Text Mining

Mantrach Amin(ULB), with
Nicolas Vanzeebroeck (ULB)
Hugues Bersini (ULB)
Marco Saerens (UCL)

Document retrieval: A short overview of some old and recent techniques

Contents

- General introduction
- Information retrieval: Basic standard techniques (content-based methods)
  - Documents pre-processing
  - Vector-space model
  - Probabilistic model
  - Assessment of performances
- Information retrieval: More recent techniques
  - Exploiting links between documents (web link analysis))
    - The PageRank algorithm
    - The HITS algorithm
  - Exploiting the relational structure in order to improve retrieval

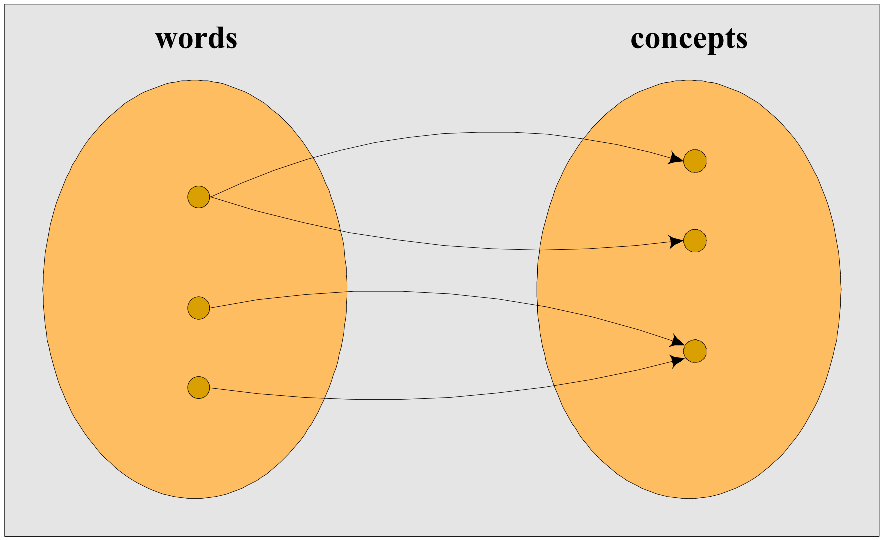
General introduction

Introduction

- We have a collection of documents (mainly text or html-based)
- We have a set of users
- A user wants to retrieve the documents related to a given concept
- He consequently submits a query expressed through words or terms
- An information retrieval system returns the documents most related to this concept

Introduction

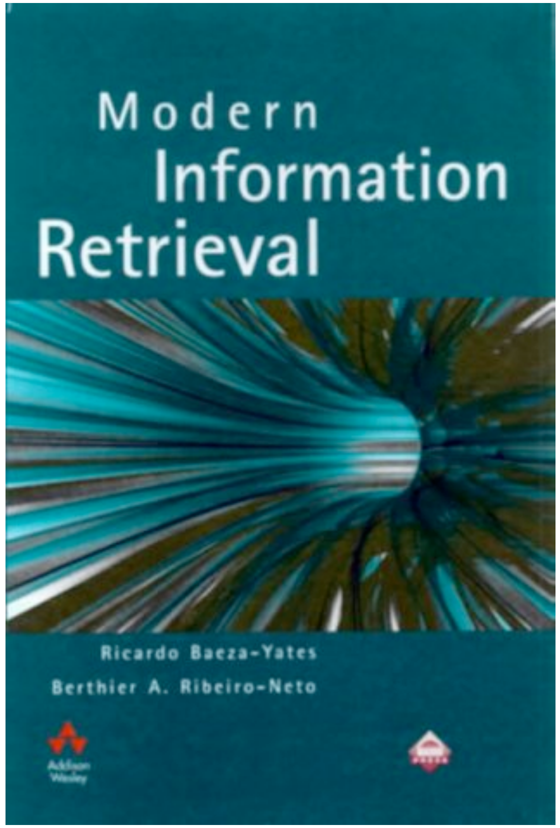
- One major problem:
  - We want to express a concept
  - With words
  - There is no one-to-one mapping (eg. marché)



Documents preprocessing

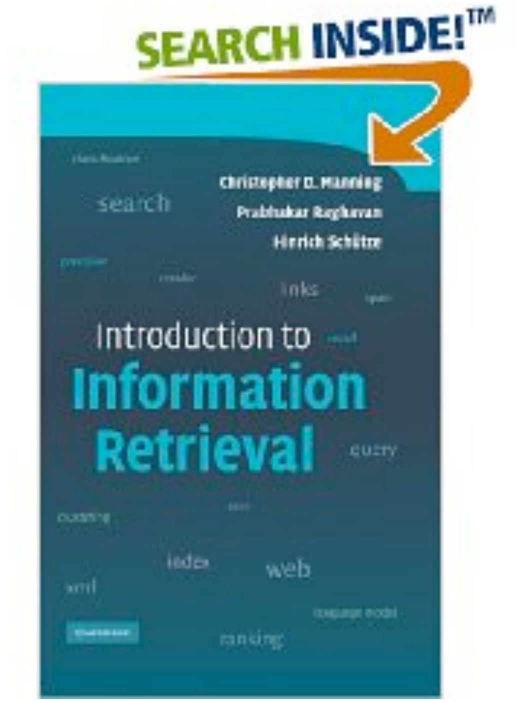
Documents preprocessing

- R. Baeza-Yates & B. Ribeiro-Neto (1999)
  - Modern Information Retrieval
  - Addison Wesley



