Chapter 2: Background

Natural language text

Natural language text

- Written and spoken
- Important means of communication:
  - natural language:
    - finite set of words and of syntactic/discourse constructs
    - -> enormous amount of possible combinations: signal the many different meanings (semantics) expressed in text
- In this course focus upon:
  - text in West European languages
  - mostly upon written text
  - **text content**

SWARM INTELLIGENCE

Following a trail of insects as they work together to accomplish a task offers unique possibilities for problem solving.

By Peter Tarasewich & Patrick R. McMullen

Even with today's ever-increasing computing power, there are still many types of problems that are very difficult to solve. Particularly combinatorial optimization problems continue to pose challenges. An example of this type of problem can be found in product design. Take as an example the design of an automobile based on the attributes of engine horsepower, passenger seating, body style and wheel size. If we have three different levels for each of these attributes, there are 3^4, or 81, possible configurations to consider. For a slightly larger problem with 5 attributes of 4 levels, there are suddenly 1,024 combinations. Typically, an enormous amount of possible combinations exist, even for relatively small problems. Finding the optimal solution to these problems is usually impractical. Fortunately, search heuristics have been developed to find good solutions to these problems in a reasonable amount of time.

Over the past decade or so, several heuristic techniques have been developed that build upon observations of processes in the physical and biological sciences. Examples of these techniques include Genetic Algorithms (GA) and simulated annealing…

Description of text

1. Micro level: **sentence and clause**
    1. phonemes and letters
    2. syllables and morphemes
    3. words
    4. phrases
    5. clauses
    6. sentences
2. Macro level: **discourse**
    1. schematic structure
    2. rhetorical structure
    3. thematic structure

Description of text

3. Text as part of a **digital document**
4. Text as part of a **document collection**
5. Text as part of a **community of people**

Text described at micro level

- At the sentence level and below: text has own syntax and semantics

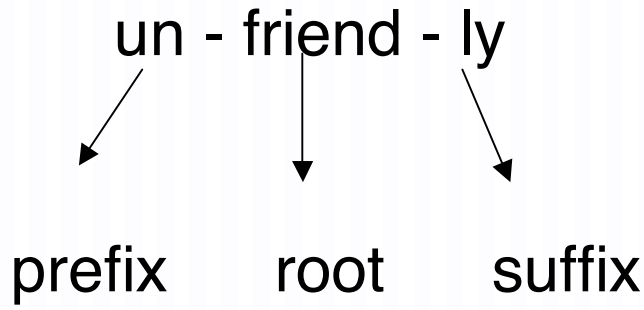
Phonemes and letters

- **Phoneme** = the smallest unit of speech that distinguishes one utterance from another

- **Letter** = character in written text, originally one letter represented one phoneme, but this has evolved

Morphemes

= the components of words



- **Root or stem**: free morpheme: can occur in isolation, and cannot be divided in smaller meaning units
- Affix (**prefix** or **suffix**) = bound morpheme
  - derivational: modifies part-of-speech of base word
  - inflectional: does not modify part-of-speech of base word, but signals changes in number, person, gender and tense, etc.
- Relation with syllables in speech

Words

- **Part-of-speech** (POS) or syntactical word class:
  - contributes to the meaning of the word in a phrase
  - distinct component in syntactic structure

  - **content words:** nouns, adjectives, verbs, adverbs
  - **function words**: have functional properties in syntactic structure: act as determiners, quantifiers, prepositions and connectives (e.g., articles, pronouns, particles, …)

Words

- **Lexical meaning**:
  - defined by origin and usage of the word
  - stored in dictionaries
  - most important difficulties when processing text:

    1) **Homonymy** (one word has different unrelated meanings) (e.g., “bark”) and **polysemy** (one word has different related meanings) (e.g., “opening”)

    2) **Synonymy**: different words have the same meaning (e.g., “vermin” and “pests”)

    3) **Hypernymy** and **hyponymy**: words are referred to by a more general term (hypernym) or more specific term (hyponym) (e.g., “fruit” is hypernym of “apple”, “apple” is a hyponym of “fruit”)

Words

4) Phoric references: a word needs another phrasing in the text for its interpretation:

- **anaphoric reference** (refers to word(s) earlier in the text) (e.g., " The student buys the book and gives **it** to **his** sister.")
- **cataphoric reference** (refers to word(s) further in the text) (e.g., "When **she** came by, the girl looked worried.")

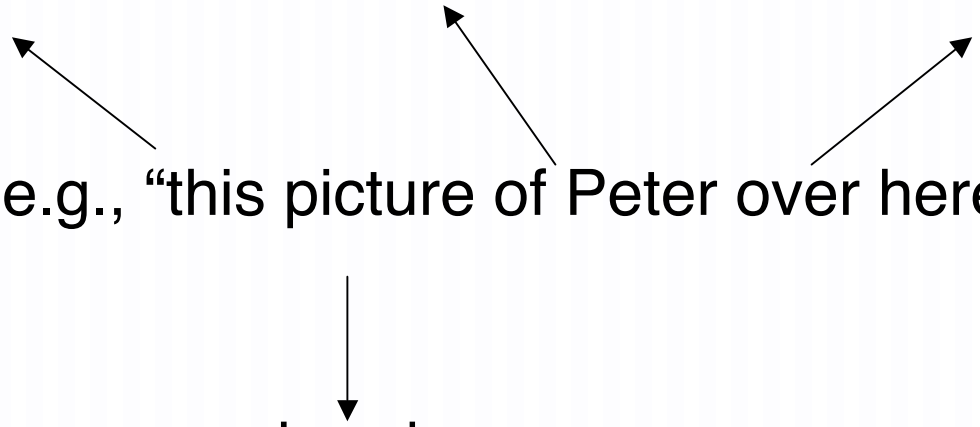
5) Coreferences: words have equivalence relationship in the described situation (e.g., "Bill Clinton", "the former president", "he")

