INSTITUTO MILENIO
SISTEMAS COMPLEJOS DE INGENIERIA

INGENIERÍA INDUSTRIAL
UNIVERSIDAD DE CHILE

fcfm
FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Chapter 5

Mining the Content

PROFESSORS

Juan D. Velásquez
Víctor Rebolledo L.

Outline

- Introduction
  - Web Content Mining (WCM)
  - Building the vector space model
- Classification of web page content
- Clustering for group having similar web page text content
- Content Mining Applications
  - WEBSOM
  - Automatic web page text summarization
  - Extraction of key-text component from web pages

Section 5.1

- Introduction

Web Content Mining (WCM)

- The goal is to **find useful information** from the web content.
- In this sense, WCM is similar to **Information Retrieval (IR).**
- However, web content is not only free text, other objects like *pictures, sound and movies belong also to the content*.
- There are **two main areas** in WCM :
  - the **mining of document contents** (web page content mining)
  - the **improvement of content search** in tools like *search engines* (search result mining).

WCM (2)

- Web content
  - text, image, audio, video, metadata and hyperlinks.
- Information Retrieval View (Structured + Semi-Structured)
  - Assist / Improve information finding
  - Filtering Information to users on user profiles
- Database View
  - Model Data on the Web
  - Integrate them for more sophisticated queries

WCM (3)

- Developing **Web query systems**
  - WebOQL, XML-QL
- Mining multimedia data
  - Mining image from satellite (Fayyad, et al. 1996)
  - Mining image to identify small volcanoes on Venus (Smyth, et al 1996) .

Issues in Web Content Mining

- Developing intelligent tools for IR
  - Finding **keywords** and **key phrases**
  - Discovering **grammatical rules** and collocations
  - **Hypertext classification/categorization**
  - Extracting **key phrases** from text documents
  - Learning extraction **models/rules**
  - Hierarchical **clustering**
  - **Predicting (words) relationship**

The Web Text

- In order to analyze we **need to process** before to use it.
    - Document free text (without tags)
    - Stop-word filtering (to explain later)
    - Stemming algorithm (to explain later)
- All these procedure are performed to have a **clean list of word** that represent a web page.

Representation of a web page: The word page vector (wp)

- Each web page can be consider as a **document text** with tags.
- Applying filters, the web page is transformed to the **feature vector**.
- Let $P = \{p_1, ..., p_Q\}$ be the **set of *Q* pages** in a web site.
- The i–th page is represented by

$$wp^i = \{wp^i_1, ..., wp^i_R\} \in WP$$

with *R* the number of words after a **stop word** and **stemming process** and WP the set of feature vectors.

Representation of a web page: The word page vector (wp) (2)

- Meaning of the k–component (wp^i_k) of the feature vector: "The importance of the word k on the page i"

- With this model we have a way to
  - Have a **numeric representation** of text
  - **Compare** 2 pages
  - Allows to use more **complex battery of mathematical tools** for text analysis and mining.
  - First (approximate) approach to **representing the meaning of a page** by list of words.

Building The Vector Space Model

Measuring Web Page Content

Web text content

- From different web page content, special attention receive the **free text**.
- For the moment, a searching is performed by using key words.
- It is necessary to *represent the text information in a* ***feature vector***, before to apply a mining process.
- The representation must consider that *the words in the web page don’t have the same importance.*

Stage of the process

1. **Parsing** the web page content
2. Identifying the text semantic: **Stemming**
3. **Calculating** the feature vector
4. Data mining **algorithm** application:
   - clustering and similarity measure.

First Stage:

Parsing the content (Tokenizer)

- **Extract text content**:
  - individualizing each word contained in the document.
- A web document is based on **HTML tags**
- The usual procedure is to extract all the free text word *avoiding all the HTML tag*.
- **Filtering**:
  - Also commonly removing **stop word** like :
    - “the”, “a”, “by”, “he”, “she”, “behind”, “above”, “below”, …
- **Result:**
  - **A raw list of word for each page.**

Stop Word:

Why we don't we don't need them

- A **full list** of them (in English):

  http://dev.mysql.com/doc/refman/5.0/en/fulltext-stopwords.html
- **The Semantic**:
  - the study of the meaning in a communication process.
- In vector space model:
  - We need to identify the **importance of a word in a text,** from the point of view of the **semantic.**
- Our first approximation: *"Stop Word doesn't contribute to the semantic of a text."*
- **It is an approximation.**
- **We cannot capture the semantic** of *"the main course was explained in the thai food course ".*

Next stage:
Identifying the semantic.

- There are words that have **“similar meaning”**: *connect, connected, connecting, …*
- It is necessary to associate a **unique identifier** of the semantic content for them.
- **Word stemming**:
  - A way to generate word with unique semantic.
- {connect, connected, connecting} -> “connect”

Web text content:
Stemming

- First work on 1968 Lovins.
- **Martin Porter** http://tartarus.org/martin/
- **Stemming**:
  - The process for removing the commoner morphological and inflexional endings from word.
- This process is widely used in *Information Retrieval Process Systems.*
- This process has the intention **to extract the semantic root** of word in a document, in order to have a more *simpler description* of the semantic of the text content.
- Usually the process works in language like **English**, others like **Arabic**, **Hebrew** are more difficult to stem.

Stemming:
The Porter Algorithm

1. Take the next **word** on the text
2. Determine if it has suffixes, like: –ED, –ING, –ION, –IONS, … and others
3. Lookup in the exception rule list if the **word** is present and then apply the rule
   - Ex: ran->run
4. If not then cut the suffix and return the remaining part
   1. Ex: connections -> connect
5. Insert the **new word** on a list and return to the ***step 1*** if another word remain on the text; if not then finish and return the list of processed **new word**

Porter Algorithm:
Implementation

- From the Porter page:

http://tartarus.org/martin/PorterStemmer/index.html

- **Snowball** Library:
  - JAVA available
  - More robust support
  - Other languages supported than **English**, like **Spanish.**

http://snowball.tartarus.org/download.php

The next stage:
Calculating the word page vector

- We have a ***clean list of stemmed word*** for each page.
- *¿How we can calculate the numeric importance of a word on the page?*
  - **Binary measure**:
    - 1 if the word k is present on i, 0 if not.
  - **Frequency measure**:
    - the *relative frequency* of the word k on the page i vs.. all the pages.
  - **Others measure**:
    - next page.

Web page:
Vector representation

- Its **vector representation** would be a matrix of *RxQ*.
- *Q* is the number of pages in the web site and *R* is the number of different words in *P*.

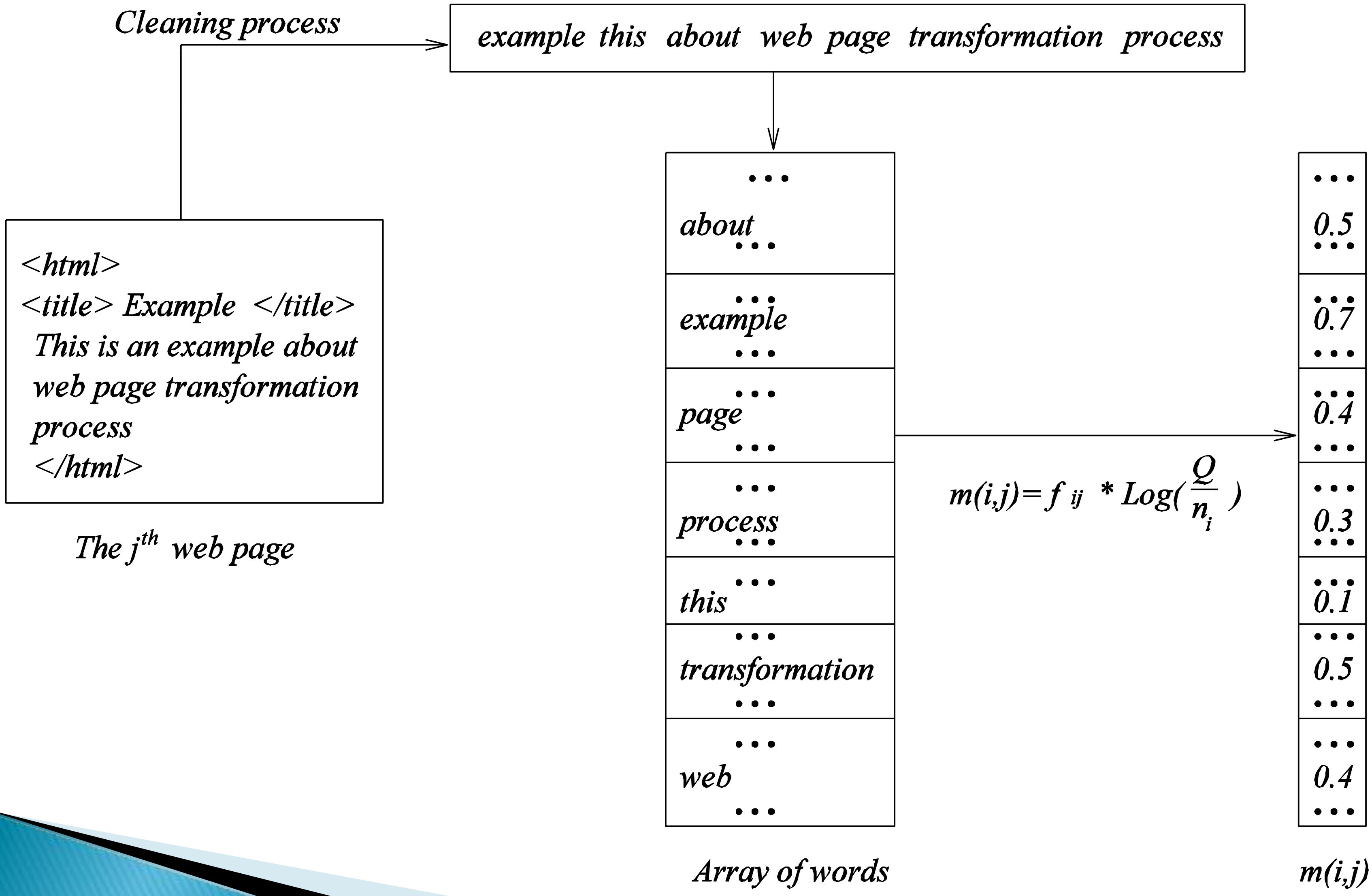
	Word	1	2	…	Q
1	advise	1	0	…	1
2	business	0	1	…	0
.	…	.	.	…	.
.	…	.	.	…	.
.	…	.	.	…	.
.	…	.	.	…	.
.	…	.	.	…	.
R	zambia	1	0	…	0