Documents preprocessing

- Manning
  - Introduction to information retrieval
  - Cambridge University Press

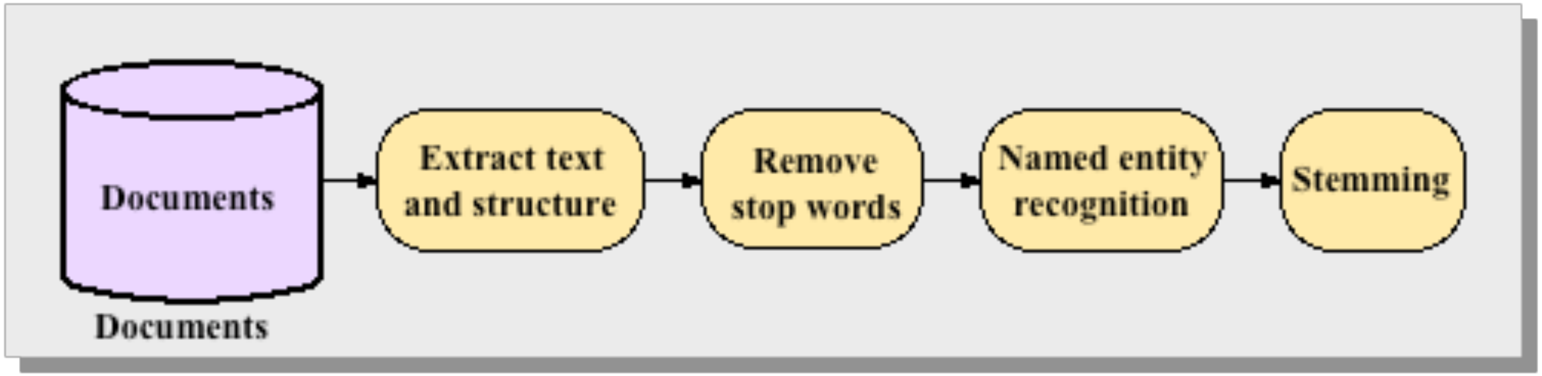


Documents pre-processing

- We have a collection of documents
- Here are the standard pre-processing steps
  - Extract text and structure (eg. from Microsoft Word or LaTeX to XML)
  - Remove stop words (eg. remove "the", "at", "all", etc)
  - Named entity recognition (eg. find proper names)
  - Stemming (eg. extract "process" from "processing)

Documents pre-processing

- It is a tedious job



  – But some tools are readily available
    - Galilei project developped at the ULB

Documents pre-processing

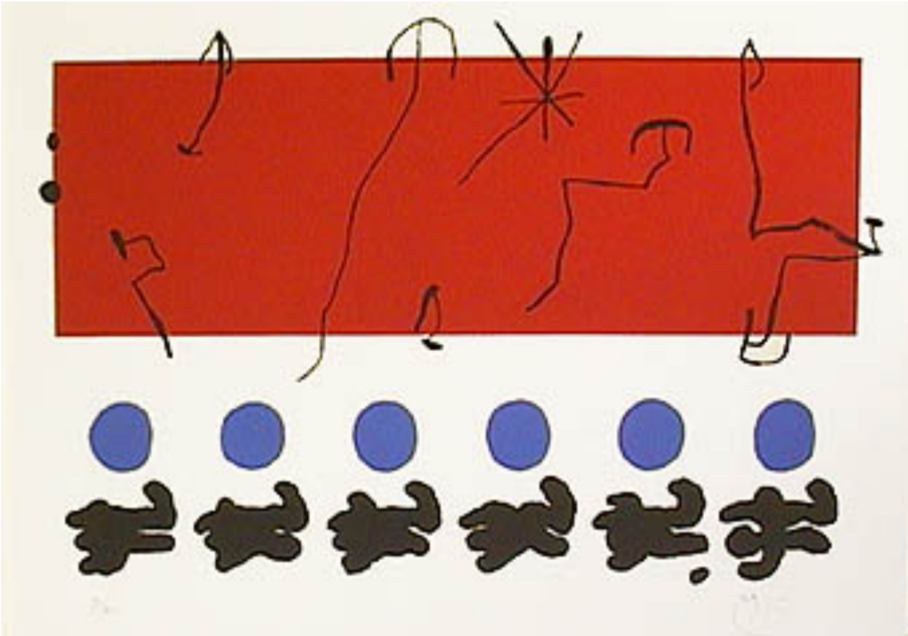
- Stemming aims to extract the « root » of the words
- Stemming can be based on
  - A dictionnary (for instance Mmorph developped at the University of Geneva)
  - A set of rules developped by linguists (like Porter's stemming algorithm for english)

Documents pre-processing

- Example of stemming rules in french:

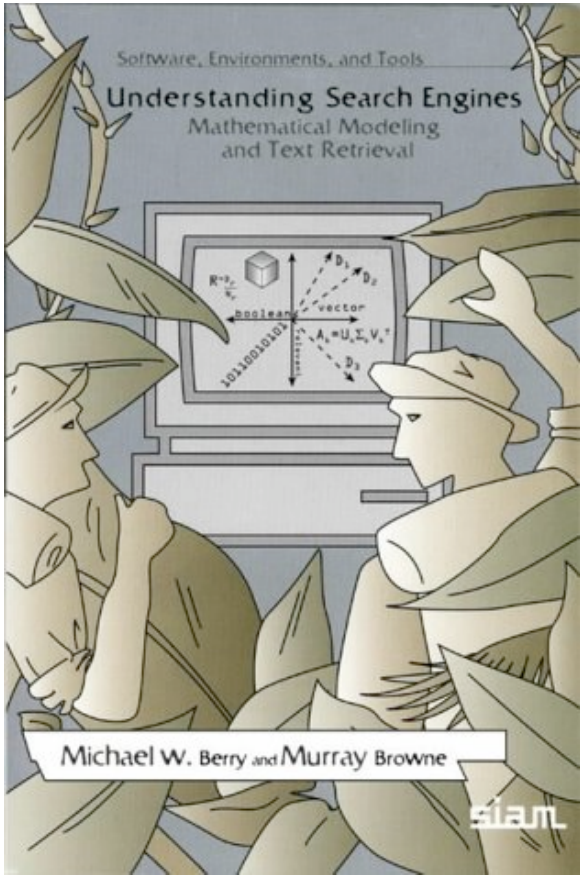
$$
\begin{aligned}
(m > 0)\ aux &\rightarrow al \\
(m > 0)\ ouse &\rightarrow ou \\
(m > 0)\ eille &\rightarrow eil \\
(m > 0)\ nne &\rightarrow n \\
(m > 0)\ fs &\rightarrow v
\end{aligned}
$$

Basic Methods
The vector space model



The vector space method

- M. W. Berry & M. Browne (1999)
  - Understanding Search Engines
  - SIAM



The vector space model: Introduction

- In its basic form, each document is represented by a vector
  - A query is also represented by a vector
  - A user profile may be represented by a vector as well
- The coordinates of the vector are words
  - Each element of the vector represents the frequency of the word in the document or the query
  - In the space of words

The vector space model: Basics

- Thus a document is represented by a vector
  - Document j is characterized by $\mathbf{d}_j$
  - f_{ij} is the frequency of word w_i in document j
  - The total number of words is n_w
- The dimension of the vector is n_w

The vector space model: Basics

- Thus each document is represented by

$$\mathbf{d}_j \triangleq \begin{bmatrix} f_{1j} \\ f_{2j} \\ \vdots \\ f_{n_w j} \end{bmatrix}$$

- This is called the « bag of words » representation in the words space
  - The order of the words is not taken into account
  - This vector is usually sparse
  - This vector is very large

The vector space model: Basics

- The total number of documents is n_d
- The terms-documents matrix is

$$\mathbf{D} \triangleq \begin{array}{c} \text{documents} \\ \left[\begin{array}{cccc} f_{11} & f_{12} & \cdots & f_{1n_d} \\ f_{21} & f_{22} & \cdots & f_{2n_d} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n_w1} & f_{n_w2} & \cdots & f_{n_wn_d} \end{array} \right] \end{array} \left.\vphantom{\begin{array}{c} f \\ f \\ \vdots \\ f \end{array}}\right\} \text{words}$$

The vector space model: Basics

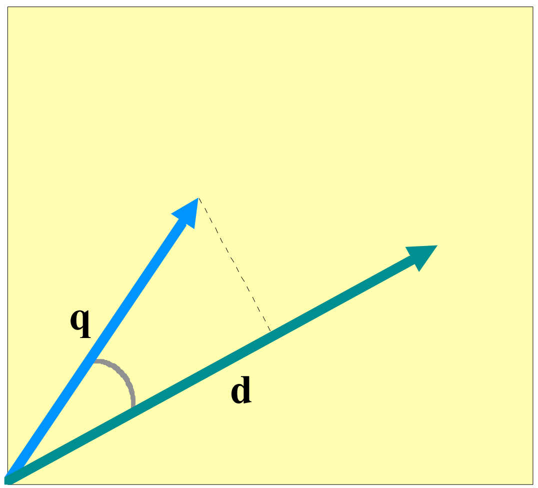
- A query is also represented by a vector
  - Here is a query $\mathbf{q}$
  - Each element is 0 or 1 (presence or absence of a word)

$$\mathbf{q} \triangleq \begin{array}{c c} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} & \begin{array}{c} \\ \\ \\ i \\ \\ \\ \\ \end{array} \end{array}$$

word w_i is present in the query

The vector space model: Basics

- The purpose is of course to retrieve documents $\mathbf{d}_i$ based on a query $\mathbf{q}$
- We have to define a notion of similarity between a query and a document



