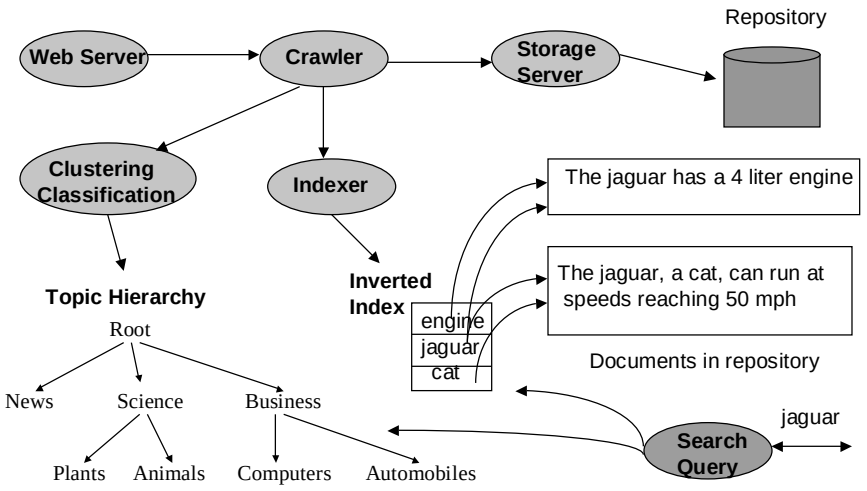
8. Mining Text and Web Data

Contents of this Chapter

8.1 The Web and Web Search



8.1 Web Search Engines

Keyword Search

"data mining"
519,000 results

8.1 Web Search Engines

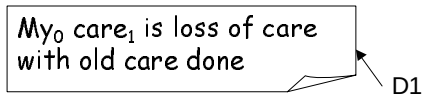
Boolean search
- care AND NOT old

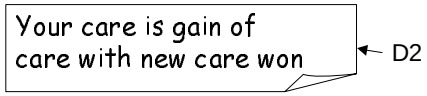
Phrases and proximity
- "new care"
- loss NEAR/5 care
- <SENTENCE>

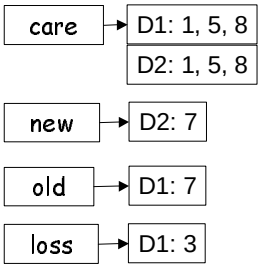
Indexing

Inverted Lists

How to rank the result lists?







8.1 Ranking Web Pages

Page Rank [Brin & Page 98]

- Idea: the more high ranked pages link to a web page, the higher its rank.

$$PageRank(v) = p + (1-p)\sum_{u \to v} \frac{PageRank(u)}{OutDegree(u)}$$

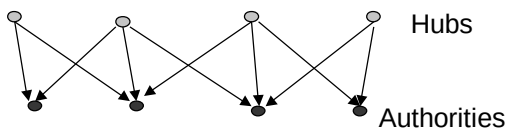
- Interpretation by random walk:
  - PageRank is the probability that a "random surfer" visits a page
  - » Parameter *p* is probability that the surfer gets bored and starts on a new random page.
  - » (1-*p*) is the probability that the random surfer follows a link on current page.
- PageRanks correspond to principal eigenvector of the normalized link matrix.
- Can be calculated using an efficient iterative algorithm.

8.1 Ranking Web Pages

Hyperlink-Induced Topic Search (*HITS*) [Kleinberg 98]

Definitions

- *Authorities*: highly-referenced pages on a topic.
- *Hubs*: pages that "point" to authorities.
- A good authority is pointed to by many good hubs; a good hub points to many good authorities.



8.1 Ranking Web Pages

HITS

Method

- Collect seed set of pages S (e.g., returned by search engine).
- Expand seed set to contain pages that point to or are pointed to by pages in seed set.
- Initialize all hub/authority weights to 1.
- Iteratively update hub weight h(p) and authority weight a(p) for each page:

$$a(p) = \sum_{q \to p} h(q) \qquad h(p) = \sum_{p \to q} a(q)$$

- Stop, when hub/authority weights converge.

8.1 Ranking Web Pages

Comparison

- PageRanks computed initially for web pages independent of search query.
- PageRank is used by web search engines such as Google.

- HITS: Hub and authority weights computed for different root sets in the context of a particular search query.
- HITS is applicable, e.g., for ranking the crawl front of a focused web crawler.
- Can use the relevance of a web page for the given topic to weight the edges of the web (sub) graph.