Processing the web site: Vector space model

- The model associates a **weight to each word** in the page, based on its *frequency* in the whole web site.
- Let n_i the number of pages with the word i and Q the amount of pages, a simple estimation of the relevance of a word is
$$wp^i_j = n_i / Q$$
- The ***inverse document frequency*** $wp^i_j = IDF = \log(Q / n_i)$ can be used like a weight.
- *A variation of the last expression is known as TF*IDF*

where f_{ij} the number of occurrences of word i in the document j $wp^i_j = TF * IDF = f_{ij} * \log(Q / n_i)$

Vector space model: a graphic example



$$m(i,j) = f_{ij} * Log(\frac{Q}{n_i})$$

Vector Model, better approaches

- Based on the **TF*IDF weights**:

  $wp_{ij} = f(i,j) * \log(Q/n_i)$

- A more **parameterized approach**:

  $wp_{ij} = f(i,j) *(1 + sw_i)* \log(Q/n_i)$

- Where sw_i is an **additional weight** that for the i-th word.
- In this way, the vector sw_i allows to include **semantic information** about *special word* in the page like tagged word in HTML (bold, italic, titles,…).

Vector Model, better approaches (2)

- Another suggestion is

$wp_{iq} = (0.5 + [0.5 * freq(i,q) / max(freq(l,q)]) * log(N / n_i)$

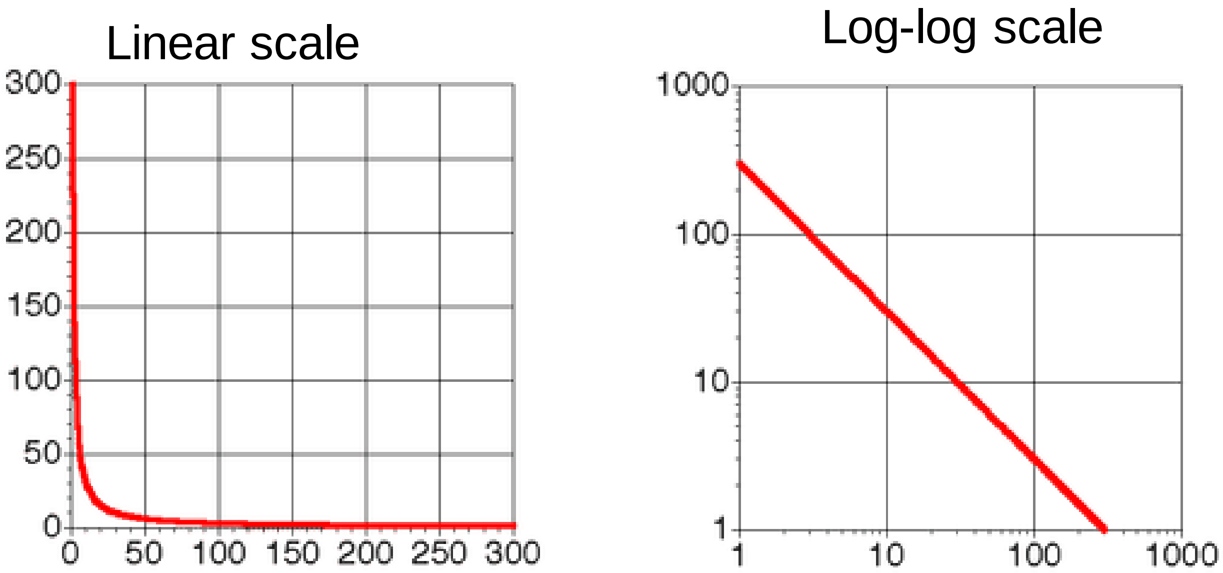
- This model is very good in practice:
  - TF*IDF works well with **general collections**
  - **Simple and fast to compute**
  - *Vector model* is usually as good as the *known ranking alternatives*
- **Why? :** These result are validated by **empirical** experiment.

The zipf law: An intuition of the logarithm

- From a 1945 study on **free text** in a document repository.
- Shown that the graph
- **Log(Frecuency of use of a word) vs. – Log(Number of Word)** is Linear!!
- This rule was verified on **several other document repository**, even in web text.
- That mean that **different word distribution** on a text follows a power law:

$$P(n) \propto n^{-\beta}$$

Zipf law



Linear scale

Log-log scale

Frequency
of appearance

Numerated
sorted word

- If the **text frequencies follows a power laws**, then the weight measure like IDF retrieve us the a *narrow approach to the most important* **(=most probable**) word in a linear way.

The next stage:
data mining

- Now we have vectors that represent pages and word importance over them.

$$wp^{i} = (wp^{i}_{1},\ldots,\ wp^{i}_{R}) \in WP$$

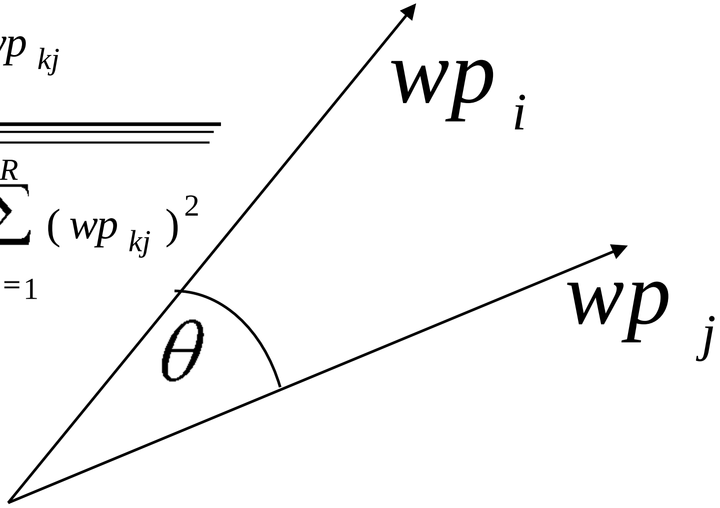
- We **have to process** this data
- Data mining techniques applies
- … But we need to *define a way to compare them.*

Comparing web pages:
Similarity Measure

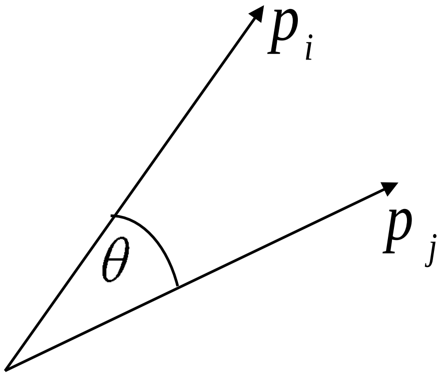
$$wp = (wp_{ij}) = f_{ij} * \log(\frac{Q}{n_i})$$

$$wp_i \rightarrow (wp_{1i},..., wp_{Ri}) \qquad wp_j \rightarrow (wp_{1j},..., wp_{Rj})$$

$$dp(wp_i, wp_j) = \cos\theta = \frac{\sum_{k=1}^{R} wp_{ki} wp_{kj}}{\sqrt{\sum_{k=1}^{R} (wp_{ki})^2} \sqrt{\sum_{k=1}^{R} (wp_{kj})^2}}$$



What is a Similarity Measure?