6) New words, changes in the meaning of words, metaphorical use of words

Phrases

- Phrase:
  - combination of words according to syntactical rules
  - consists of head word and words that specify the **head** word:
    - according to the location in the phrase:

prehead modifier, posthead modifier and complement



Phrases

- Classes: base form possibly with modifiers and compliments
  - **noun** phrase: base form or its referent = noun or pronoun (e.g., "modern information retrieval", "Los Angeles", "she")
  - **adjective** phrase: base form = adjective (e.g., "dangerous to drive")
  - **verb** phrase: base form = verb (e.g., "gave the sentence to the accused without hesitation")
  - **adverb** phrase: base form = adverb (e.g., " too quickly")
  - special case: **prepositional** phrase: base form = preposition followed by noun phrase (e.g., "from the court")

Sentences

- General structure of a sentence:
  - composed of a **topic** and of **additions** (comment, focus) to the topic
  - coded according to the grammatical rules of the language
- **Syntactic structure**: sentence = composed of phrase classes that combine in regular ways

  Example: “The judge buried the case”

  <S> ::= <NP> <VP>

  <NP> ::= <ART> <NOUN>

  <VP> ::= <VERB> <NP>
- Representation of content of sentence = **proposition** (predicate followed by arguments): (BURY JUDGE CASE)

Clauses

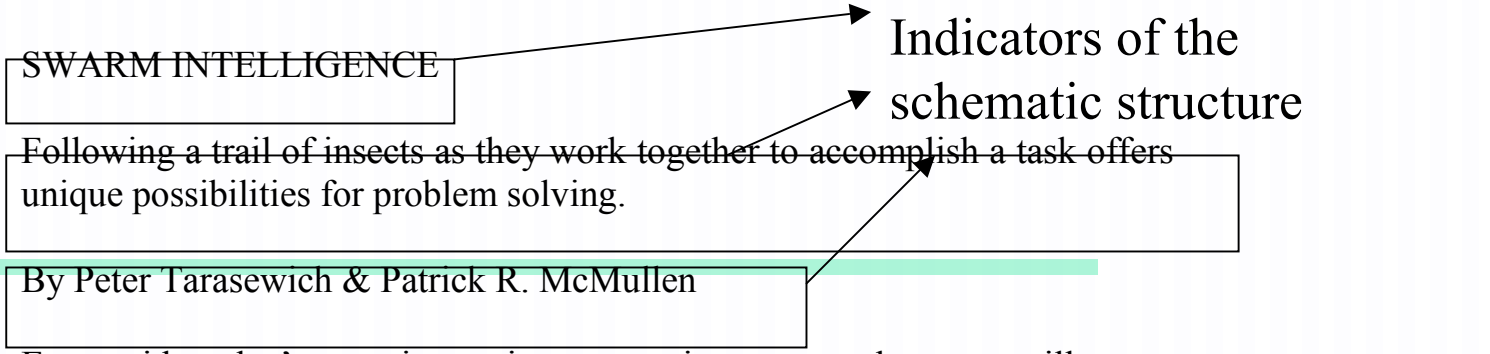
- Clause = smaller sentence that is a component of a complex sentence:
  - embedded sentence as noun phrases: e.g., “To go to jail ...”
  - relative clause of noun phrases: e.g., “who …”
- Main clauses generally foreground topics, whereas subordinate clauses generally background them

Text described at a macro level

- At the discourse level: text has own syntax and semantics
- Text structures:
  - guaranty **coherence** = configuration of concepts and relations that underlie the surface text in order to form a global mental representation of the text
  - signaled by surface organizational patterns that connect the elements of a text into a whole = **cohesion**

Text structures

- **Schematic structure** = superstructure
  - ordering of segments typical of text type
- **Rhetorical structure**
  - text type independent relationships between sentences and clauses
- **Thematic structure**:
  - organization of topics and types of thematic progression



SWARM INTELLIGENCE

Following a trail of insects as they work together to accomplish a task offers unique possibilities for problem solving.

By Peter Tarasewich & Patrick R. McMullen

Even with today's ever-increasing computing power, there are still many types of problems that are very difficult to solve. Particularly, combinatorial optimization problems continue to pose challenges. An example of this type of problem can be found in product design. Take as an example the design of an automobile based on the attributes of engine horsepower, passenger seating, body style and wheel size. If we have three different levels for each of these attributes, there are 3^4, or 81, possible configurations to consider. For a slightly larger problem with 5 attributes of 4 levels, there are suddenly 1,024 combinations. Typically, an enormous amount of possible combinations exist, even for relatively small problems. Finding the optimal solution to these problems is usually impractical. Fortunately, search heuristics have been developed to find good solutions to these problems in a reasonable amount of time.

Over the past decade or so, several heuristic techniques have been developed that build upon observations of processes in the physical and biological sciences. Examples of these techniques include Genetic Algorithms (GA) and simulated annealing…

———— : Underlined words: indicators of rhetorical structure

Text structures	Signaling cues
Schematic structure	Ordering of text segments
	Cue phrases
Rhetorical structure	Ordering of text segments
	Cue phrases
	Pronoun and reference use
	Tense and aspect of verbs
	Marks
Thematic structure	Locational cues
	Cue phrases
	Content terms

Text length

- Often computed as number of (different) content words

Text as part of a digital document

- Often stored in document format with **markups**, e.g, HyperText Markup Language (HTML), Extensible Markup Language (XML)
- Markups:
  - logical structure (e.g., headings, sections), layout
  - other metadata (e.g., author, date, content)
  - hypertext links
  - hypermedia links
  - references to applets, scripts
  - …

Text as part of a document collection

- Collection or corpus can exhibit different:
  - languages
  - text types
  - subject domains
  - sizes
  - media
- Texts in the collection can be explicitly (e.g., by means of hypertext links) or implicitly (e.g., threads in e-mail conversation) **linked** !