8.1 Directory Services

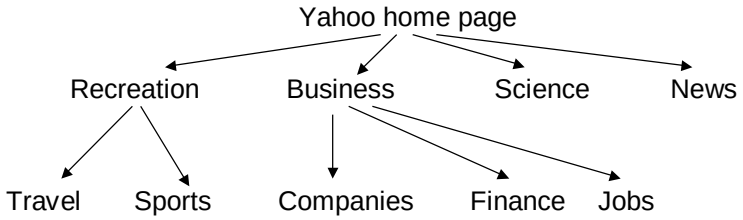
Browsing

"data mining"
~ 200 results

8.1 Directory Services

Topic Hierarchy

- Provides a hierarchical classification of documents / web pages (e.g., Yahoo!)



- Topic hierarchy can be browsed interactively to find relevant web pages.
- Keyword searches performed in the context of a topic: return only a subset of web pages related to the topic.

8.1 Mining Text and Web Data

Shortcomings of the Current Web Search Methods

Low precision

- Thousands of irrelevant documents returned by web search engine
  99% of information of no interest to 99% of people.

Low recall

- In particular, for directory services (due to manual acquisition).
- Even largest crawlers cover less than 50% of all web pages.

Low quality

- Many results are out-dated, broken links etc.

8.1 Mining Text and Web Data

Data Mining Tasks

- Classification of text documents / web pages

  To insert into topic hierarchy, as feedback for focused web crawlers, . . .

- Clustering of text documents / web pages

  To create topic hierarchy, as front-end for web search engines . . .

- Resource discovery

  Discovery of relevant web pages using a focused web crawler,
  in particular ranking of the links at the crawl front

8.1 Mining Text and Web Data

Challenges

- Ambiguity of natural language
  synonyms, homonyms, . . .
- Different natural languages
  English, Chinese, . . .
- Very high dimensionality of the feature spaces
  thousands of relevant terms
- Multiple datatypes
  text, images, videos, . . .
- Web is extremely dynamic
  > 1 million pages added each day
- Web is a very large distributed database
  huge number of objects, very high access costs

8.2 Text Representation

Preprocessing

- Remove HTML tags, punctuation etc.
- Define terms (single-word / multi-word terms)
- Remove stopwords
- Perform stemming
- Count term frequencies
- Some words are more important than others
  smooth the frequencies,
  e.g. weight by inverse document frequency

8.2 Text Representation

Transformation

- Different definitions of inverse document frequency

  n(d,t): number of occurrences of term t in document d

$$\frac{n(d,t)}{\sum_d n(d,t)}, \quad \frac{n(d,t)}{\sum_t n(d,t)}, \quad \frac{n(d,t)}{\max_t n(d,t)}$$

- Select "significant" subset of all occurring terms
- Vocabulary *V*, term t_i, document *d* represented as

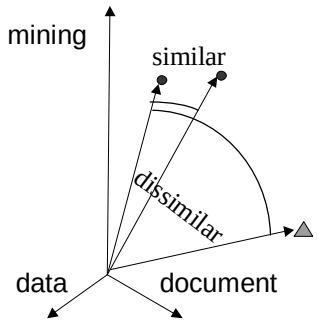
$$rep(d) = \{n(d,t_i)\}_{t_i \in V} \qquad \textit{Vector Space Model}$$

→ Most *n*'s are zeroes for a single document

8.2 Text Representation

Similarity Function

$$similarity(d_1, d_2) = \frac{\langle rep(d_1), rep(d_2) \rangle}{\|rep(d_1)\| \cdot \|rep(d_2)\|} \qquad \langle \cdot, \cdot \rangle = \text{inner product}$$



Cosine Similarity

8.2 Text Representation

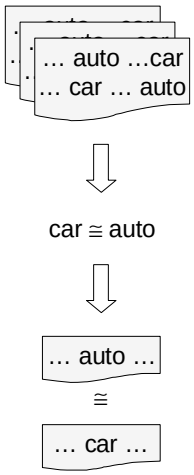
Semantic Similarity

Where can I fix my scooter?

A great garage to repair your 2-wheeler is …

→ A scooter is a 2-wheeler.

→ Fix and repair are synonyms.