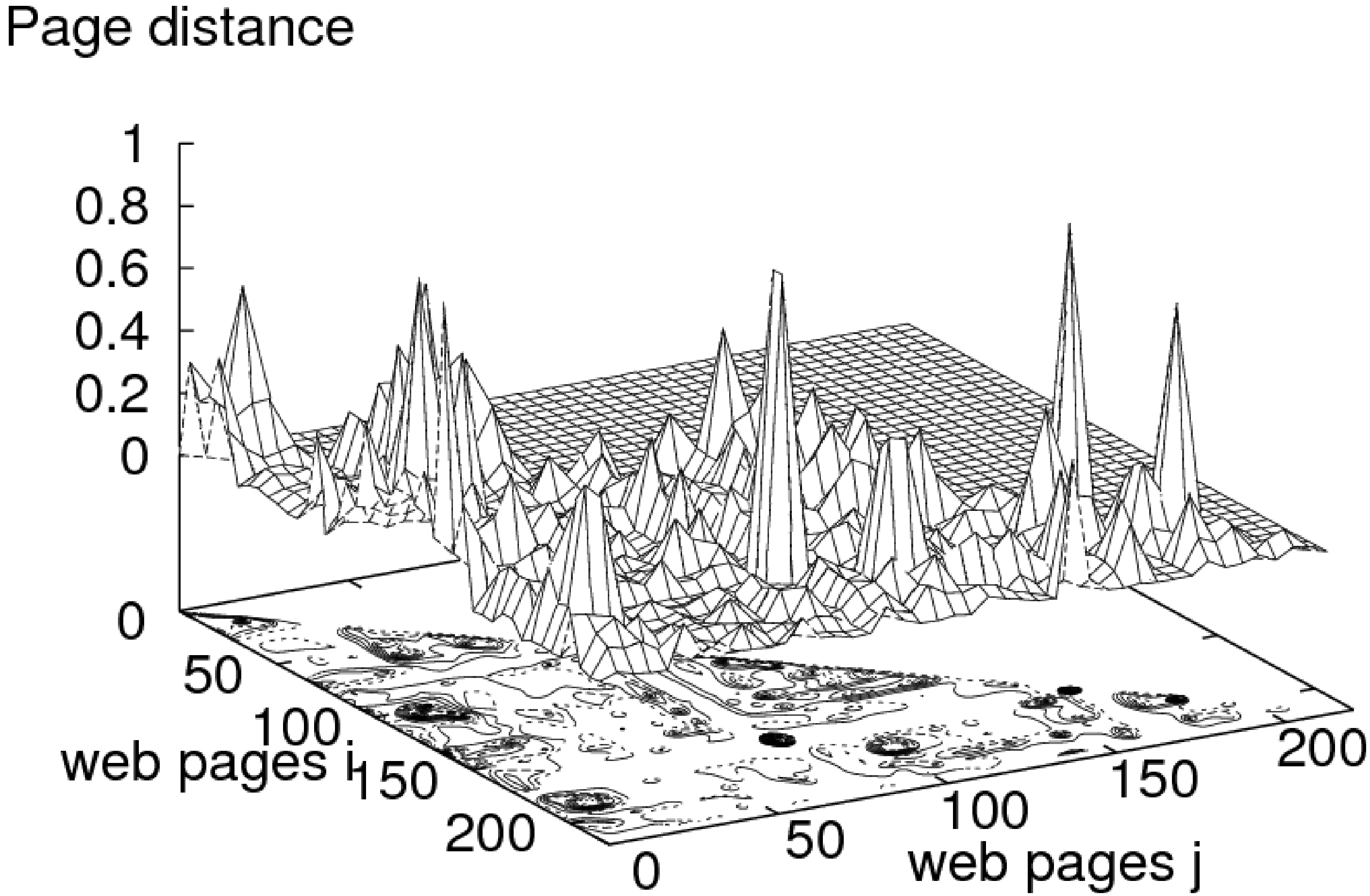


- Any method for **grouping needs to have an understanding for how similar observation are to each others** (*clustering*).
- *You don't need to have triangular inequality property*.
- In the case of the cosine measure, we have the benefits that is scale invariant. The **Euclidean distance** function doesn't have this property!! That is:

$$dp(wp_i, wp_j) = dp(\lambda wp_i, \lambda wp_j)$$

- *We need this property because the **units** of the wp values are NOT important for the TEXT PROBLEM.*

Similarity measure visualization



We are ready for mining

- With the word vector we give a **numeric meaning of pages**.
- *With similarity measure we could compare word vectors.*
- Then data mining algorithm will use as **feature the word vector.**
- Some aspect of the problem:
  - Business applications
  - To label or not to label (Supervised/Unsupervised)
  - *Which algorithm to use?, How to compare the results?*

Business Applications

- Decisions support in CRM:
  - Customer **text complain** analysis
  - The correlation between the **number of satisfied customer and text from them** (emails, messages, etc., …)
- Personalization's in ecommerce
  - Suggestions based on text personal information, messages, emails, text complain.

Business Applications (2)

- Bank customer messages (or email) repository.
  - **Analysis of customers requirement** (urgencies, request, insult, …)
  - Bank management need “to know” what are the **principal problems on the business**.
  - **Anticipating** problems, **retaining** customers.
  - Allow to **modify the site** *in order to cover new and demanding systems.*

Business Applications (3)

- Online Movies recommender system like http://www.netflix.com/
- **Based on your personal history** and **personal text info**, **the system recommend a movie**.
- A prize of **1 million US$** for the *best algorithm for recommendation.*
- http://www.netflixprize.com/
- In 2008 No candidate has already win the contest! They release **2Gb of data to train the model**.
- **WHAT ARE YOU WAITING FOR!**

Supervised vs. Unsupervised

- Supervised algorithm:
  - Like regression
  - Better adjustment
  - Over fitting issues
  - Be careful with the training set (known labels)
    - Validation set test for adjustment of parameters
    - Test set to measure the quality of the adjustment.
    - Feature selection avoiding curse of dimensionality.
  - Allow to have risk minimization: using the parameter of the model adjusting to the minimal risk or cost function results ().

Supervised vs. Unsupervised (2)

- Unsupervised Algorithm:
  - There is no training set with known classification!!
  - It really discover hidden information.
  - Useful when data are to large in order to examine and label, like survey's.
  - Need human expert verification of the results that sometime could be noisy.
  - Once the expert confirm a correct labeling, we have found natural documents aggregation of data.

Supervised vs. Unsupervised (3)

- In the text mining context:
  - Not all the times we have a clear labeling of the problem. Most of the time the labels need to be discover.
  - When we have some classification of document the supervised method are more precise than the others. But if we have few training examples over fitting issues could be important.

¿Which algorithm to use?: Statistical accuracy

- Experimentation: A general automatic solution to all problem doesn’t exist. YOU NEED TO EXPERIMENT WITH THE ALGORITHM FIRST.
- Each particular algorithm have pro/cons issues. It an statistical fallacy to use a favorite algorithm.
- Try to use a “diverse” set of algorithm, and your results will be statistically credible.
- There are meta-algorithm that perform better than each individual one:
  - Bagging or Bootstrap Aggregation
  - Boosting
  - Co-training

¿Which algorithm to use?: Meta Algorithm (1)

- **Bagging or Bootstrap aggregating**:
  - Given a Training Set T, we select random subsets $S_i \subseteq T, i=1 \ldots N$
  - For each S_i subset we train N models
  - The final model is the average of the output of the N model. For classification the average correspond to the “majority voting”.
- Bagging Properties:
  - Improve classification an regression accuracy
  - Reduce variance
  - Help Avoiding Over fitting
  - Doesn’t work if the training set is small.

¿Which algorithm to use?: Meta Algorithm (2)

- **Boosting**:
  - Having a set of M different algorithm for data mining.
  - The output result of each algorithm is averaged to produce the final result. In the case of classification the averaging is by "majority voting".
- Boosting Properties:
  - Same than bagging.
  - Works also with small training set.
  - The result is always better than individual algorithm approach.
  - The way to average (or combine) the algorithm could be parametric. Like linear combination that call LPBoost, AdaBoost (Adaptive Boost) where the final result is found iterating over the best averaging result.

¿Which algorithm to use?: Meta Algorithm (3)

- **Co-Training**:
  - Blum, Mitchell, "Combining labeled and unlabeled data with co-training”, COLT 1998.
  - The training set size is small.
  - We can still split the training set in K subset.
  - We train K models on these K training subset.
  - The resulting models of these training processes are used to obtain new labels in order to grow-up the training set.
  - Repeat until the expected amount training data has been complete
- **Co-Training Properties**:
  - Solve partially the over fitting issue of small training set.
  - Error of the method are lowers than the others methods.