The vector space model: Basics

- The similarity between a query $\mathbf{q}$ and a document $\mathbf{d}_i$ can be defined as
  - The cosinus of the angle between these two vectors:

$$\mathrm{sim}(\mathbf{q}, \mathbf{d}_i) \triangleq \cos(\mathbf{q}, \mathbf{d}_i) = \frac{\mathbf{q}^{\mathrm{T}}\mathbf{d}_i}{\|\mathbf{q}\| \, \|\mathbf{d}_i\|}$$

  - Euclidean distance does not work well because queries contain much lesser words than documents
  - It is called the cosine similarity

The vector space model: Basics

- The similarity between the query and all documents can be computed by using the term-document matrix

$$
\begin{aligned}
\cos(\mathbf{q}, \mathbf{D}) &= \frac{\mathbf{q}^{\mathrm{T}}}{\|\mathbf{q}\|} \mathbf{D}\, \mathrm{diag}\left[\frac{1}{\|\mathbf{d}_i\|}\right] \\
&= \frac{\mathbf{q}^{\mathrm{T}}}{\|\mathbf{q}\|} \left[\begin{array}{ccccc} \mathbf{d}_1 & \ldots & \mathbf{d}_i & \ldots & \mathbf{d}_{n_d} \end{array}\right] \mathrm{diag}\left[\frac{1}{\|\mathbf{d}_i\|}\right] \\
&= \left[\begin{array}{ccccc} \frac{\mathbf{q}^{\mathrm{T}}\mathbf{d}_1}{\|\mathbf{q}\|} & \ldots & \frac{\mathbf{q}^{\mathrm{T}}\mathbf{d}_i}{\|\mathbf{q}\|} & \ldots & \frac{\mathbf{q}^{\mathrm{T}}\mathbf{d}_{n_d}}{\|\mathbf{q}\|} \end{array}\right] \mathrm{diag}\left[\frac{1}{\|\mathbf{d}_i\|}\right] \\
&= \left[\begin{array}{ccccc} \frac{\mathbf{q}^{\mathrm{T}}\mathbf{d}_1}{\|\mathbf{q}\|\,\|\mathbf{d}_1\|} & \ldots & \frac{\mathbf{q}^{\mathrm{T}}\mathbf{d}_i}{\|\mathbf{q}\|\,\|\mathbf{d}_i\|} & \ldots & \frac{\mathbf{q}^{\mathrm{T}}\mathbf{d}_{n_d}}{\|\mathbf{q}\|\,\|\mathbf{d}_{n_d}\|} \end{array}\right]
\end{aligned}
$$

The vector space model: Refinements

- Two refinements of the basic model:
  – Term weighting
  – Latent semantic models