Text as part of a community of people

- Content:
  - is left implicit by writer
  - is not always well-formed (e.g., spam mail, blogs)
  - might be complemented with textual tags, bookmarks, comments and other media (images, audio,...)
- But, inferred by reader

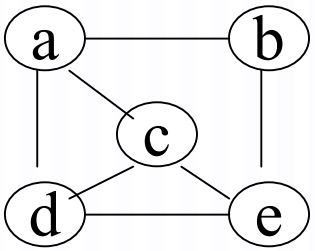
Link and Graph Based Algorithms

Motivation

- A sentence, text, text collection, ... can be **represented** as a graph: e.g.,
  - parse tree of a sentence
  - entities in a text are linked by means of coreferent or other referent relations (e.g., subject-object, head-modifier)
  - hyperlinks in a document collection
  - ...
- Graphs allow **inferences**, walks, ...: e.g., exploited in retrieval models, text analysis
- Information retrieval and natural language processing increasingly rely on graph and link based algorithms

Graph

- A graph $G = (V,E)$ is a data structure composed of:
  - V = set of **vertices** or **nodes**
  - E = set of **edges** connecting the vertices in V
- An edge $e = (u,v)$ is a pair of vertices

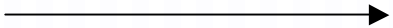


$V = \{a,b,c,d,e\}$

$E = \{(a,b),(a,c),(a,d),(b,e),(c,d),(c,e),(d,e)\}$

Graph

- **Undirected graph**: pair of vertices in an edge is unordered
- **Directed graph**: pair of vertices in each edge is ordered:

  tail  head

- **Degree** of *v*: number of edges incident to *v*
  - Incident: given edge $e = (v_i, v_j)$
    - *e* is incident on v_i and v_j
- **In-degree** of *v*: number of edges with *v* as head
- **Out-degree** of *v*: number of edges with *v* as tail

Graph

- In **weighted graph** $G = (V, E, W)$: edges have a weight associated
- **Adjacency matrix** A of G with n vertices: n x n array:

  IF edge(v_i,v_j), A[i,j] = 1 or weight

  ELSE A[i,j] = 0
- **Path** = sequence of vertices, $v_1, v_2, \ldots, v_k$ such that consecutive vertices, v_i and v_i+1, are adjacent
- **Sink node** = vertex with no outgoing link

Graph-based centrality measures

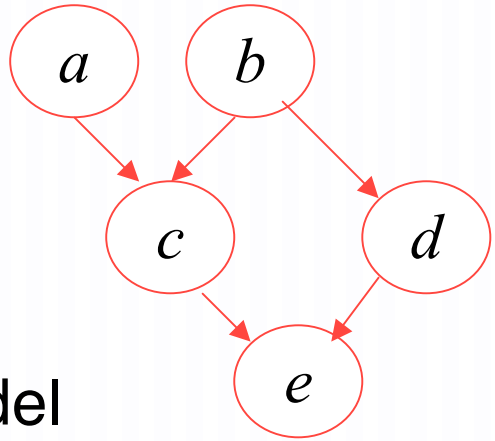
- **Centrality (*C*(*V*))** = measure of importance of a vertix in a (directed or undirected) graph
- Definitions of *C*(*V*):
  - Degree centrality: number of *V*'s neighbors
  - Distance centrality: average distance of *V* to all other nodes
  - Betweenness centrality: how many paths include *V*
  - Eigenvector centrality: how likely is a **random walk** on the graph to end in *V*

Algorithms and applications

- **Node rank algorithms**: the most popular is PageRank (Google) based on eigenvector centrality
  - a node's centrality is defined as a degree-weighted average of the centralities of its neighbors
  - used in:
    - link-based retrieval models [see Web information retrieval]
    - summarization [see Text summarization]
- Others: minimum spanning trees (e.g., used in (term) clustering), min-cut/max-flow algorithm (used in (sentiment) classification), graph matching algorithms (e.g., used in question answering), ...

Probabilistic graphical models

- Node represents a random variable (or group of random variables)
- Edge represents probabilistic relationships between these variables
- **Bayesian network:**
  - directed graphical model



- **Markov random field:**
  - undirected graphical model

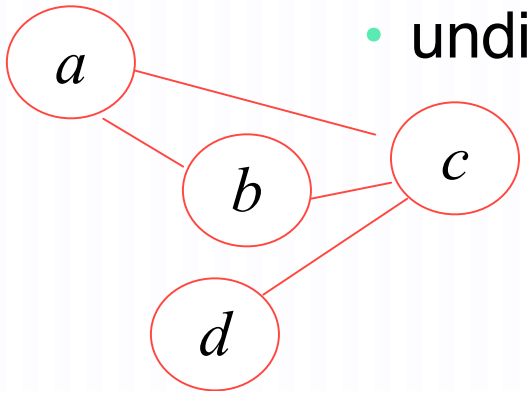
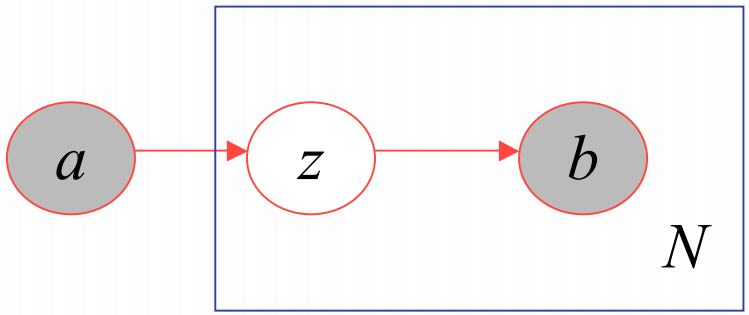


Plate notation

- Plate:
  - graphical notation that allows multiple nodes to be expressed more compactly
  - represents *N* nodes of which only a single example is shown



- shaded node: observed variable
- non-shaded node: hidden variable
- directed edge: conditional dependency of head node on tail node

Statistical and Machine Learning Techniques

Classification

- Classification tasks in text retrieval
- Classification schemes
- Types of learning techniques
  - supervised leaning methods
  - unsupervised learning methods
  - weakly supervised learning methods
- Feature selection and extraction
- Overview of the techniques and their main applications

Classification

- Classification (also known as pattern recognition)
  - Aims at classifying data (patterns) based on:
    - knowledge that is acquired by human experts
    - knowledge that is automatically learned from data
- (Pattern) **classifier**: a system that automatically sorts patterns into classes

Classification tasks in text retrieval

- **Text classification**: putting data items into groups
- Natural language processing attempts more sophisticated interpretations of text …,

  but classifications is often all we can do reliably
- We will emphasize classification based on content
- Other forms of text classification (some of them use the same techniques):

  - author identification
  - language identification, …

Classification tasks in text retrieval

1. **Text retrieval**: distinguishes documents that are relevant to a user query from non-relevant ones
2. **Text filtering/routing**: sends incoming documents to appropriate destination based on content
3. **Text categorization**: assigns one or more labels of a predefined set of semantic categories to documents
4. **Information extraction**: assigns one or more labels of a predefined set of semantic categories to information units (for text: to terms, clauses, sentences …)
5. **Textual discovery**: discovery of groupings, relationships and other patterns in text data
   - e.g., document clustering, term clustering, automated hypertext construction