¿How to compare performance of algorithm on a problem?

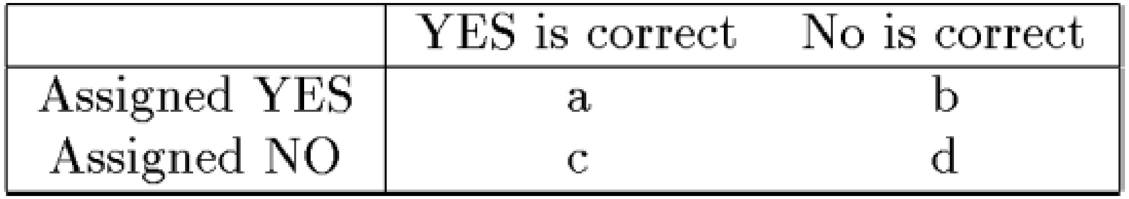
- Now we have several algorithmic methods for machine learning.
- The problem is to have a methodology to compare the performance of them in an specific example.

Performance Measures

- Training Set: The set from the classifier is constructed.
- Test Set: The set from we measure the quality of the classifiers.
- Breakeven point: Threshold a confidence value for accepting the declaration of the a class label.
- Important values:
  - CLF: Correct Label found (= threshold accepted and label is correct in the test set)
  - TCL: Total Correct Label (= the number of correct label)
  - TLF: Total Label found (= the number of label that commit the threshold)
  - TI: Total Incorrect cases.
  - N: Total cases in Test.
  - Θ: The actual threshold

Performance Measures (2)

- R: Recall number = CLF/TCL
- P: Precision = CLF/TLF
- E: Error= TI/N
- Example: If we have 2 classes



	YES is correct	No is correct
Assigned YES	a	b
Assigned NO	c	d

- R=a/(a+c) if a+c>0 otherwise R=1
- P=a/(a+b) if a+b>0 otherwise P=1
- E=(b+c)/N ; of course N=a+b+c+d

Performance Measures (4)

- We interpret R as the probability of that a document in the class YES is classified in this class. P is the probability that the document classified in the class YES truly belong to it.
- => We want that both probability be the same. Then The threshold Θ value that we want is the one that do R=P.
- Θ: Is interpreted as the MEASURE of performance of the model.
- This performance measure is accepted in international papers as standard.

Performance Measures (5)

- For more than 2 class label, we use MICROAVERAGING that the previous calculus are performed on each label separately.
- From this we obtain several threshold that equal P and R for each column. We take as the performance value the Minimum Θ^* (worst case) of them.
- Θ^* : The performance measure of the algorithm.

Section 5.2

- Classification of Web Page Content – Using classifiers on text

Text Classification

- Assign a label to a text. Ex:
  - Classify as a Political Document
  - Classify as a particular Product Marketing page
  - Classify as a Study on Molecular Biology
  - .. Etc
- The label could be:
  - Pre-defined: By the direction of the study -> SUPERVISED LEARNING
  - Unknown: To discover! -> UNSUPERVISED LEARNING

Practical Uses

- Extracting Domain Specific Information
  - Grouping document in different domain.
  - Finding the most representative
- Learn reading interests of users
- Automatically classification of e-mail
- On-line New Event Detection:
  - Opinion blog scanners.
  - Social activities detection.

Classification

- A text classifier . Given a document d and return a scalar value with a category $c_i \in C / \mathrm{Y}\, c_i = C$ [Sebastiani99].

- The function is known as “Categorization Status Value"“

$$CSV_i : D \rightarrow [0,1]$$

- The $CSV_i(d)$ takes different expressions, according with the classifier in use.
- For instance, it can be a probability approach [Lewis92] basis on Naive Bayes theorem or a distance between vectors in a r-dimensional space [Schutze95].

Classification (2)

- The classification was implemented by semi-automatic of full-automatic [Asirvatham05] approaches, like:
  - Neighbor [Kwon03]
  - Bayesian models [McCallum98]
  - Support Vector Machines [Joachims97]
  - Artificial Neural Networks [Honkela97]
  - Decision Trees [Apte04] .
  - MAXENT algorithm [McCallum99]

Classification (3)

- The web pages classification algorithms can be grouped in [Asirvatham05] :
  - Manual categorization -> too expensive!!!
  - Applying clustering approaches. Previous to classify the web pages, a clustering algorithm is used to find the possible clusters in a training set.
  - Meta tags: It use the information contained in the web page tags (<META name=``keywords"> and <META name=``description">).
  - Text content based categorization.
  - Link and content analysis: It is based on the fact that the hyperlink contain the information about which kind of pages is pointed (href tag)

Issues on text classification

- The size of the space of text different possible classification is very large.
- Very high number of possible "dimension", records are not structurally identical.
- Very subtle relationship between concept in text
- Ambiguity and text sensitivity: "the main course was explained in the thai food course"
- Need a very large labeled example for training the models.

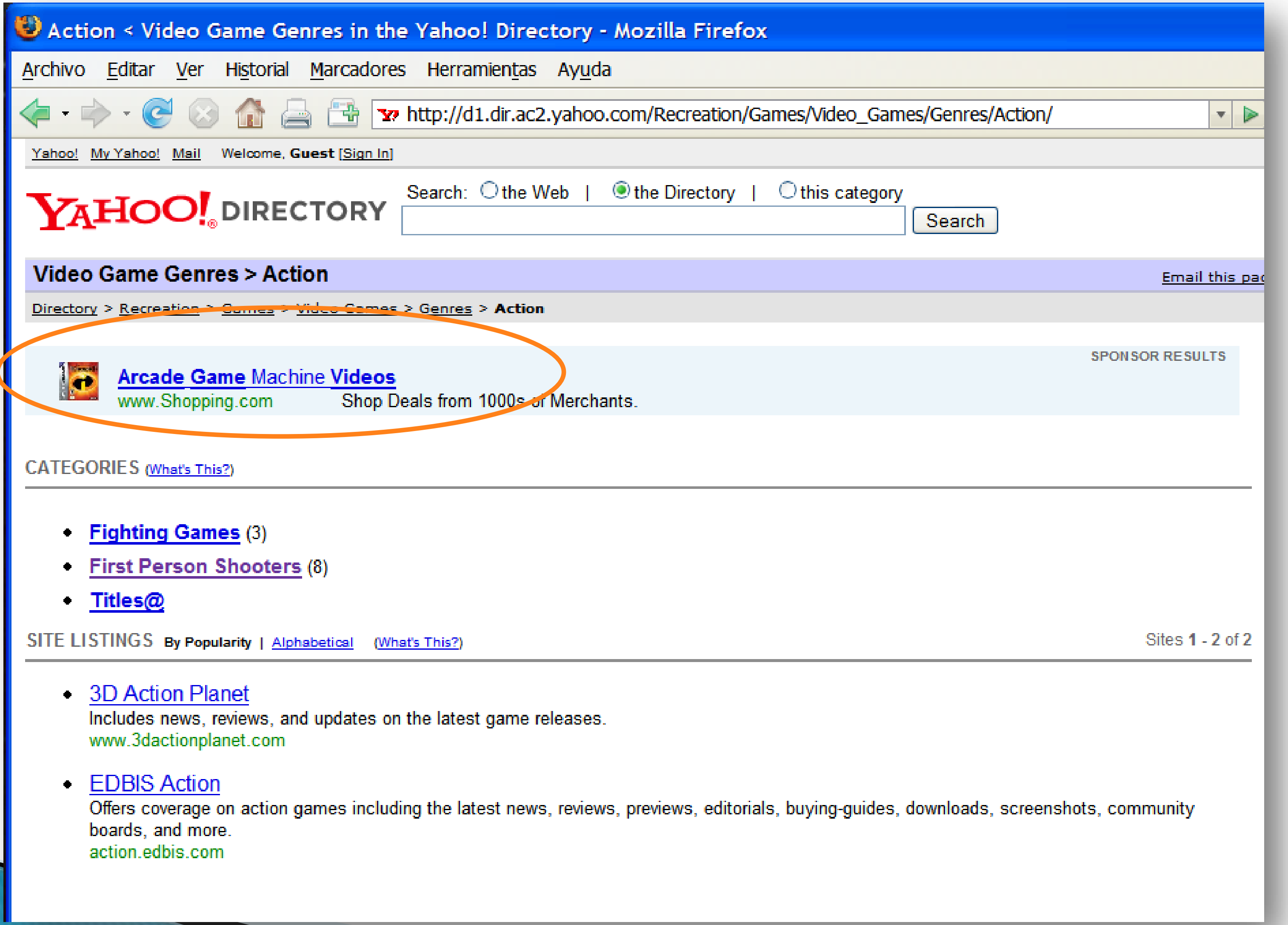
Further refining for text mining

- Semantic Processing
  - Extracting meaning
  - Named entities extraction (people names, company names, locations, …)
  - Phrase recognition
  - Tagged phrases
- The semantic process result in more complex numeric vector structure with nested relationship (tree-like).