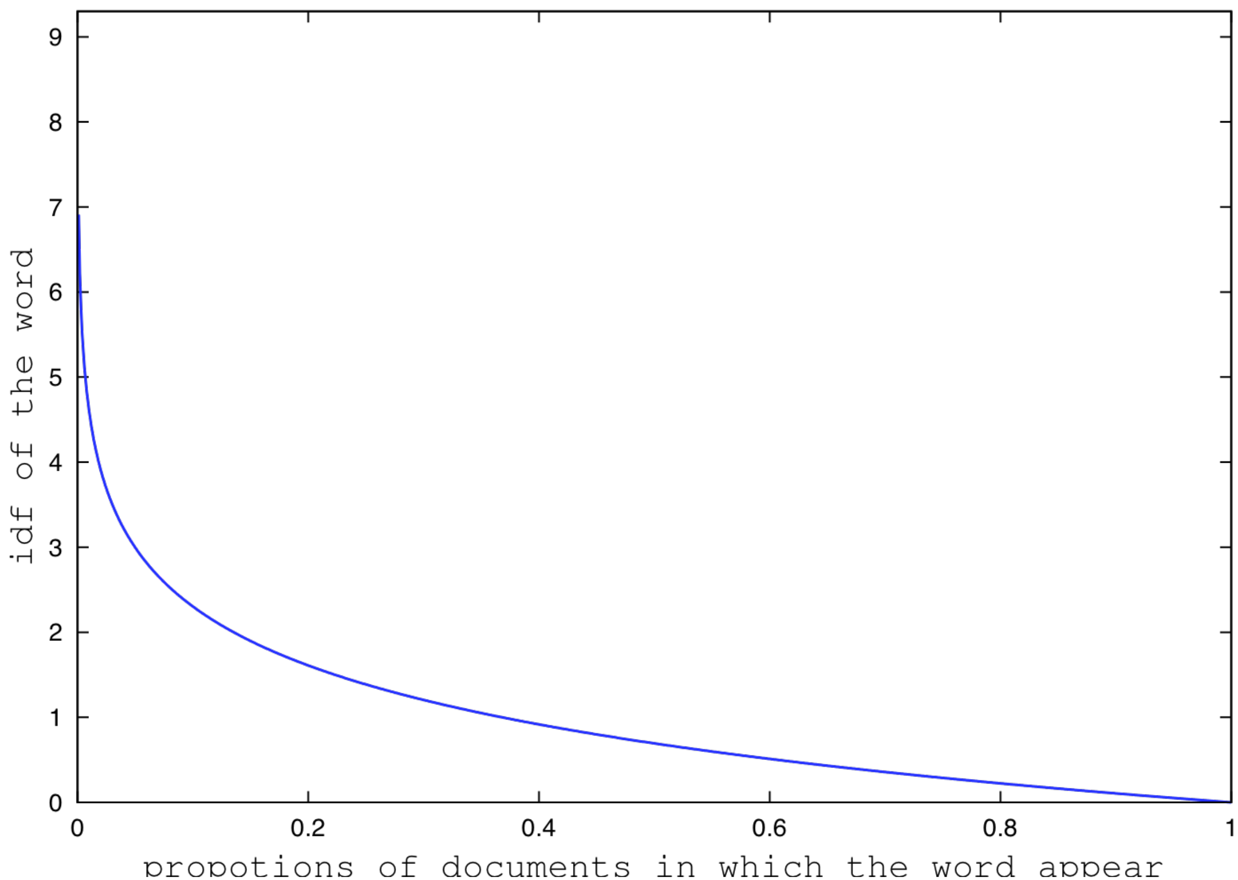
The vector space model: Term weighting

- We now introduce term weighting
  - Of course, each word does not have the same « weight »
  - We would like to take account of the "discriminative power" of every word
  - For instance, if a word is present in every document, it is useless
  - $P(w_i)$ is the a priori probability that word w_i appears in a document

The vector space model: Term weighting

- This quantity is often called the inverse document frequency (idf) associated to word w_i: $idf_i = -\log_2[P(w_i)]$
  - It is a measure of the general importance of the word (or term) w_i
- It is estimated by taking the logarithm of
  - the number of documents in which w_i appears, divided by the total number of documents

The vector space model: Term weighting



The vector space model: idf score

- Another quantity of interest is the term frequency, tf_{ij}

$$tf_{ij} = \frac{f_{ij}}{\sum_{i=1}^{n_w} f_{ij}}$$

  – It measures the importance of the term w_i within the particular document d_j
  – It is normalized to prevent a bias towards longer documents

The vector space model: tf.idf score

- The tf-idf score is simply the product of the tf and the idf scores, $tf.idf_{ij} = idf_i \, . \, tf_{ij}$
  - The tf-idf weighting scheme is often used in the vector space model together with cosine similarity
  - To determine, for instance, the similarity between two documents
- By replacing the term-frequency elements of the terms-documents matrix by the tf-idf scores

The vector space model: Term weighting

- We redefine the query vector $\mathbf{q}$ as

$$\mathbf{q} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -\log_2\left[P(w_i)\right] \\ 0 \\ \vdots \\ 0 \end{bmatrix} i$$

- Each word w_i is weighted by the information provided by knowing the presence of the word

The vector space model: Latent semantic models

- Latent semantic models
  - These models try to capture some semantic information
  - For instance, if we introduce a query with "newborn", it would be nice if documents containing "baby" but not "newborn" are also retrieved
  - We say that words are semantically related when they are used in the same context

The vector space model: Latent semantic models

- This way, we can capture some « semantic similarity » between words
  - In the present case, we will say that two words are semantically related
  - When they often occur in the same document

The vector space model: Latent semantic models

- One solution to this problem is to use "sub-space projection methods" like
  - "Singular Value Decomposition" (SVD) or
  - "factor analysis"
- The rank m SVD of a matrix of rank n is the « best approximation » to this matrix having rank $m < n$
  - In the present case, we use a SVD in order to reduce the rank of the term-document matrix

The vector space model: Latent semantic models

- This allows to reduce the dimensionality of the space by clustering the words that are semantically "similar",
- That is, used in the same documents
- This allows us to build a kind of concept space

The vector space model: Latent semantic models

- Every matrix has a "singular value decomposition":

$$\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$$

$$\text{where } \mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I} \text{ and } \mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}$$

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \end{bmatrix}$$

$$\text{with } \sigma_1 > \sigma_2 > \dots > \sigma_n > 0$$

The vector space model: Latent semantic models

- If we want the best rank-m approximation to $\mathbf{D}$, we put

$$\sigma_{m+1} = 0, \sigma_{m+2} = 0, \ldots, \sigma_n = 0$$

The vector space model: Latent semantic models

- So that we obtain

$$\widetilde{\boldsymbol{\Sigma}} = \begin{bmatrix} \sigma_1 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & \sigma_2 & 0 & & & & 0 \\ \vdots & 0 & \ddots & \ddots & & & \vdots \\ \vdots & & \ddots & \sigma_m & 0 & & \vdots \\ \vdots & & & 0 & 0 & \ddots & \vdots \\ \vdots & & & & \ddots & \ddots & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 & 0 \end{bmatrix}$$