Classification scheme

- Classification labels are arranged in classification scheme:
  - list
  - hierarchy
  - binary scheme
  - **ontology**

Automation of text classification tasks

- In the past: text classification was done by humans:
  - librarians, indexers
- Today:
  - Assignment of **metadata** to texts is still mostly manually done
- But, people are (often) expensive, slow and inconsistent
- Considerable interest in automating classification
    - particularly for high-volume data of variable quality: Web pages, personal e-mail, etc.
    - aids human classification

Automation of text classification tasks

- Text classification tasks need a large amount of knowledge:
  - linguistic knowledge of the vocabulary, syntax and semantics of the language and the discourse
  - knowledge of the subject domains
  - background knowledge of the person who uses the texts at a certain moment in time

Automation of text classification tasks

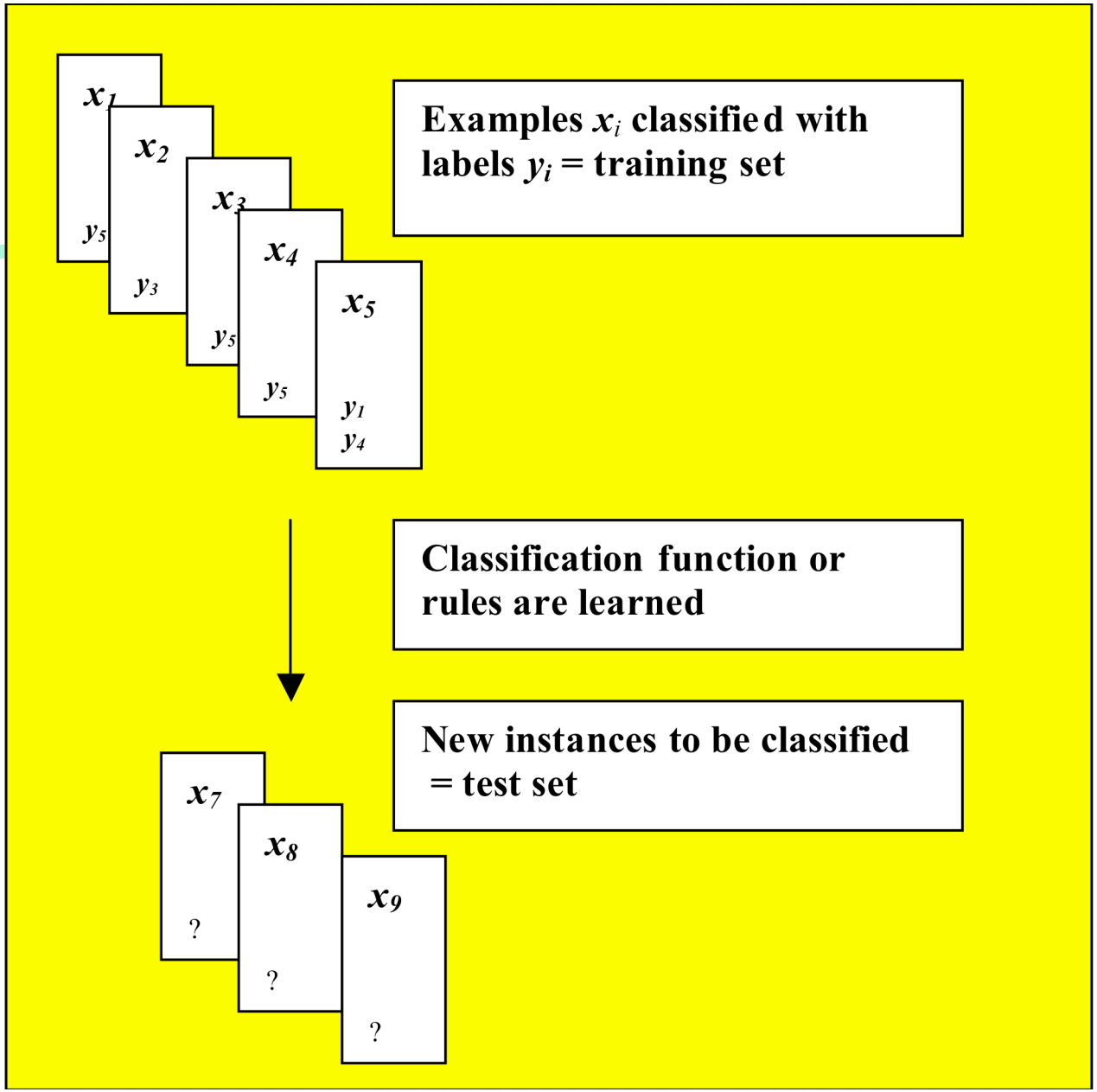
- Approaches: pattern recognition
  - **symbolic techniques**: knowledge or part of it:
    - formally implemented
  - **statistical and machine learning techniques**: knowledge or part of it:
    - automatically acquired

Supervised learning

- **Techniques of supervised learning***:*
  - **training set**: example objects classified by an expert or teacher
  - detection of general, but high-accuracy classification patterns (function or rules) in the training set based on object features and their values
  - patterns are predictable to correctly classify new, previously unseen objects in a **test set** considering their features and feature values

Supervised learning

- Text classification can be seen as a:
  - two-class learning problem:
    - an object is classified as belonging or not belonging to a particular class
    - convenient when the classes are not mutually exclusive
  - single multi-class learning problem
- Result = often **probability** of belonging to a class, rather than simply a classification



Generative versus discriminative classification

- In classification: given inputs $\boldsymbol{x}$ and their labels y:
  - **Generative classifier** learns a model of the joint probability $p(\boldsymbol{x},y)$ and makes its predictions by using Bayes' rule to calculate $p(y|\boldsymbol{x})$ and then selects the most likely label y: e.g.,
    - Naive Bayes, hidden Markov model
  - **Discriminative classifier** is trained to model the conditional probability $p(y|\boldsymbol{x})$ directly and selects the most likely label y, or learns a direct map from inputs $\boldsymbol{x}$ to the class labels: e.g.,
    - Maximum entropy model, support vector machine