Hierarchies:
Natural Human Classification

- ¿A number is something useful as classification? -> No
- Human need always to have a “context” for classification of something.
- This “context” contains classes, but the content also need to have a “context”.
- That create a tree-like or directory like hierarchy of contexts.
- In the web these hierarchies are called “Directory”

Hierarchies: Yahoo Directory



Action < Video Game Genres in the Yahoo! Directory - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda
http://d1.dir.ac2.yahoo.com/Recreation/Games/Video_Games/Genres/Action/
Yahoo! My Yahoo! Mail Welcome, Guest [Sign In]
YAHOO! DIRECTORY
Search: ○ the Web | ● the Directory | ○ this category
Search
Email this pac

Video Game Genres > Action

Directory > Recreation > Games > Video Games > Genres > **Action**

SPONSOR RESULTS

Arcade Game Machine **Videos**
www.Shopping.com Shop Deals from 1000s of Merchants.

CATEGORIES (What's This?)

- **Fighting Games** (3)
- **First Person Shooters** (8)
- **Titles@**

SITE LISTINGS **By Popularity** | Alphabetical (What's This?) Sites **1** - 2 of 2

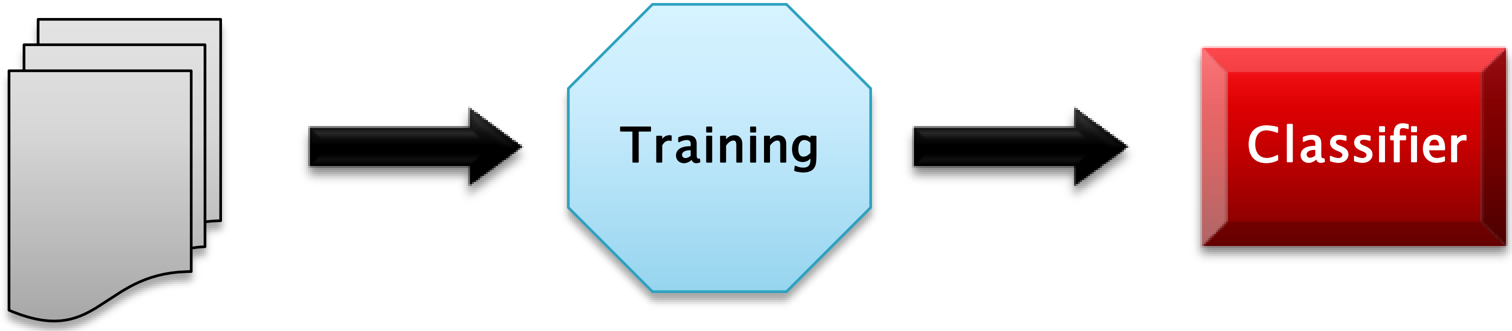
- 3D Action Planet
  Includes news, reviews, and updates on the latest game releases.
  www.3dactionplanet.com
- EDBIS Action
  Offers coverage on action games including the latest news, reviews, previews, editorials, buying-guides, downloads, screenshots, community boards, and more.
  action.edbis.com

Web Hierarchy are Web Directories

dmoz open directory project

- Example:
  - **Yahoo Directories, Google directories.**
  - Each time that we perform a search the result appears as belonging to a directory structure.
- An open source web directory information is available on the DMOZ project (from Netscape) http://www.dmoz.org/
- You can download free 1 Gb of human web classification on RDF format.
- Some software allows to parse this format and translate to database format. The project dmoz2mysql http://sourceforge.net/projects/dmoz2mysql/

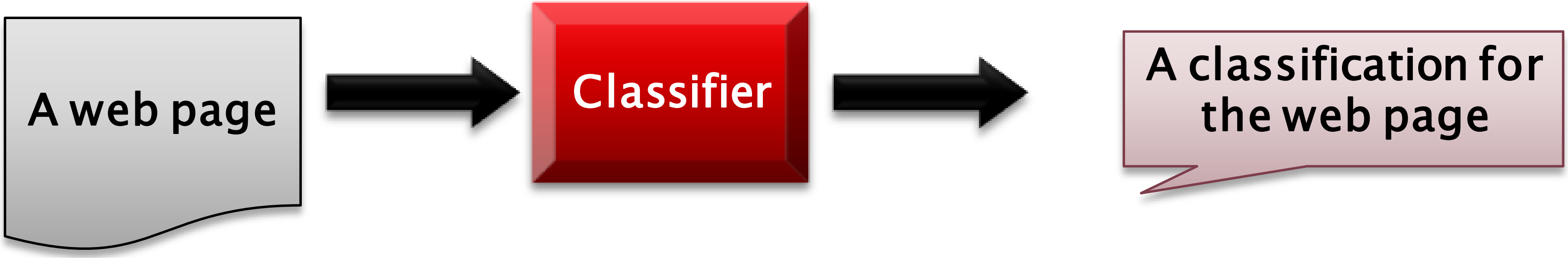
Web directory importance: Building Training Set (Supervised).



Training

Classifier

Labeled set of web pages
from web directory classification



A web page

Classifier

A classification for the web page

Algorithm for text classification

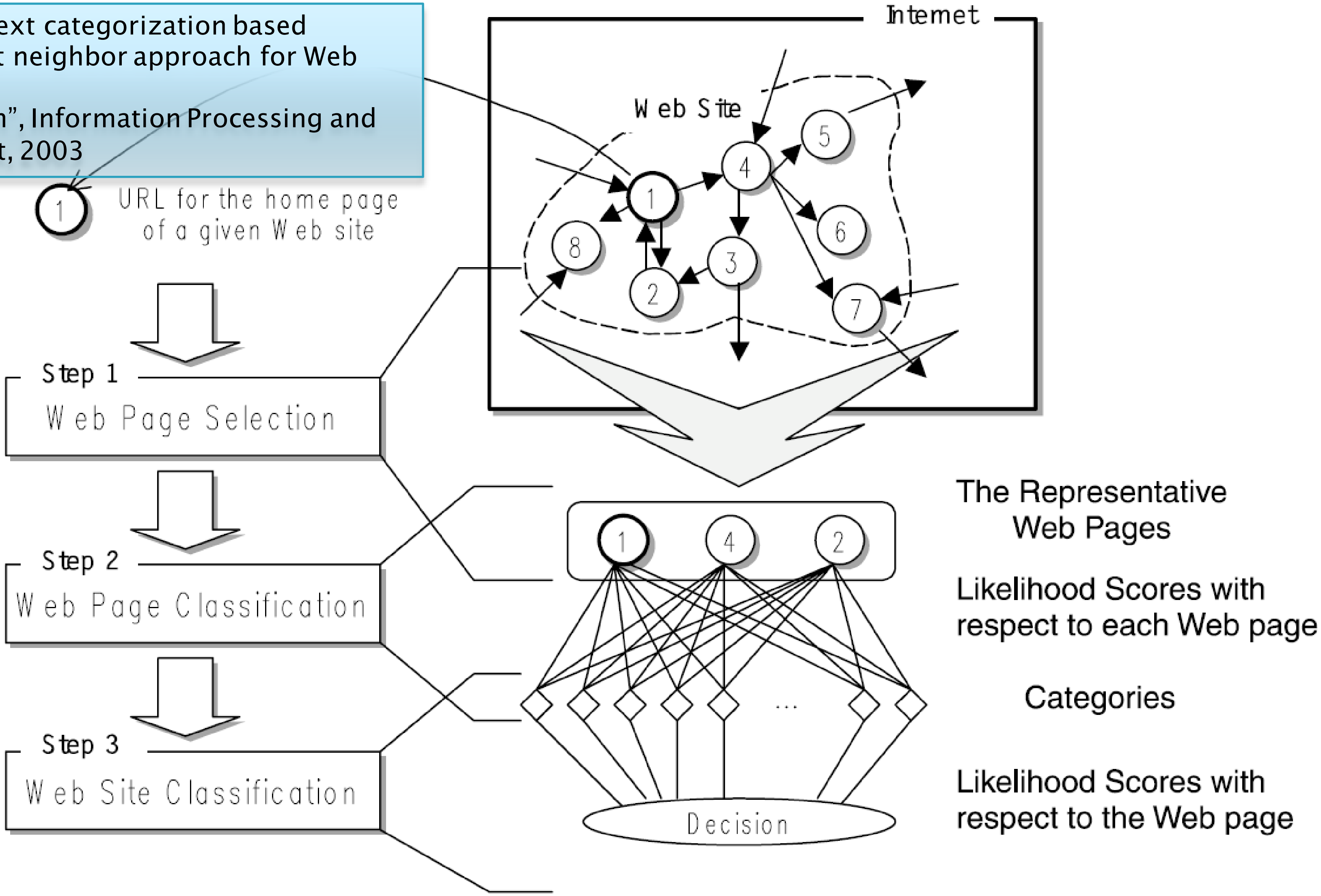
A revision of the current literature

K-means clustering

- “Text categorization based on k-nearest neighbor approach for Web site classification”, O. Kwon, J. Lee, 2003.
- Given:
  - Set of word vector (TFIDF value)
  - Similarity measure (cosine)
  - An estimation of the number of classes
- For each class initialize randomly the Centroid.
- Iterate assigning the nearest group for each page and recalculate the Centroid.
- Finish when Centroid converge.

