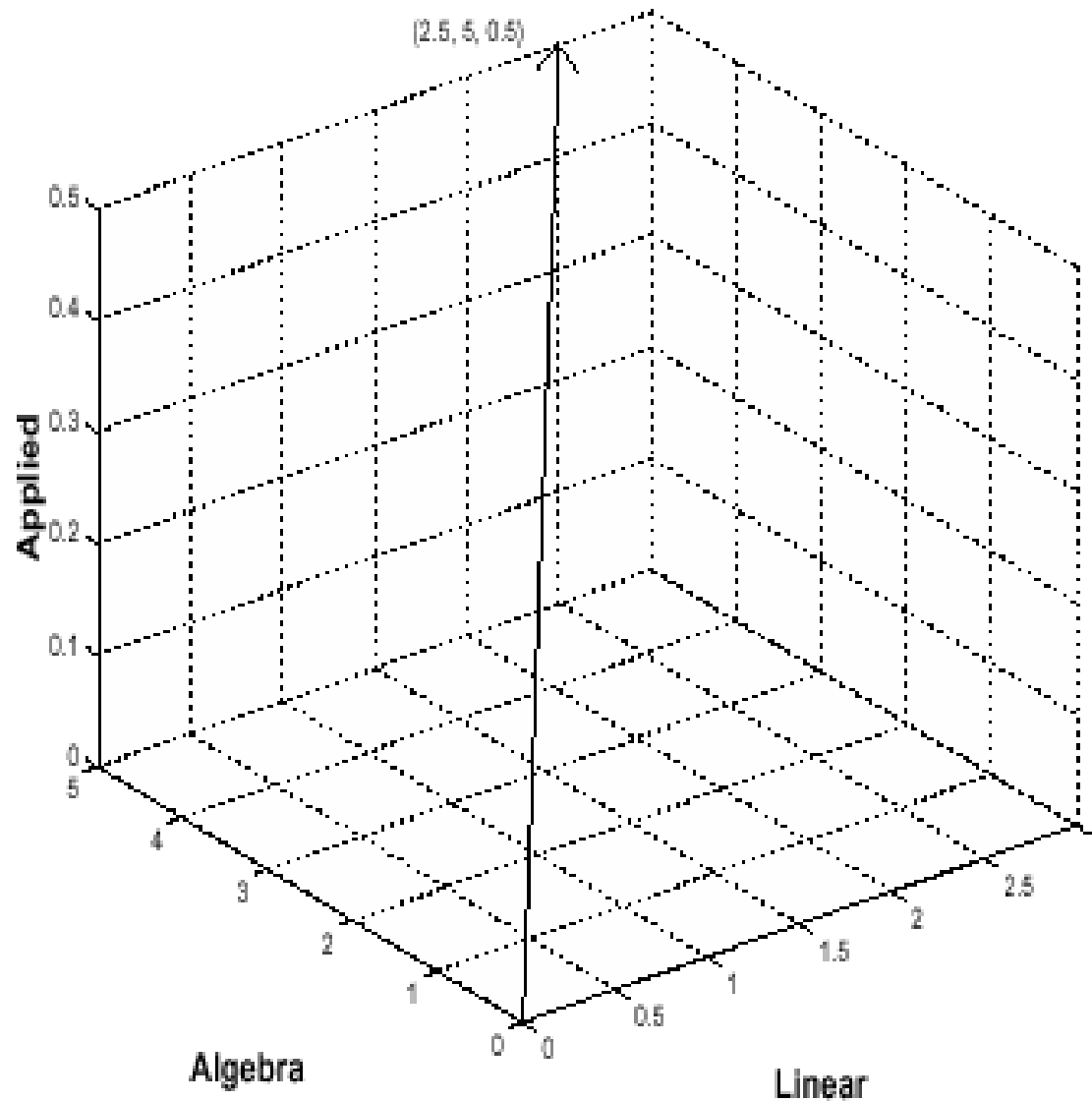
$$sim(v_1, v_2) = \frac{v_1 \cdot v_2}{|v_1| |v_2|}$$

# Vector space model

- Documents and user queries are represented as  $m$ -dimensional vectors, where  $m$  is the total number of index terms in the document collection
- The degree of similarity of the document  $d$  with regard to the query  $q$  is calculated as the correlation between the vectors that represent them, using measures such as the *Euclidean distance* or the *cosine* of the angle between these two vectors