Maximum entropy principle

- Text classifiers are often trained with **incomplete information**
- Probabilistic classification can adhere to the principle of maximum entropy: When we make inferences based on incomplete information, we should draw them from that probability distribution that has the maximum entropy permitted by the information we have: e.g.,
  - maximum entropy model, conditional random fields

Context-dependent classification

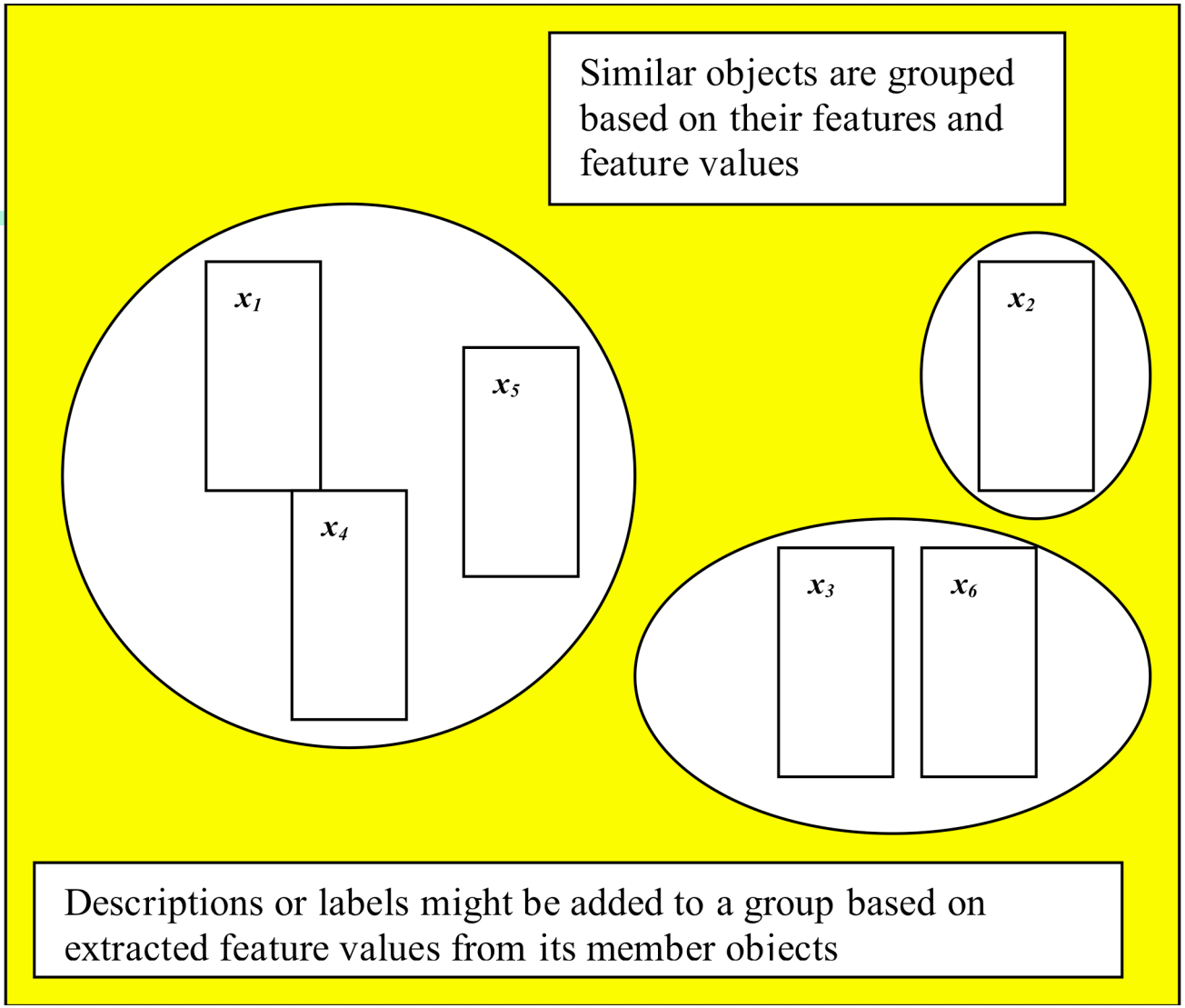
- When there exist a relation between various classes: it is valuable not to classify an object separately from other objects
- **Context-dependent classification***:* the class to which a feature vector is assigned depends on:
  - the object itself
  - other objects
  - the existing relation among the various classes
    - e.g., hidden Markov model, conditional random fields

Combining classifiers

- Multiple classifiers are learned and combined:
  - **Bagging**: e.g., sample of size *n* is taken randomly with replacement from the original set of *n* training documents
    - Adaptive resampling: no random sampling, but objective = to increase the odds of sampling documents that previously induced classifiers have erroneously been classified
  - **Stacking**: predictions from different classifiers are used as input for a meta-learner
  - **Boosting**: generating a sequence of classifiers, after each classification greater weights are assigned to objects with an uncertain classification, and classifier is retrained

Unsupervised learning

- **Techniques of unsupervised learning** :
  - natural groupings of similar objects are sought based on object features and their values
  - often use of simple hard and fuzzy clustering techniques



Weakly supervised learning

- Techniques of **weakly supervised learning**
  - supervised learning starting from a limited set of classified **seed** objects
  - exploit knowledge from set of unlabeled examples
  - often iterative learning until results on validation set cannot anymore be improved

Weakly supervised learning

- **Self-training and co-training**:
  - Starts with a set of labeled data and trains 1 (self-training) or more (co-training) classifiers, but in case of co-training each classifier trains with a disjoint (usually conditionally independent) subset of features
  - Repeat: the classifiers are applied on a set of unlabeled examples
    - when the labeling confidence (by the different classifiers) of an example exceeds a certain threshold, it is added to the training set (usually class distribution is maintained)
    - the classifiers are retrained
  - Until the classifiers reach a certain level of accuracy on a test set