K-means clustering (2)

Kwon,Lee "Text categorization based on k-nearest neighbor approach for Web site classification", Information Processing and Management, 2003


Internet
Web Site
1
2
3
4
5
6
7
8
1
URL for the home page of a given Web site
Step 1
Web Page Selection
Step 2
Web Page Classification
Step 3
Web Site Classification
1
4
2
...
Decision
The Representative Web Pages
Likelihood Scores with respect to each Web page
Categories
Likelihood Scores with respect to the Web page

K-means clustering (3)

- New similarity measure are also explored.

$$sim\_mf\,(D_x, D_l) = \left(a + \frac{mf}{|D_x|} + \frac{mf}{|D_l|}\right)^{mf-1} sim\,(D_x, D_l)$$

- D: is a document, mf is the number of matching between document, a is a constant, sim(x,y) is the cosine measure.
- Reveals to improve the text clustering results.

Support Vector Machines for Text Classification

- T. Joachims, “*Text Categorization with Support Vector Machines: Learning with Many Relevant Features*”. 1997.
- Word Vector are sometimes very high dimensional, sometimes 10000 different keyword per document.
- **Feature selection**: the process that allows to choose the correct keyword (in this case) in order to have a low dimensional model.
- Chi-square, Information Gain, are commonly used.
- Word vector are also sparse.
- … but we could loose valuable information for clustering!

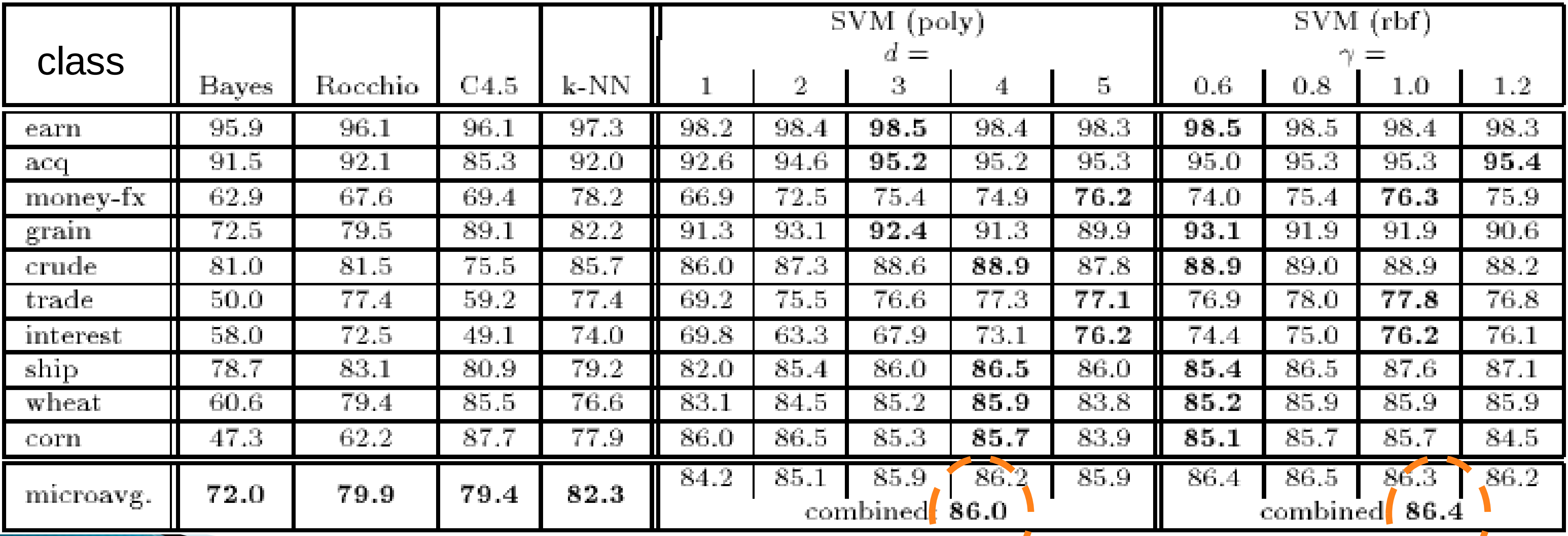
Support Vector Machines for Text Classification (2)

- SVM in short:
  - As a result define “hyperplanes” that separate the data in the different classes.
  - The “hyperplane” are defined to maximize the distance to the training point.
  - The methodology generalize to non-linear geometry, where the dot product between vector are nonlinear kernel function.
  - The resulting set of hyper-plane are not plane, there are curved hyper-surfaces

SVM for Text Classification

- SVM: Can handle high dimensional space.
- Comparing between method using microaveraging [Joachims, 1998] threshold performance measure:

Other classifiers Polynomial Kernel Gaussian Kernel



class					SVM (poly)					SVM (rbf)			
					$d =$					$\gamma =$			
	Bayes	Rocchio	C4.5	k-NN	1	2	3	4	5	0.6	0.8	1.0	1.2
earn	95.9	96.1	96.1	97.3	98.2	98.4	**98.5**	98.4	98.3	**98.5**	98.5	98.4	98.3
acq	91.5	92.1	85.3	92.0	92.6	94.6	**95.2**	95.2	95.3	95.0	95.3	95.3	**95.4**
money-fx	62.9	67.6	69.4	78.2	66.9	72.5	75.4	74.9	**76.2**	74.0	75.4	**76.3**	75.9
grain	72.5	79.5	89.1	82.2	91.3	93.1	**92.4**	91.3	89.9	**93.1**	91.9	91.9	90.6
crude	81.0	81.5	75.5	85.7	86.0	87.3	88.6	**88.9**	87.8	**88.9**	89.0	88.9	88.2
trade	50.0	77.4	59.2	77.4	69.2	75.5	76.6	77.3	**77.1**	76.9	78.0	**77.8**	76.8
interest	58.0	72.5	49.1	74.0	69.8	63.3	67.9	73.1	**76.2**	74.4	75.0	**76.2**	76.1
ship	78.7	83.1	80.9	79.2	82.0	85.4	86.0	**86.5**	86.0	**85.4**	86.5	87.6	87.1
wheat	60.6	79.4	85.5	76.6	83.1	84.5	85.2	**85.9**	83.8	**85.2**	85.9	85.9	85.9
corn	47.3	62.2	87.7	77.9	86.0	86.5	85.3	**85.7**	83.9	**85.1**	85.7	85.7	84.5
microavg.	**72.0**	**79.9**	**79.4**	**82.3**	84.2	85.1	85.9	86.2	85.9	86.4	86.5	86.3	86.2
							combined: **86.0**					combined: **86.4**	

SVM has the best score

MAXENT:

Maximum Entropy for text classification

- Nigam, Lafferty, McCallum “Using Maximum Entropy for Text Classification”. IJCA 1999.
- Promising technique prove to be significantly better than naïve Bayes techniques and sometimes better than SVM in the context of text classification.

MAXENT in text mining

- **MAXENT**: General technique for estimating probability distribution from data.
- **Concept**: Entropy is maximum when the probabilities are close to be uniform, in others word when we have the least information over the variables (tending to uniform distribution).
- **What the maxent do**: Maximize the entropy, but introducing the corresponding restriction of the problem.

MAXENT in text mining (2)

- Clusters (or class) are labeled by the class number c.
- The clusters are described by the degree of belonging of a document (d_i) to a class, by the conditional probability value: $P(c| d_i)$
- Need functions that associate a number to the document (d_i) and the class c: $f_k(d_i , c)$ we call them feature functions.
- The restriction: $E(f_k())= F_k$. The average values need to be known.
- Of course our well known vector space model give us a set of feature functions.