# Vector space model (2)





# Vector space model (4)

- Represent a document by a *term or feature vector*
  - Term: basic concept, for example, *word* or *phrase* (like “data mining”)
  - Each term defines one dimension (large number of dimensions!)
  - $N$  terms define a  $N$ -dimensional space
  - Element of vector corresponds to term weight
  - For example,  $d = (x_1, \dots, x_N)$ ,  $x_i$  is “importance” of term  $i$
  - These term vectors are *sparse* (most weights are 0)
- New document is assigned to the most likely category based on vector similarity

# How to assign weights

- Two-fold heuristics based on frequency
- TF (Term Frequency)
  - More frequent *within* a document → more relevant to semantics
  - For example, “classification” versus “SVM”
  - Raw TF =  $f(t, d)$  (how many times term  $t$  appears in doc  $d$ )
  - Document length varies => relative frequency preferred
  - Perform normalisation (for example, *maximum frequency normalisation*)

$$TF(t, d) = 0.5 + \frac{0.5 * f(t, d)}{MaxFreq(d)}$$

- IDF (Inverse Document Frequency)

- Less frequent *among* documents → more discriminative
- For example “algebra” versus “science”
- Formula:
  - $n$  = total number of documents
  - $k$  = number of documents with term  $t$  appearing

$$IDF(t) = 1 + \log\left(\frac{n}{k}\right)$$

# TF-IDF weighting

- TF-IDF weighting:  $weight(t, d) = TF(t, d) * IDF(t)$ 
  - Frequent within doc  $\rightarrow$  high TF  $\rightarrow$  high weight
  - Selective among docs  $\rightarrow$  high IDF  $\rightarrow$  high weight
- Recall vector space model
  - Each selected term represents one dimension
  - Each document is represented by a *term* or *feature vector*
  - Its *t*-term coordinate of document *d* is the TF-IDF weight
- Just for illustration ...
  - Many complex and more effective weighting variants exist in practice

# How to measure similarity?

- Given two document

$$D_i = (w_{i1}, w_{i2}, \dots, w_{iN}) \quad D_j = (w_{j1}, w_{j2}, \dots, w_{jN})$$

- Similarity definition

- Dot product

$$Sim(D_i, D_j) = \sum_{t=1}^N w_{it} * w_{jt}$$

Normalised dot product (or *cosine similarity*)

$$Sim(D_i, D_j) = \frac{\sum_{t=1}^N w_{it} * w_{jt}}{\sqrt{\sum_{t=1}^N (w_{it})^2 * \sum_{t=1}^N (w_{jt})^2}}$$

# Illustrative example

doc1  
 text  
 mining  
 search  
 engine  
 text

$$Sim(newdoc, doc1) = 4.8 * 2.4 + 4.5 * 4.5$$

$$Sim(newdoc, doc2) = 2.4 * 2.4$$

doc2  
 travel  
 text  
 map  
 travel

$$Sim(newdoc, doc3) = 0$$

doc3  
 government  
 president  
 congress

To whom is *newdoc* more similar?  
 (it contains the words *text* and *mining*)

	text	mining	travel	map	search	engine	govern	president	congress
IDF(faked)	2.4	4.5	2.8	3.3	2.1	5.4	2.2	3.2	4.3
doc1	2(4.8)	1(4.5)			1(2.1)	1(5.4)			
doc2	1(2.4)		2(5.6)	1(3.3)					
doc3							1(2.2)	1(3.2)	1(4.3)
newdoc	1(2.4)	1(4.5)							

.....

# Vector space model-based classifiers

- What do we have so far?
  - A feature space with similarity measure
  - This is a classic supervised learning problem
  - Search for an approximation to classification hyper plane
- Vector space model based classifiers
  - Decision tree based
  - Neural networks
  - Support vector machine
  - ...

# Types of text data mining

- Keyword-based association analysis
- Automatic document classification
- Similarity detection
  - Cluster documents by a common author
  - Cluster documents containing information from a common source
- Link analysis: unusual correlation between entities
- Sequence analysis: predicting a recurring event
- Anomaly detection: find information that violates usual patterns
- Hypertext analysis
  - Patterns in anchors/links (for example, anchor text correlations with linked objects)
- Applications: news article classification, automatic e-mail filtering, Web page classification, etc.