Self-training

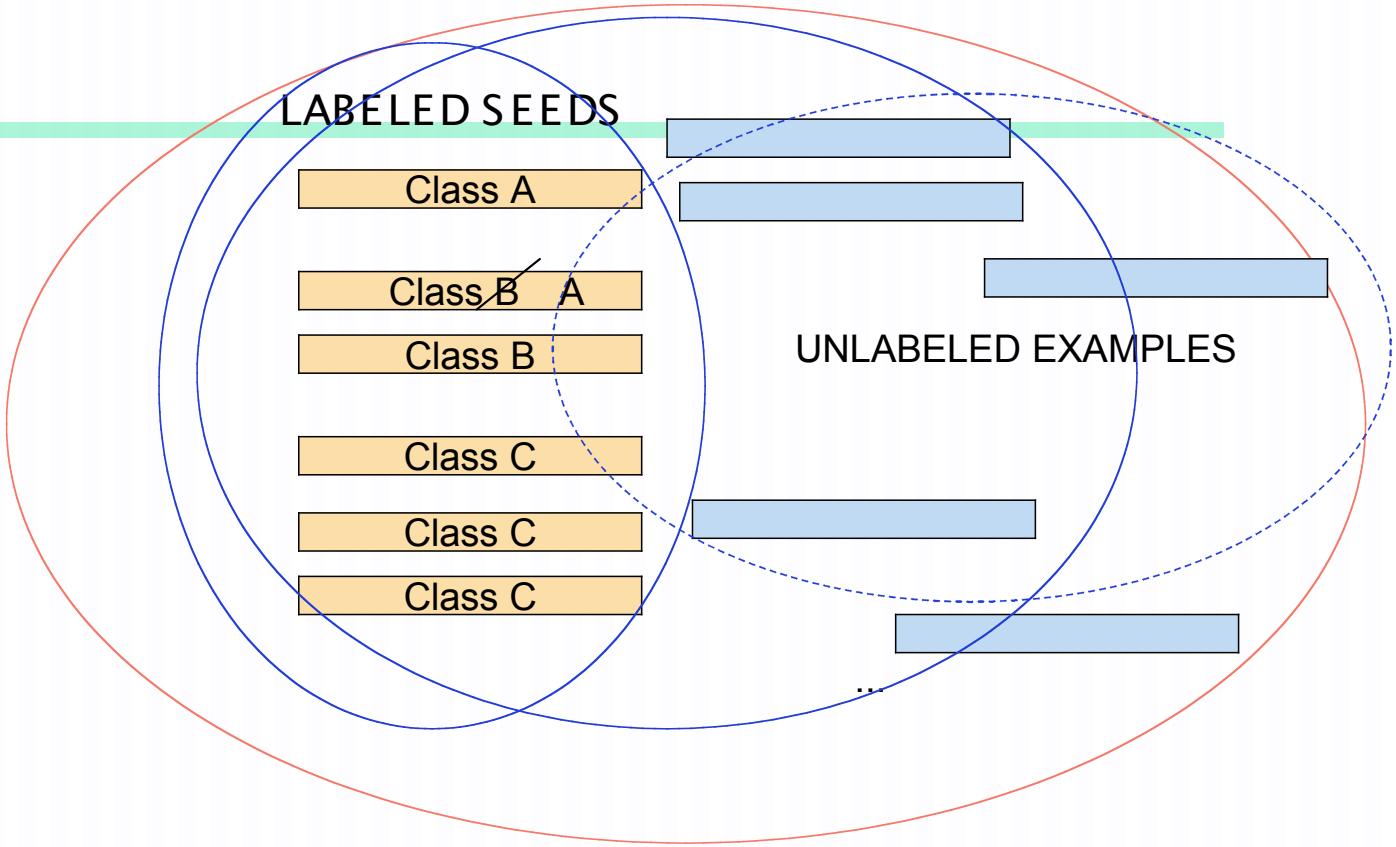


Fig. 6.3. Self-training: A classifier is incrementally trained (blue line), first based on the labeled seeds, and then based on the labeled seeds and a set of unlabeled examples that are labeled with the current classifier. The dotted blue line represents the set of all unlabeled examples that were considered for labeling in this step.