Two basic approaches for semantic similarity

- Hand-made thesaurus (WordNet)
- Co-occurrence and associations

8.3 Text Clustering

Overview

Standard (Text) Clustering Methods

- ❑ Bisecting k-means
- ❑ Agglomerative Hierarchical Clustering

Specialised Text Clustering Methods

- ❑ Suffix Tree Clustering
- ❑ Frequent-Termset-Based Clustering

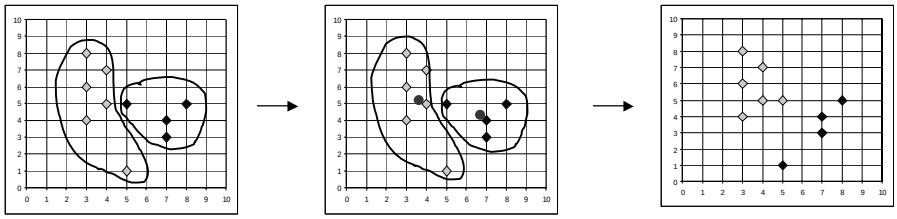
So far, no special methods for

- Exploiting hyperlinks
- Using other datatypes than text.

8.3 Text Clustering

Bisecting k-means [Steinbach, Karypis & Kumar 2000]

K-means



Bisecting k-means

- Partition the database into 2 clusters
- **Repeat**: partition the largest cluster into 2 clusters . . .
- **Until** *k* clusters have been discovered

8.3 Text Clustering

Bisecting k-means

Two types of clusterings

- Hierarchical clustering
- Flat clustering: any cut of this hierarchy

Distance Function

- Cosine measure (similarity measure)

$$s(\alpha,\beta) = \frac{\langle g(c(\alpha)), g(c(\beta))\rangle}{\|g(c(\alpha))\| \cdot \|g(c(\beta))\|}$$

In high-dimensional space: similarity values tend to be similar

8.3 Text Clustering

Agglomerative Hierarchical Clustering

1. Form initial clusters consisting of a singleton object, and compute the distance between each pair of clusters.
2. Merge the two clusters having minimum distance.
3. Calculate the distance between the new cluster and all other clusters.
4. If there is only one cluster containing all objects:
   Stop, otherwise go to step 2.

Representation of a Cluster C

$$rep(C) = \left\{ \sum_{d \in C} n(d, t_i) \right\}_{t_i \in V}$$

8.3 Text Clustering

Experimental Comparison [Steinbach, Karypis & Kumar 2000]

Clustering Quality
- Measured as entropy on a prelabeled test data set
- Using several text and web data sets
- Bisecting k-means outperforms k-means.
- Bisecting k-means outperforms agglomerative hierarchical clustering.

Efficiency
- Bisecting k-means is much more efficient than agglomerative hierarchical clustering.
- O(n) vs. $O(n^2)$

8.3 Text Clustering

Suffix Tree Clustering [Zamir & Etzioni 1998]

Forming Clusters

- Not by similar feature vectors
- But by common terms

Strengths of Suffix Tree Clustering (STC)

- Efficiency: runtime O(*n*) for *n* text documents
- Overlapping clusters

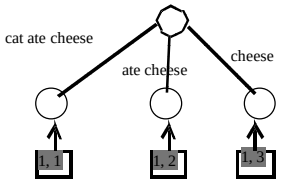
Method

1. Identification of „basic clusters“
2. Combination of basic clusters

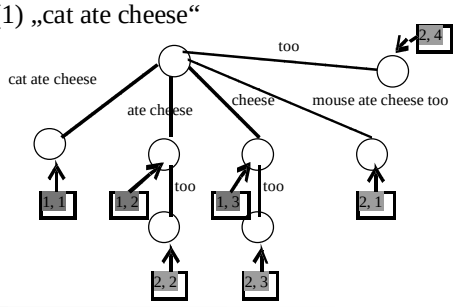
8.3 Text Clustering

Identification of Basic Clusters

- *Basic Cluster*: set of documents sharing one specific phrase
- *Phrase*: multi-word term
- Efficient identification of basic clusters using a *suffix-tree*



Insertion of (1) „cat ate cheese“



Insertion of (2) „mouse ate cheese too“

8.3 Text Clustering

Combination of Basic Clusters

- Basic clusters are highly overlapping
- Merge basic clusters having too much overlap
- *Basic clusters graph*: nodes represent basic clusters

  Edge between A and B iff $|A \cap B| / |A| > 0{,}5$ and $|A \cap B| / |B| > 0{,}5$
- *Composite cluster*:

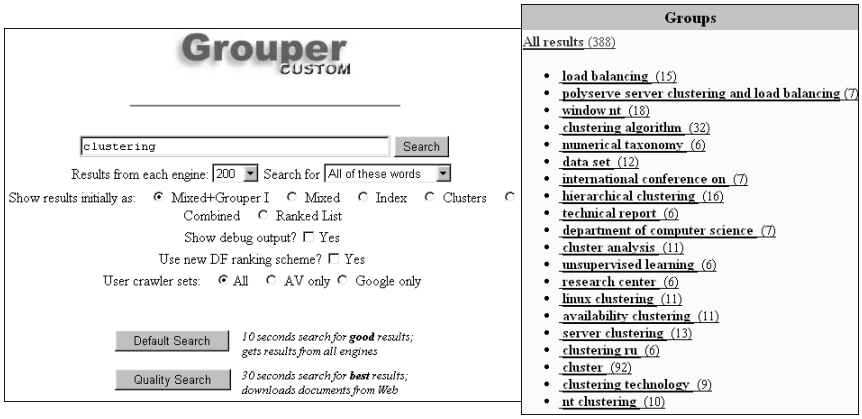
  a component of the basic clusters graph
- Drawback of this approach:

  Distant members of the same component need not be similar

  No evaluation on standard test data

8.3 Text Clustering

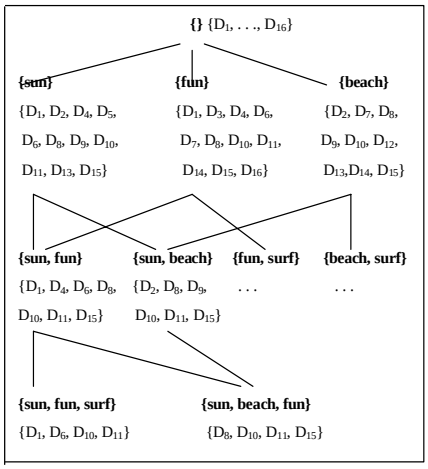
Example from the Grouper System



8.3 Text Clustering

Frequent-Term-Based Clustering [Beil, Ester & Xu 2002]

- Frequent term set: *description of a cluster*
- Set of documents containing all terms of the frequent term set: *cluster*
- *Clustering*: subset of set of all frequent term sets covering the DB with a low mutual overlap



8.3 Text Clustering

Method

- Task: efficient calculation of the overlap of a given cluster (description) F_i with the union of the other cluster (description)s
- F: *set of all frequent term sets* in D
- f_j: the number of all frequent term sets supported by document D_j

$$f_j = |\{F_i \in F \mid F_i \subseteq D_j\}|$$

- *Standard overlap* of a cluster C_i:

$$SO(C_i) = \frac{\sum_{Dj \in Ci} (f_j - 1)}{|C_i|}$$

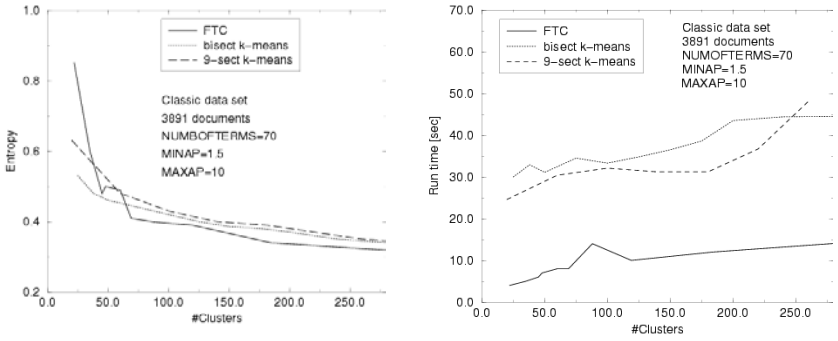
8.3 Text Clustering

Algorithm FTC

```
FTC(database D, float minsup)
SelectedTermSets:= {};
n:= |D|;
RemainingTermSets:= DetermineFrequentTermsets(D, minsup);
while |cov(SelectedTermSets)| ≠ n do
  for each set in RemainingTermSets do
    Calculate overlap for set;
  BestCandidate:=element of Remaining TermSets with minimum overlap;
  SelectedTermSets:= SelectedTermSets ∪ {BestCandidate};
  RemainingTermSets:= RemainingTermSets - {BestCandidate};
  Remove all documents in cov(BestCandidate) from D and from the
    coverage of all of the RemainingTermSets;
return SelectedTermSets;
```