MAXENT in text mining (3)

- In order to simplify the equations, the conditional probability will be p_k and the feature function will be $f_{k,j}$, Then the optimization problem are:

$$\underset{p_j}{Max}\ S = -\sum_j p_j \log(p_j)$$

$$\sum_j p_j = 1$$

$$\sum_j f_{k,j} p_j = F_k$$

- The value F_k could be calculated as the simple average :

$$F_k = \frac{\sum_j^N f_{k,j}}{N}$$

MAXENT in text mining (4)

- The solution of the previous system (first order condition) are elegantly written by mean of the partition function Z.

$$p_j = \frac{e^{-\sum_k f_{k,j}\lambda_k}}{Z}$$

$$Z = \sum_j e^{-\sum_k f_{k,j}\lambda_k}$$

$$F_k = -\frac{\partial}{\partial \lambda_k} \log(Z)$$

- The last set of equation has to be solved in order to obtain the Lagrange multiplier λ_k .

MAXENT in text mining (5)

- The maxent equation consist in solving a non-linear set of equation obtaining the Lagrange multipliers and then the probabilities value.
- The method for solving these equation are usually iterative ones like: fix-point, gradient methods, etc.

MAXENT in text mining (6)

- The method have much better adjustment than others data mining tools in text mining.
- The intuition of this result are based on the probabilistic nature of the lexical properties of the text content.
- Like others methods, it suffers of the over fitting issue. This occur when the data training set is small or sparse.
- Implementation of the maxent algorithm are found in the McCallum free Java library MALLET:
- http://mallet.cs.umass.edu/

Section 5.3

- Clustering for group having web page text content

Clustering: Unsupervised method

- Clustering is a process of finding natural groups in a unsupervised way.
- To group web pages allows perform efficient searching task and semi-automatic or full-automatic document's categorizations.
- The clustering techniques need a similarity measure in order to compare two vectors by common characteristics [Strehl00] .

Clustering

- It is necessary a similarity of distortion measure to compare the vectors in a training set.
- For instance a simple distance like the angle's cosine between two pages in a vector representation.

$$dp(wp_i, wp_j) = \cos\theta = \frac{\sum_{k=1}^{R} wp_{ki}\, wp_{kj}}{\sqrt{\sum_{k=1}^{R} (wp_{ki})^2}\sqrt{\sum_{k=1}^{R} (wp_{kj})^2}}$$

Clustering (2)

- For document clustering, more complex and semantic based similarity have been proposed [Strehl00] .
- Let $C = \{c_1, ..., c_l\}$ be the set of clusters extracted from *WP*.
- Since the hard clustering point of view

$$\exists! c_k \in C \,/\, wp^i \in c_k$$

(it belong to a only one class)

Clustering (3)

- Whereas in soft clustering, a vector can belong to two or more clusters [Karypis99, Koutri04] .

- Several document clustering algorithms have appeared in the last years [Feldman95, Willet88] .

- An interesting approaches is the utilization of K-means and its variations in overlapping clusters, known as Fuzzy C-means [Jang97] .

- In these cases a word vector could belong to several classes.

Effect of different similarity measure on clustering

- Strehl, Gosh, Mooney, “Impact of Similarity Measures on Web-page Clustering”, AAAI-2000.
- We have already known the cosine measure. But there are others.

$$s^{(c)}(x_a, x_b) = \frac{x_a^{\mathrm{T}} x_b}{\|x_a\|_2^2 \|x_b\|_2^2}$$

Effect of different similarity measure on clustering (2)

- Pearson Measure: where $\bar{x}$ denote the average.

$$s^{(P)}(x_a, x_b) = \frac{1}{2}\left(\frac{(x_a - \bar{x}_a)^{\mathrm{T}}(x_b - \bar{x}_b)}{\|x_a - \bar{x}_a\|_2 \|x_b - \bar{x}_b\|_2} + 1\right)$$

- Jaccard Measure:

$$s^{(J)}(x_a, x_b) = \frac{x_a^{\mathrm{T}} x_b}{\|x_a\|_2^2 \|x_b\|_2^2 - x_a^{\mathrm{T}} x_b}$$

- Euclidean Measure:

$$s^{(E)}(x_a, x_b) = \|x_a - x_b\|_2$$

Effect of different similarity measure on clustering (3)

- Observation:
  - Euclidean measure have the worst results, even bad than a random clustering.
  - Cosine and Jaccard measure are the best ones.
- The Jaccard measure appears as an alternative.
- Its represents an approximation of the quotient information of (A and B) versus (A or B).

A relaxed problem: few labeled document.

- If we have a very few labeled document for training, the problem is close to unsupervised clustering.
- Nigam, McCallum, Thrun, Mitchel, "Text Classification from Labeled and Unlabeled Documents using EM", Machine Learning, 2000.
- Using Naïve Bayes Classifiers they infers the label of the others iteratively until the classifiers converge.

Section 5.4

- Content Mining Application

Applications

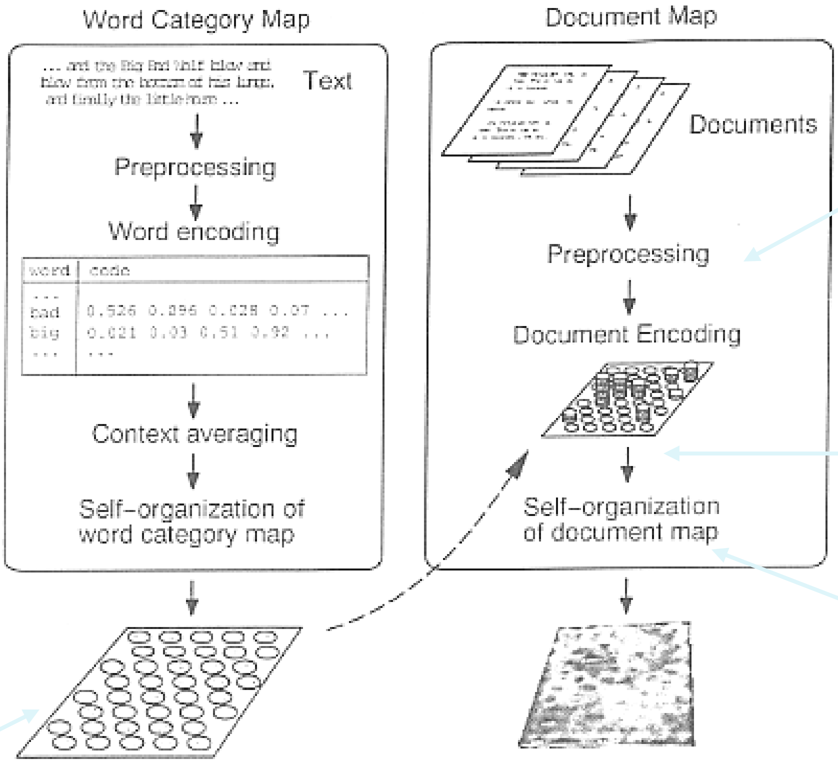
- WEBSOM
- Automatic web page text summarization
- Extraction of key-text component from web pages

WEBSOM

WEBSOM (1)

- It is a means for organizing miscellaneous text documents into meaningful maps for exploration and search.
- It is based on SOM (Self-Organizing Map) that automatically organizes documents into a two-dimensional grid so that related documents appear close to each other
- http://www.cis.hut.fi/websom