Co-training

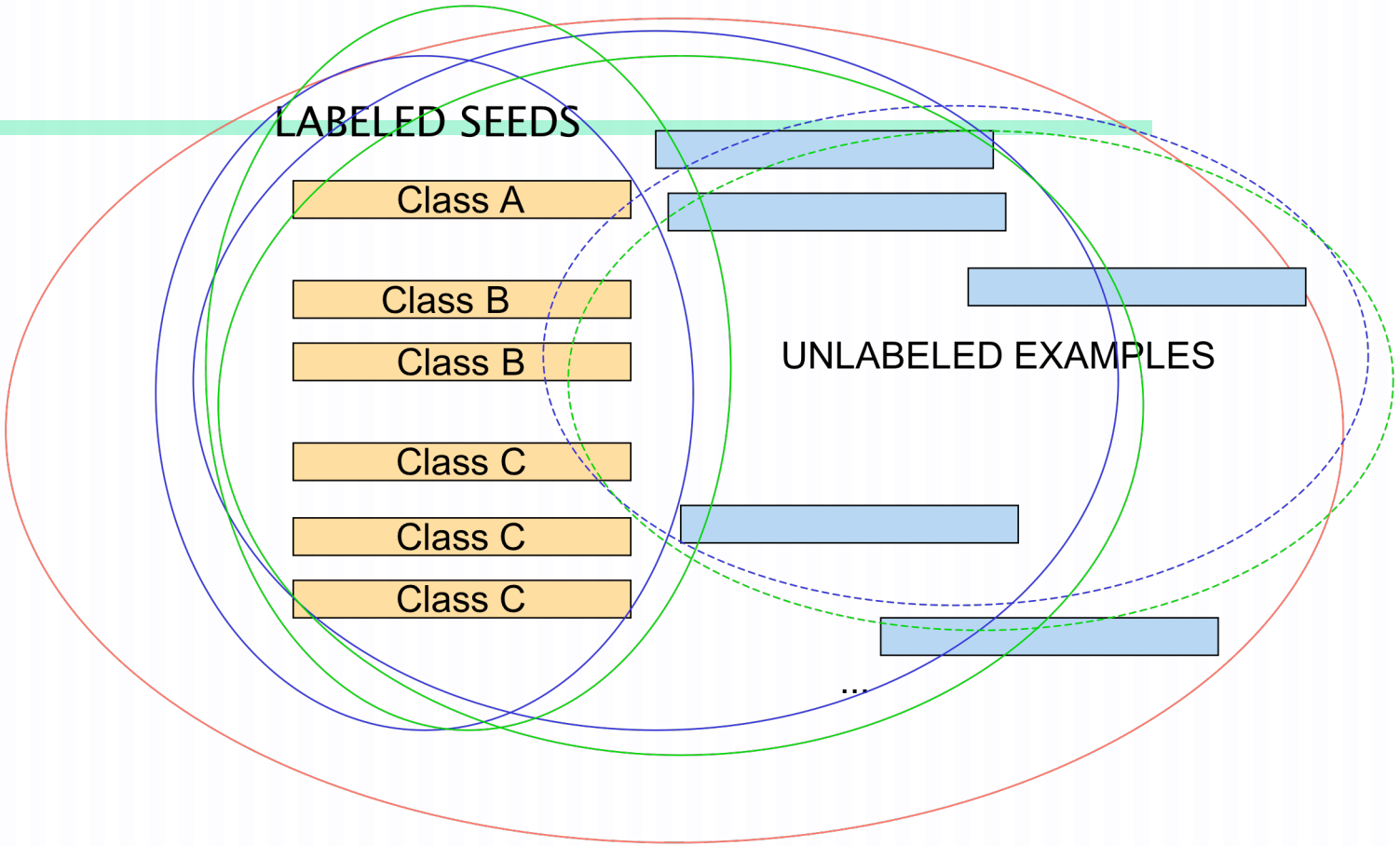


Fig. 6.4. Co-training: Two classifiers are incrementally trained (blue and green lines), first based on the labeled seeds, and then based on the labeled seeds and a set of unlabeled examples that are labeled with the current classifier. The dotted blue and green lines represent the set of all unlabeled examples that were considered for labeling in this step.

Weakly supervised learning

- **Active learning** = all examples to train from are labeled by a human, but the limited set of examples is carefully selected by the machine
- (Starts with labeled set on which the classifier is trained)
- Repeat
  - 1 example or set of examples is selected to label:
    - which are classified by the current classifier as most uncertain (informative examples)
    - that are representative or diverse (e.g., found by clustering)
- Until the trained classifier reaches a certain level of accuracy on a test set

Active learning

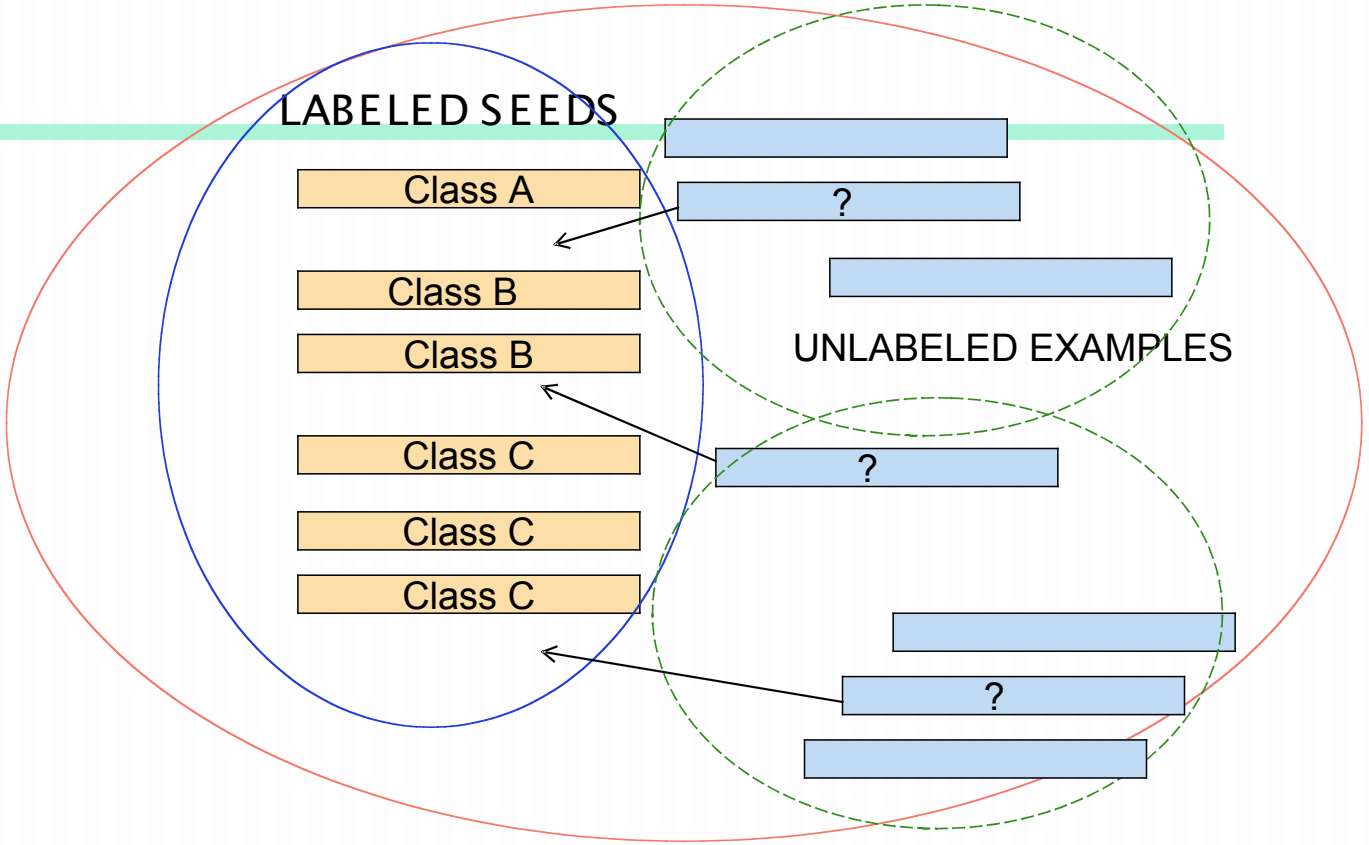


Fig. 6.5. Active learning: Representative and diverse examples to be labeled by humans are selected based on clustering.