The vector space model: Latent semantic models

- $\widetilde{\mathbf{D}}$ is the best rank-m approximation to $\mathbf{D}$

$$\widetilde{\mathbf{D}} = \mathbf{U}\widetilde{\boldsymbol{\Sigma}}\mathbf{V}$$

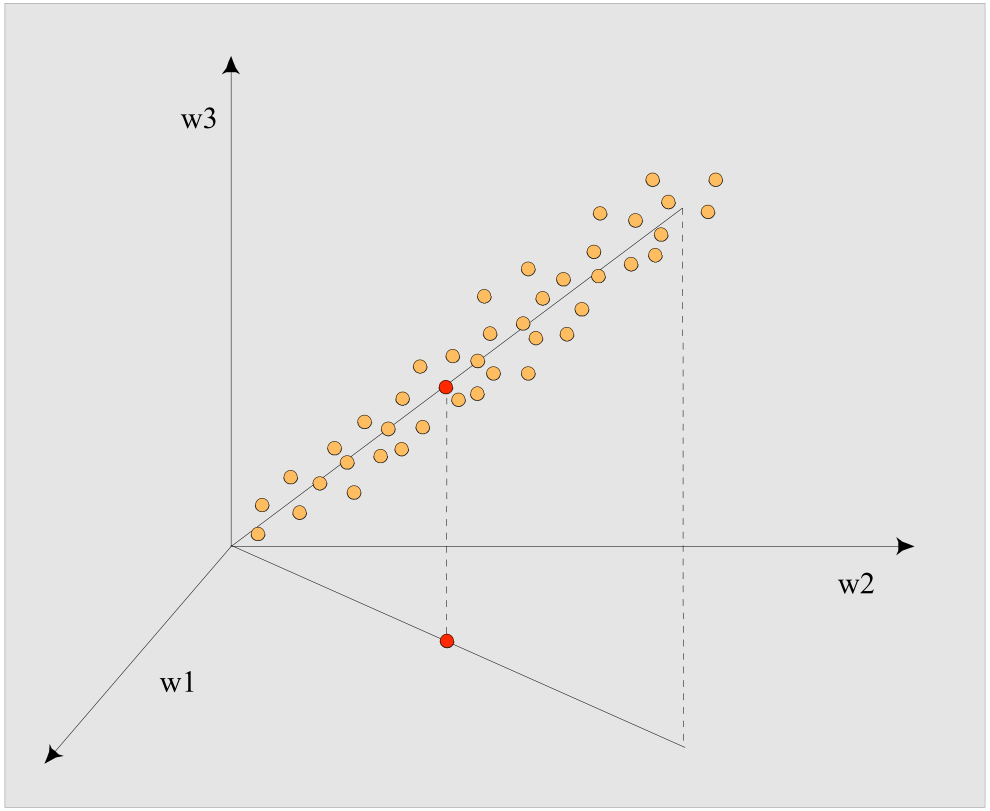
  – That is, there is no rank-m matrix closer to $\mathbf{D}$ in terms of the Frobenius norm

- The queries are now adressed to $\widetilde{\mathbf{D}}$ instead of $\mathbf{D}$

$$\mathrm{sim}(\mathbf{q}, \widetilde{\mathbf{d}}_i) = \cos(\mathbf{q}, \widetilde{\mathbf{d}}_i) = \frac{\mathbf{q}^{\mathrm{T}}\widetilde{\mathbf{d}}_i}{\|\mathbf{q}\| \left\|\widetilde{\mathbf{d}}_i\right\|}$$

The vector space model: Latent semantic models

- But how does it work ?



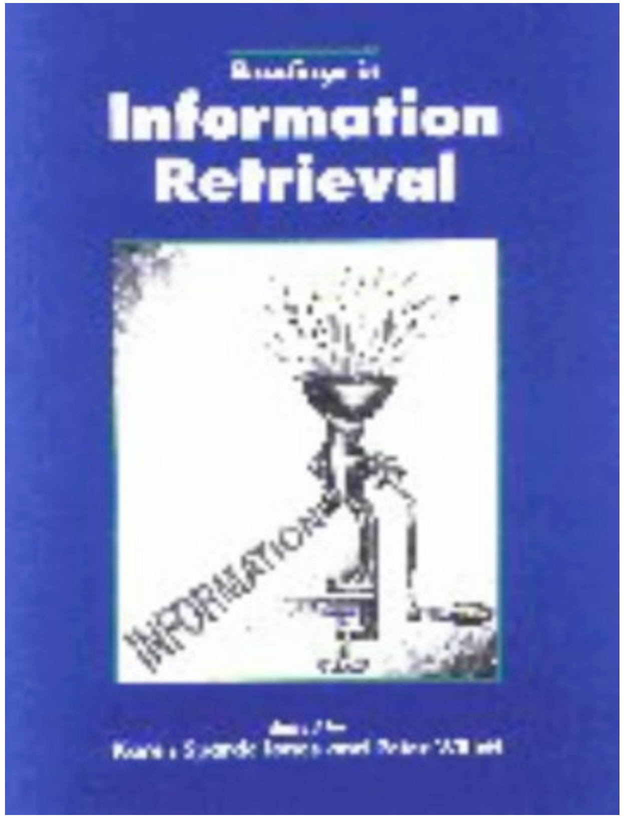
The vector space model: Conclusion

- The vector-space method relies on linear algebra concepts
- The SVD approach allows to work in a latent space representing concepts
- The main problem: How many dimensions of the subspace do we keep?