# Keyword-based association analysis

- Motivation
  - Collect sets of keywords or terms that occur frequently together and then find the *association* or *correlation* relationships among them
- Association analysis process
  - Pre-process the text data by parsing, stemming, removing stop words, etc.
  - Evoke association mining algorithms
    - Consider each document as a transaction
    - View a set of keywords in the document as a set of items in the transaction
  - Term level association mining
    - No need for human effort in tagging documents
    - The number of meaningless results and the execution time is greatly reduced

# Text classification

- **Motivation**

- Automatic classification for the large number of on-line text documents (Web pages, e-mails, corporate intranet documents, etc.)

- **Classification process**

- Data pre-processing
- Definition of training set and test sets
- Creation of the classification model using the selected classification algorithm
- Classification model validation
- Classification of new/unknown text documents

- **Text document classification differs from the classification of relational data**

- Document databases are not structured according to attribute-value pairs

# Text classification (2)

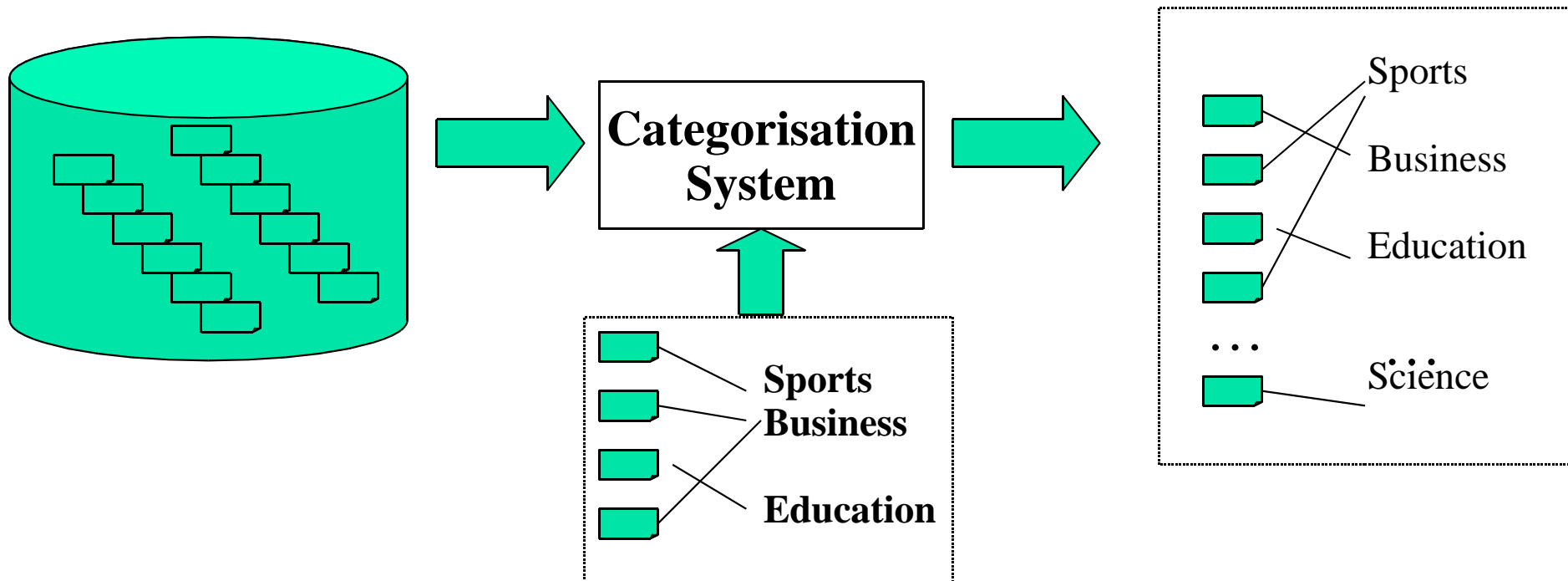
## • Classification Algorithms

- Support vector machines
- K-nearest neighbors
- Naïve Bayes
- Neural networks
- Decision trees
- Association rule-based
- Boosting
- ...