Weakly supervised learning

- Problems with expansion and self-training:
  - How to choose the seeds and how many?
  - How to define additional constraints?
- Additional problem with co-training:
  - Can we find a natural split of independent features
- Issue with all techniques: How to select **good features** used in the classifier

Clustering with prior knowledge

- Clustering based on prior knowledge can also be considered as a form of weakly supervised learning:
  - Constraints built into the clustering process
  - Incorporate additional knowledge to modify distance metrics

Feature selection and extraction

- In classification tasks: object is described with set of **attributes** or **features**
- Typical features in text classification tasks:
  - word, phrase, syntactic class of a word, text position, the length of a sentence, the relationship between two sentences, an *n*-gram, a document (term classification), ….
  - choice of the features is application- and domain-specific
- Features can have a value, for text the value is often:
  - numeric, e.g., discrete or real values
  - nominal, e.g. certain strings
  - ordinal, e.g., the values 0= small number, 1 = medium number, 2 = large number

Feature selection and extraction

- The features together span a multi-variate space called the measurement space or feature space:
  - an object $\boldsymbol{x}$ can be represented as:
    - a **vector of features**:

      $\boldsymbol{x} = [x_1, x_2, \ldots, x_p]^T$

      where p = the number of features measured
    - as a **structure**: e.g.,
      - representation in first order predicate logic
      - graph representation (e.g., tree) where relations between features are figured as edges between nodes and nodes can contain attributes of features

Examples of classification features

SWARM INTELLIGENCE

Following a trail of insects as they work together to accomplish a task offers unique possibilities for problem solving.

By Peter Tarasewich & Patrick R. McMullen

Even with today's ever-increasing computing power, there are still many types of problems that are very difficult to solve. Particularly combinatorial optimization problems continue to pose challenges. An example of this type of problem can be found in product design. Take as an example the design of an automobile based on the attributes of engine horsepower, passenger seating, body style and wheel size. If we have three different levels for each of these attributes, there are 3^4, or 81, possible configurations to consider. For a slightly larger problem with 5 attributes of 4 levels, there are suddenly 1,024 combinations. Typically, an enormous amount of possible combinations exist, even for relatively small problems. Finding the optimal solution to these problems is usually impractical. Fortunately, search heuristics have been developed to find good solutions to these problems in a reasonable amount of time.

Over the past decade or so, several heuristic techniques have been developed that build upon observations of processes in the physical and biological sciences. Examples of these techniques include Genetic Algorithms (GA) and simulated annealing…

Sentence position

Words

POS-tag

The following and preceding word

Sentence length

Feature vectors for an example text

A Java Applet that scans Java Applets

- Binary values, based on lower-cased words:
  [a: 1, apple: 0, applet: 1, applets: 1, …., java: 1, …]
- Remove stopwords :
  [apple : 0, applet : 1, applets : 1, … , java : 1 …]
- Numeric value: based on text term frequency (*tf)*:
  [apple : 0, applet : 1, applets : 1, … , java : 2 …]
- Numeric value: based on text term frequency of lower cased *n-grams* (*tf)*:
  [aa: 0, a_a: 2, a_b: 0, …]
- Numeric attribute value based on latent semantic indexing:
  [F1: 0.38228938, F2: 0.000388, F3: 0.201033, …]
- ...

Feature selection

- = eliminating low quality features:
  - redundant features
  - noisy features
- decreases computational complexity
- decreases the danger of overfitting in supervised learning (especially when large number of features and few training examples)
- increases the chances of detecting valuable patterns in unsupervised learning and weakly supervised learning
- **Overfitting:**
  - the classifier perfectly fits the training data, but fails to generalize sufficiently from the training data to correctly classify the new case

Feature selection

- In supervised learning:
  - feature selection often incorporated in training algorithms:
    - incrementally add features, discard features, or both, evaluating the subset of features that would be produced by each change (e.g., algorithms that induce decision trees or rules from the sample data)
  - feature selection can be done after classification of new objects:
    - by measuring the error rate of the classification
    - those features are removed from or added to the feature set when this results in a lower error rate on the test set

Feature extraction

- = creates new features by applying a set of operators upon the current features:
  - a single feature can be replaced by a new feature (e.g., replacing words by their stem)
  - a set of features is replaced by one feature or another set of features
    - use of logical operators (e.g., disjunction), arithmetical operators (e.g. mean, LSI)
  - choice of operators: application- and domain-specific
- In supervised learning can be part of training or done after classification of new objects

Overview of the techniques and their main applications

- **Supervised learning** methods, e.g.,
  - statistical methods
    - support vector machines
    - naive Bayes classifiers
    - maximum entropy modeling
    - hidden Markov models,
    - conditional random fields
  - rule and tree learning

  = > used in text categorization, information extraction, text summarization, question answering, ...

  [see Text categorization, Information extraction, Text summarization]

Overview of the techniques and their main applications

- **Unsupervised learning** methods; e.g.,
  - clustering methods

  ⇒ *used in term clustering and expansion, document clustering, text summarization, ...*

  [see Text clustering, Text summarization]

Overview of the techniques and their main applications

- **Weakly supervised** learning methods: e.g.,
  - self-training and co-training
  - active learning

  => used in information extraction, word sense disambiguation, ...

  [see Information extraction, Cross-language information retrieval]

What have we learned?

- Typical text features to be used in retrieval and classification
- Basics graph computations
- (Weakly) supervised as well as unsupervised learning methods are important in text retrieval:
  - many tasks regard classification

Further reading

Allen, J. (1995). *Natural Language Understanding* (2nd ed.). Redwood City, CA: Benjamin/Cummings. (p. 23 ff.)

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer.

Klein, D., Kamvar, S.D. & Manning, C.D. (2002). From instance-level constraints to space-level constraints. Making the most of prior knowledge in data clustering. In *Proceedings of the 19th International Conference on Machine Learning* (pp. 307-314). San Francisco, CA: Morgan Kaufman.

Mihalcea, R. & Radev, D. (2006). Graph-based algorithms for information retrieval. Tutorial at the *Human Language Technology / North American Chapter of the Association for Computational Linguistics Conference 2006*.

Moens, M.-F. (2006). *Information Extraction: Algorithms and Prospects in a Retrieval Context* (*The Information Retrieval Series* 21). New York: Springer (Chapters 5 and 6).