8.3 Text Clustering

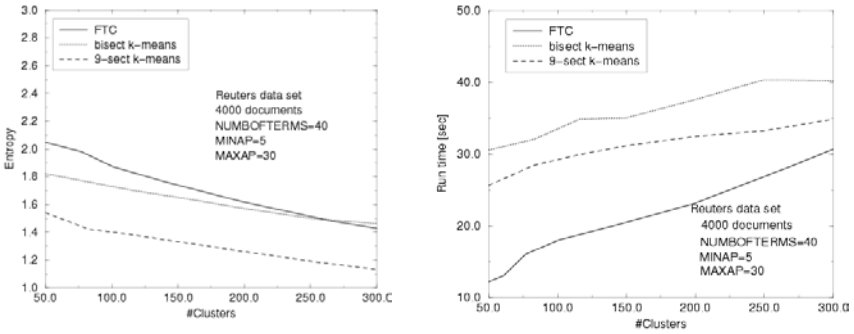
Experimental Evaluation



Classic Dataset

8.3 Text Clustering

Experimental Evaluation



Reuters Dataset

FTC achieves comparable clustering quality more efficiently
Better cluster description

8.4 Text Classification

Overview

Standard Text Classification Methods

- Naive Bayes Classifier
- Bag of Words Model

Advanced Methods

- Use EM to exploit non-labeled documents
- Exploit hyperlinks

8.4 Text Classification

Naive Bayes Classifier

- Decision rule of the *Bayes Classifier*: $\underset{c_j \in C}{argmax}\, P(d \mid c_j) \cdot P(c_j)$
- Assumptions of the *Naive Bayes Classifier*:
  - $d = (d_1, \ldots, d_d)$
  - the attribute values d_i are *independent* from each other
- Decision rule of the *Naive Bayes Classifier* :

$$\underset{c_j \in C}{argmax}\, P(c_j) \cdot \prod_{i=1}^{d} P(d_i \mid c_j)$$

8.4 Text Classification

Bag of Words Model

- Allows to estimate the $P(d_j \mid c)$ from the training documents
- Model for generating a document d of class c consisting of n terms
- Neglects order of terms in document (bag!)
- Bernoulli experiment:

  Throw n times a dice which has one side for each term t_i

  Probability of t_i: relative frequency of term t_i in training documents of class c

8.4 Text Classification

Bag of Words Model

Problem

- Term t_i appears in no training document of class c_j
- t_i appears in a document d to be classified
- Document d also contains terms which strongly indicate class c_j

$$P(d_j|\, c) = 0 \text{ and } P(d|\, c) = 0$$

Solution

- Smoothing of the relative frequencies in the training documents

8.4 Text Classification

Experimental Evaluation [Craven et al. 1998]

Training data

- 4127 web pages of CS departments
- Classes: department, faculty, staff, student, course, . . ., other

Major Results

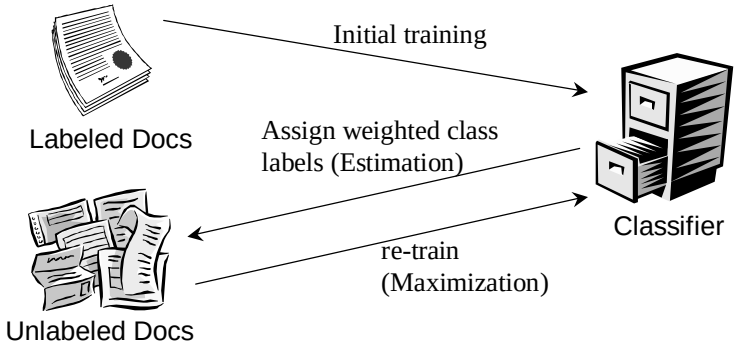
- Classification accuracy of 70% to 80 % for most classes
- Only 9% accuracy for class *staff*, but 80% correct in superclass *person*
- Low classification accuracy for class *other*

Web pages are harder to classify than „professional“ text documents

8.4 Text Classification

Exploiting Unlabeled Documents [Nigam et al. 1999]

- Labeling is laborious, but many unlabeled documents available.
- Exploit unlabeled documents to improve classification accuracy:



8.4 Text Classification

Exploiting Unlabeled Documents

- Let training documents *d* belong to classes in a *graded* manner Pr(*c*|*d*).
- Initially labeled documents have 0/1 membership.
- *Expectation Maximization* (*EM*)

  repeat

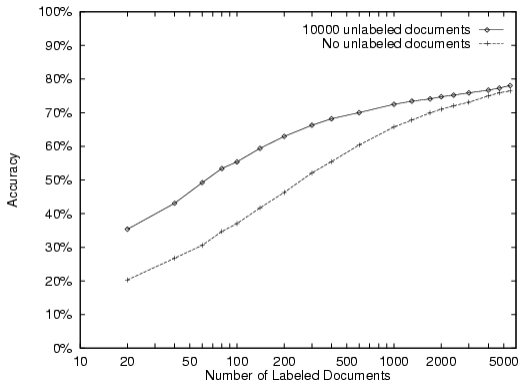
    calculate class model parameters P(d_i|*c*);

    determine membership probabilities P(*c*|*d*);

  until classification accuracy does no longer improve.

8.4 Text Classification

Exploiting Unlabeled Documents



Small labeled set → large accuracy boost

8.4 Text Classification

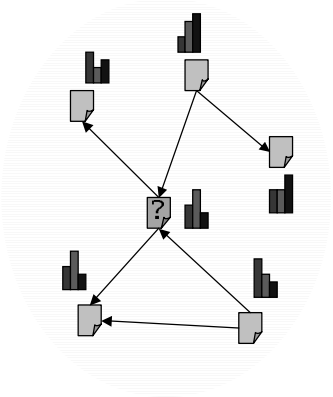
Exploiting Hyperlinks [Chakrabarti, Dom & Indyk 1998]

- Logical "documents" are often fragmented into several web pages.
- Neighboring web pages often belong to the same class.
- Web documents are extremely diverse.
- Standard text classifiers perform poorly on web pages.
- For classification of web pages, use
  - text of the page itself
  - text and class labels of neighboring pages.

8.4 Text Classification

Exploiting Hyperlinks

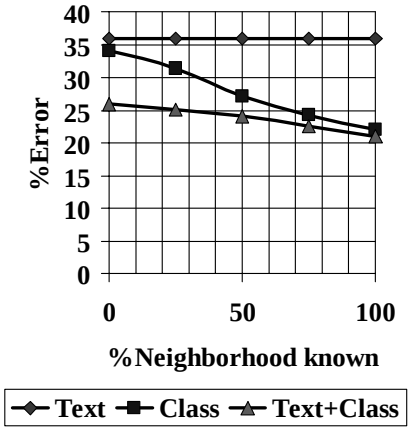
- c = class, t = text, N = neighbors
- Text-only model: P(t|c)
- Using neighbors' text to judge my topic: P(t, t(N) | c)
- Using neighbors' class label to judge my topic: P(t, c(N) | c)
- Most of the neighbor's class labels are unknown.
- Method: iterative relaxation labeling.



8.4 Text Classification

Experimental Evaluation

- Using neighbors' text *increases* classification error.
- Even when tagging non-local texts.
- Using neighbors' class labels *reduces* classification error.
- Using text *and* class of neighbors yields the *lowest classification error*.



References

Beil F., Ester M., Xu X.: "Frequent Term-Based Text Clustering", Proc. KDD '02, 2002.

Brin S., Page L.: "The anatomy of a large-scale hypertextual Web search engine", Proc. WWW7, 1998.

Chakrabarti S., Dom B., Indyk P.: "Enhanced hypertext categorization using hyperlinks", Proc. ACM SIGMOD 98, 1998.

Craven M., DiPasquo D., Freitag D., McCallum A., Mitchell T., Nigam K., Slattery S.: "Learning to Extract Symbolic Knowledge from the World Wide Web", Proc. AAAI 98, 1998.

Deerwater S., Dumais S. T., Furnas G. W., Landauer T. K., Harshman R.: "Indexing by latent semantic analysis", Journal of the Society for Information Science, 41(6), 1990.

Kleinberg J.: "Authoritative sources in a hyperlinked environment", Proc. SODA, 1998.

Nigam K., McCallum A., Thrun S., Mitchell T.: "Text Classification from Labeled and Unlabeled Documents using EM", Machine Learning, 39(2/3). pp. 103-134. 2000.

Steinbach M., Karypis G., Kumar V.: "A comparison of document clustering techniques", Proc. KDD Workshop on Text Mining, 2000.

Zamir O., Etzioni O.: "Web Document Clustering: A Feasibility Demonstration", Proc. SIGIR 1998.