Basic Methods
Probabilistic methods

Probabilistic methods

- K. Sparck Jones & P. Willett (Editors) (1997)
  - Readings in Information Retrieval
  - Morgan Kaufmann
- Collection of papers



Probabilistic methods: Introduction

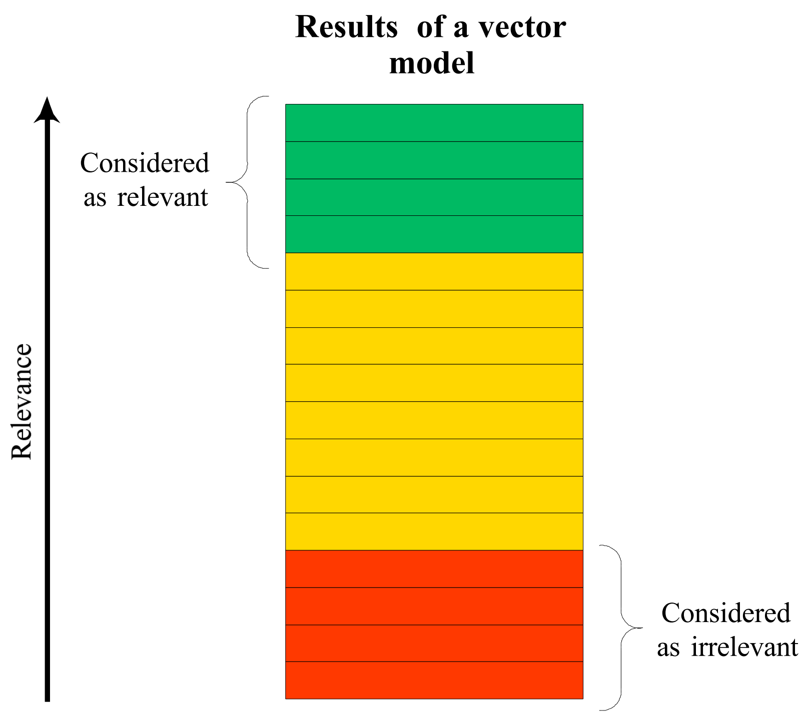
- The probabilistic methods rely on statistical models
  - Each user profile is represented by a statistical model
- A document can be relevant or not to a user
  - $R = 1$ if it is relevant; $R = 0$ if it is not relevant

Probabilistic methods: Introduction

- Based on
  - Relevance feedback from the user
  - Or simply the ranking of a vector space model
- We can build a probabilistic model
  - It will estimate the probability that a document is relevant
- We will introduce the binary independence retrieval model

Probabilistic methods: Introduction

- We introduced a query
  – Based on a vector space model, we obtain



Probabilistic methods: Introduction

- Expanding the query based on
  - – the most relevant documents or
  - – a relevance feedback from the used
- Is called query expansion

Probabilistic methods: Basic model

- Each document $\mathbf{d}_i$ is represented by a binary vector

$$\mathbf{d}_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

This word is present in the document

- $[\mathbf{d}_i]_j = 1$ if word w_j is in document $\mathbf{d}_i$
- $[\mathbf{d}_i]_j = 0$ if word w_j is not in document $\mathbf{d}_i$

Probabilistic methods: Basic model

- Based on ranking, some documents are considered as relevant $(R = 1)$
- And some documents are considered as not relevant $(R = 0)$
- To a user u_k

Probabilistic methods: Basic model

- We define $P(\mathbf{d} = \mathbf{x} | R = 1, u_k)$
  - as the probability of observing a document $\mathbf{d} = \mathbf{x}$ given that this document is relevant for user u_k
- We will see that it is easy to estimate these probabilities for the binary independence model

Probabilistic methods: Basic model

- However, during the document retrieval phase, we are mainly interested in:

$$P(R = 1|\mathbf{d} = \mathbf{x}, u_k)$$

- The larger this value, the more likely the document $\mathbf{x}$ is relevant
- This probability has to be computed for each document in the database

Probabilistic methods: Basic model

- Now, instead of computing

$$P(R=1|\mathbf{d}=\mathbf{x},u_k)$$

- It is easier to compute the odds

$$\lambda = \frac{P(R=1|\mathbf{d}=\mathbf{x},u_k)}{P(R=0|\mathbf{d}=\mathbf{x},u_k)}$$

$$= \frac{P(R=1|\mathbf{d}=\mathbf{x},u_k)}{1-P(R=1|\mathbf{d}=\mathbf{x},u_k)}$$

Probabilistic methods: Basic model

- It is a monotonic increasing function of

$$P(R = 1|\mathbf{d} = \mathbf{x}, u_k)$$

- It therefore provides the same ranking
- The larger this value λ, the more likely the document is relevant

Probabilistic methods: Basic model

- Remember Bayes' law !

$$P(R=1|\mathbf{d}=\mathbf{x},u_k) = \frac{P(\mathbf{d}=\mathbf{x}|R=1,u_k)P(R=1|u_k)}{P(\mathbf{d}=\mathbf{x}|u_k)}$$

$$P(R=0|\mathbf{d}=\mathbf{x},u_k) = \frac{P(\mathbf{d}=\mathbf{x}|R=0,u_k)P(R=0|u_k)}{P(\mathbf{d}=\mathbf{x}|u_k)}$$