			#1	#2	#3	#4	#5
		# of documents	21,450	14,347	13,272	12,902	12,902
		# of training documents	14,704	10,667	9,610	9,603	9,603
		# of test documents	6,746	3,680	3,662	3,299	3,299
		# of categories	135	93	92	90	10
System	Type	Results reported by					
WORD	(non-learning)	[Yang 1999]	.150	.310	.290		
	probabilistic	[Dumais et al. 1998]				.752	.815
	probabilistic	[Joachims 1998]					.720
PROPBAYES	probabilistic	[Lam et al. 1997]	.443 ( $MF_1$ )				
BIM	probabilistic	[Lewis 1992a]	.650				
	probabilistic	[Li and Yamanishi 1999]				.747	
NB	probabilistic	[Li and Yamanishi 1999]				.773	
	probabilistic	[Yang and Liu 1999]				.795	
C4.5	decision trees	[Dumais et al. 1998]					.884
IND	decision trees	[Joachims 1998]					.794
	decision trees	[Lewis and Ringuette 1994]	.670				
SWAP-1	decision rules	[Apté et al. 1994]		.805			
RIPPER	decision rules	[Cohen and Singer 1999]	.683	.811		.820	
SLEEPING EXPERTS	decision rules	[Cohen and Singer 1999]	.753	.759		.827	
DL-ESC	decision rules	[Li and Yamanishi 1999]				.820	
CHARADE	decision rules	[Moulinier and Ganascia 1996]		.738			
CHARADE	decision rules	[Moulinier et al. 1996]		.783 ( $F_1$ )			
LSP	regression	[Yang 1999]		.855	.810		
LSP	regression	[Yang and Liu 1999]				.849	
BALANCED-WINNOW	on-line linear	[Dagan et al. 1997]	.747 (M)	.833 (M)			
WIDROW-HOFF	on-line linear	[Lam and Ho 1998]				.822	
ROCCIO	batch linear	[Cohen and Singer 1999]	.660	.748		.776	
FINESIM	batch linear	[Dumais et al. 1998]				.617	.646
ROCCIO	batch linear	[Joachims 1998]					.799
ROCCIO	batch linear	[Lam and Ho 1998]				.781	
ROCCIO	batch linear	[Li and Yamanishi 1999]				.625	
CLASSI	neural network	[Ng et al. 1997]		.802			
NNET	neural network	[Yang and Liu 1999]				.838	
	neural network	[Wiener et al. 1995]			.820		
GIS-W	example-based	[Lam and Ho 1998]				.860	
k-NN	example-based	[Joachims 1998]					.823
k-NN	example-based	[Lam and Ho 1998]				.820	
k-NN	example-based	[Yang 1999]	.690	.852	.820		
k-NN	example-based	[Yang and Liu 1999]				.856	
SVMLIGHT	SVM	[Dumais et al. 1998]				.870	.920
SVMLIGHT	SVM	[Joachims 1998]					.864
SVMLIGHT	SVM	[Li and Yamanishi 1999]				.841	
SVMLIGHT	SVM	[Yang and Liu 1999]				.859	
ADABOOST.MH	committee	[Schapire and Singer 2000]		.860			
	committee	[Weiss et al. 1999]				.878	
	Bayesian net	[Dumais et al. 1998]				.800	.850
	Bayesian net	[Lam et al. 1997]	.542 ( $MF_1$ )				

# Text classification (3)

- Pre-given classes (categories) and labeled document examples (categories may form hierarchy)
- Classify new documents
- A standard classification (supervised learning ) problem



# Categorisation methods

- **Manual: Typically rule-based**
  - Does not scale up (labor-intensive, rule inconsistency)
  - May be appropriate for special data on a particular domain
- **Automatic: Typically exploiting machine learning techniques**
  - Vector space model based
    - Prototype-based (Rocchio)
    - K-nearest neighbor (KNN)
    - Decision-tree (learn rules)
    - Neural networks (learn non-linear classifier)
    - Support vector machines (SVM)
  - Probabilistic or generative model based
    - Naïve Bayes classifier

# Document clustering

- Motivation

- Automatically group related documents based on their contents
- No predetermined training sets or taxonomies
- Generate a taxonomy at runtime

- Clustering process

- Data preprocessing: remove stop words, stem, feature extraction, lexical analysis, etc.
- Hierarchical clustering: compute similarities applying clustering algorithms
- Model-based clustering (neural network approach): clusters are represented by “exemplars” (for example Self-Organising Maps, SOM)