Probabilistic methods: Basic model

- We can easily compute λ by assuming conditional independence between the words (d_n is element n of vector $\mathbf{d}$)

$$
\begin{aligned}
\lambda &= \frac{P(R=1|\mathbf{d}=\mathbf{x},u_k)}{P(R=0|\mathbf{d}=\mathbf{x},u_k)} \\
&= \frac{P(\mathbf{d}=\mathbf{x}|R=1,u_k)}{P(\mathbf{d}=\mathbf{x}|R=0,u_k)} \times \frac{P(R=1|u_k)}{P(R=0|u_k)} \\
&= \frac{\prod_{n=1}^{n_w} P(d_n=x_n|R=1,u_k)}{\prod_{n=1}^{n_w} P(d_n=x_n|R=0,u_k)} \times \frac{P(R=1|u_k)}{P(R=0|u_k)}
\end{aligned}
$$

Probabilistic methods: Basic model

- And finally λ is proportional to

$$\lambda \propto \frac{\prod_{n=1}^{n_w} P(d_n = x_n | R = 1, u_k)}{\prod_{n=1}^{n_w} P(d_n = x_n | R = 0, u_k)}$$

- This is really a naive Bayes classifier
- The $P(d_n = x_n | R = 1, u_k), P(d_n = x_n | R = 0, u_k)$ are easy to compute
  = Likelihoods estimated by frequencies

Probabilistic methods: Basic model

- The $P(d_n = x_n | R = 1, u_k)$, $P(d_n = x_n | R = 0, u_k)$ are easy to compute
  - – Likelihoods estimated by frequencies
- They are estimated by the proportion of documents containing the word w_n among relevant/irrelevant documents

Probabilistic methods: Conclusion

- The binary independence probabilistic retrieval model makes strong asumptions about independence of word occurrence
- More sophisticated models are available
  - For instance Poisson models can be used in order to take account of the number of words appearing in the document
  - We can also take account of second-order interactions between words (correlations)

Assessment of documents retrieval systems

- In general, we compute two measures:
  - The precision
  - The recall
- As well as the F-measure

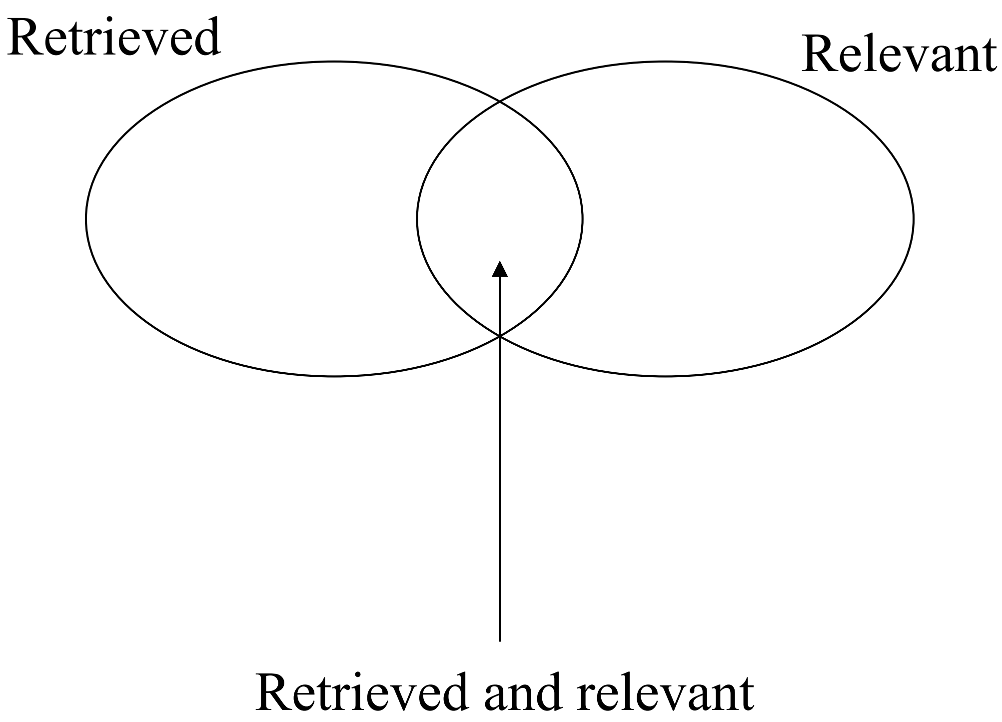
Assessment of documents retrieval systems

- The precision measure estimates the percentage of relevant retrieved documents in the set of all retrieved documents
  - Precision indicates to which extend the retrieved documents are indeed relevant

Assessment of documents retrieval systems

- The recall measure estimates the percentage of relevant retrieved documents in the set of all relevant documents
  - Recall indicates to which extend the relevant documents are indeed retrieved

Assessment of documents retrieval systems



Assessment of documents retrieval systems

- There is a trade-off between precision and recall

- The F-measure, taking both precision and recall is

$$F = 2\,(precision \times recall)/(precision + recall)$$