WEBSOM



All words of document are mapped into the word category map

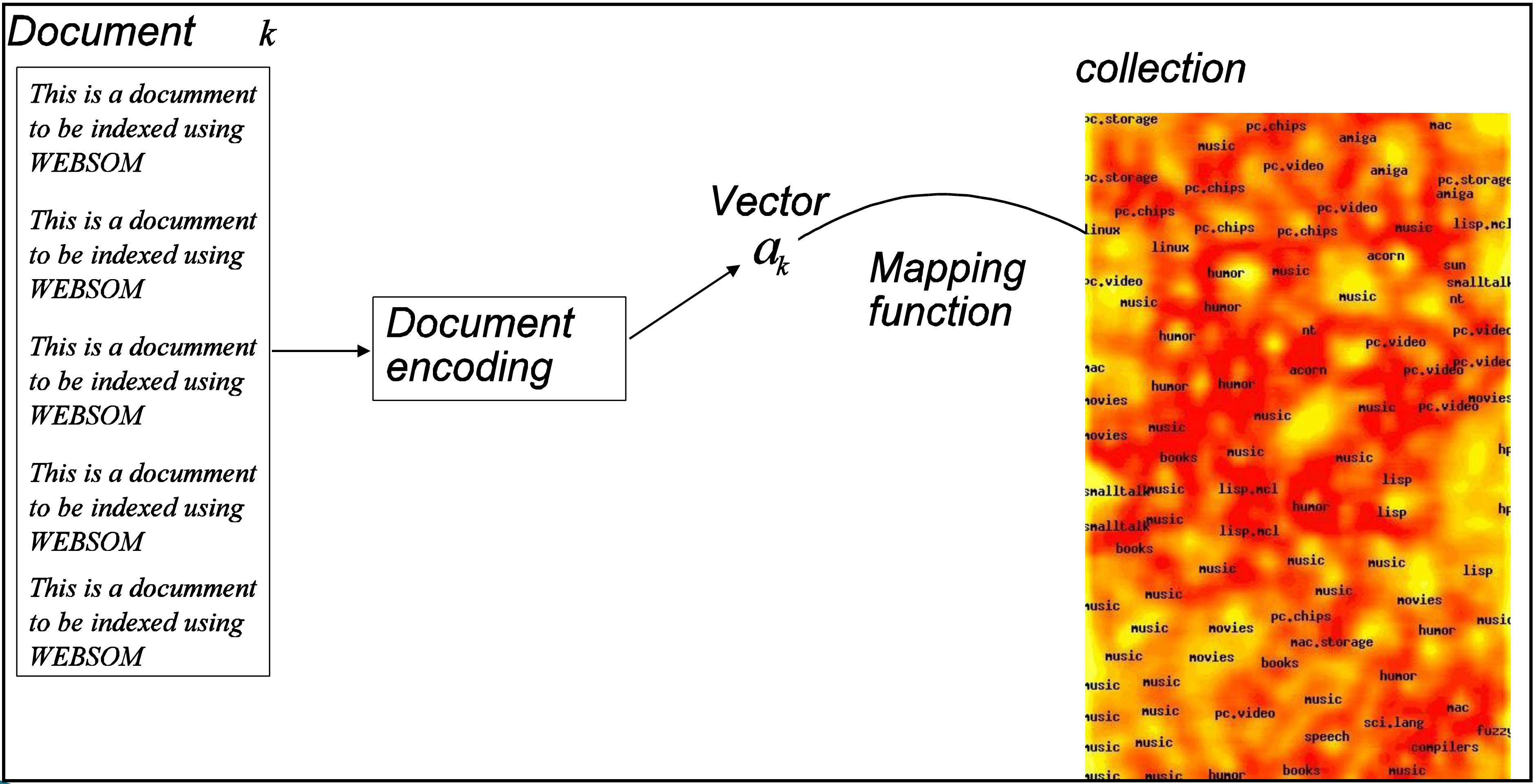
Histogram of "hits" on it is formed

Self-organizing map.
Largest experiments have used:
- word-category map
  315 neurons with 270 inputs each
- Document-map
  104040 neurons with 315 inputs each

Self-organizing semantic map.
15x21 neurons
Interrelated words that have similar contexts appear close to each other on the map

- Training done with 1124134 documents

WEBSOM



Document k

This is a documment to be indexed using WEBSOM

This is a documment to be indexed using WEBSOM

This is a documment to be indexed using WEBSOM

This is a documment to be indexed using WEBSOM

This is a documment to be indexed using WEBSOM

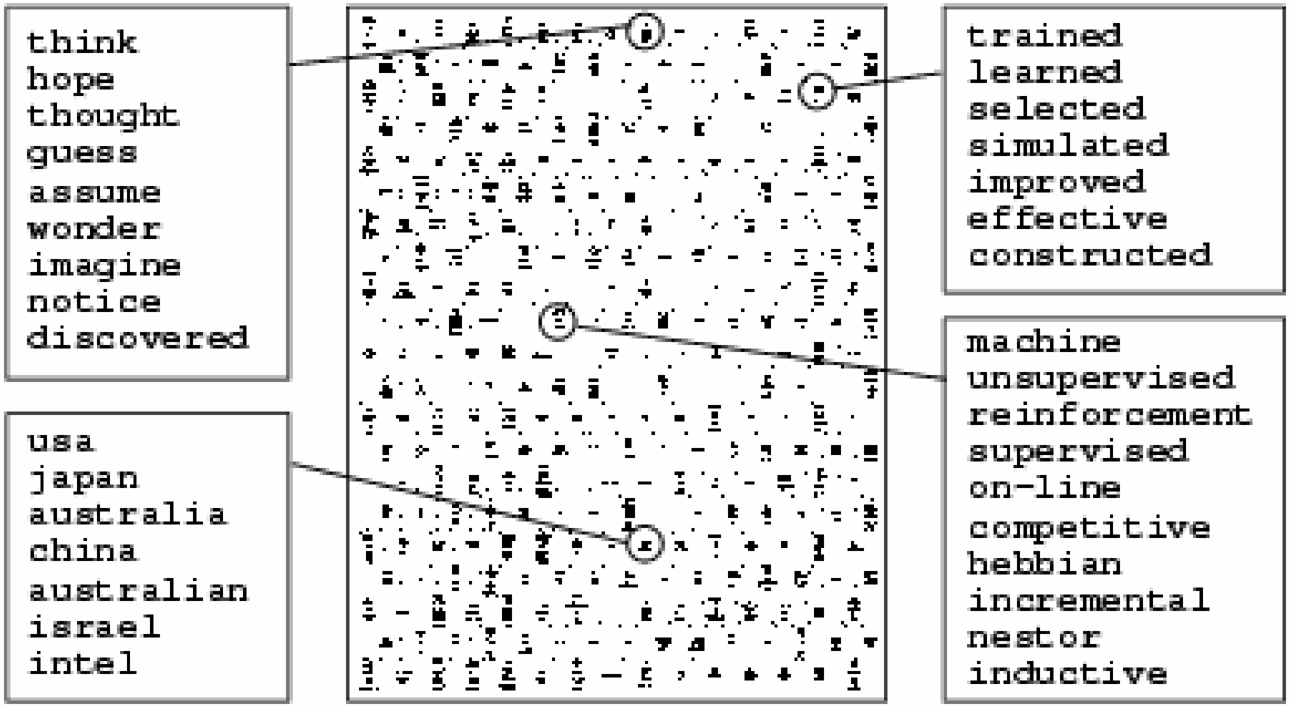
Document encoding

Vector a_k

Mapping function

collection

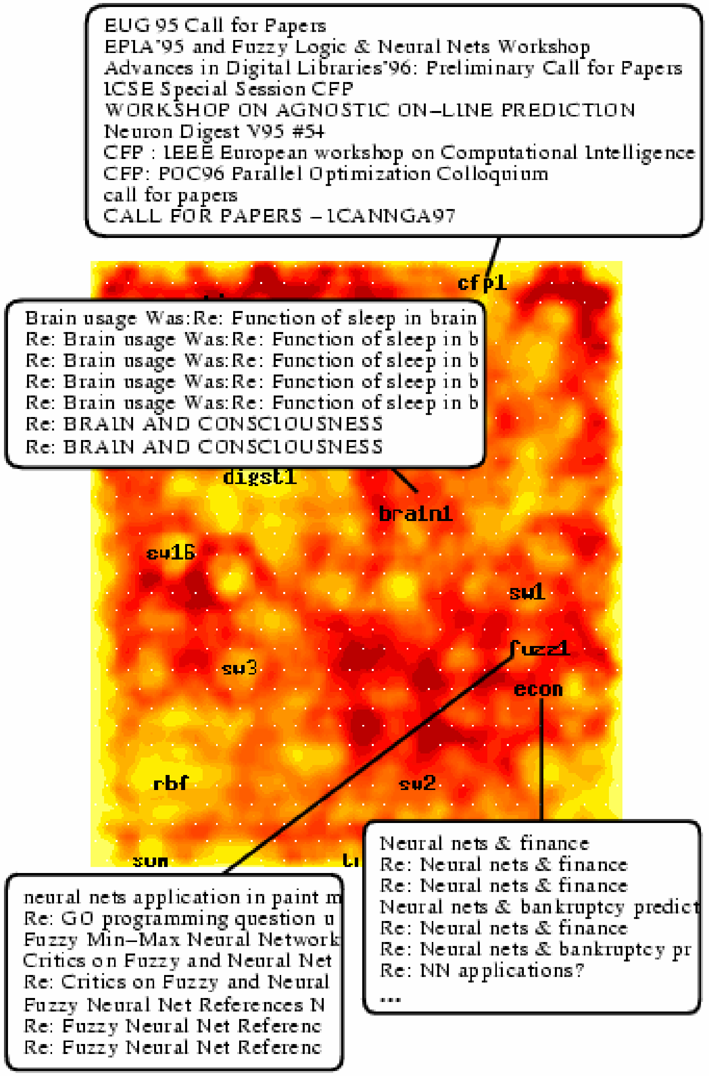
Word categories



Each inset shows the words that have been mapped into one word category

A map of documents

- A set of documents related with neural networks is mapped by using WEBSOM method.
- Browsing for the interface, it is possible to see the "labels" for documents



EUG 95 Call for Papers
EPIA'95 and Fuzzy Logic & Neural Nets Workshop
Advances in Digital Libraries'96: Preliminary Call for Papers
ICSE Special Session CFP
WORKSHOP ON AGNOSTIC ON-LINE PREDICTION
Neuron Digest V95 #54
CFP : IEEE European workshop on Computational Intelligence
CFP: POC96 Parallel Optimization Colloquium
call for papers
CALL FOR PAPERS - ICANNGA97

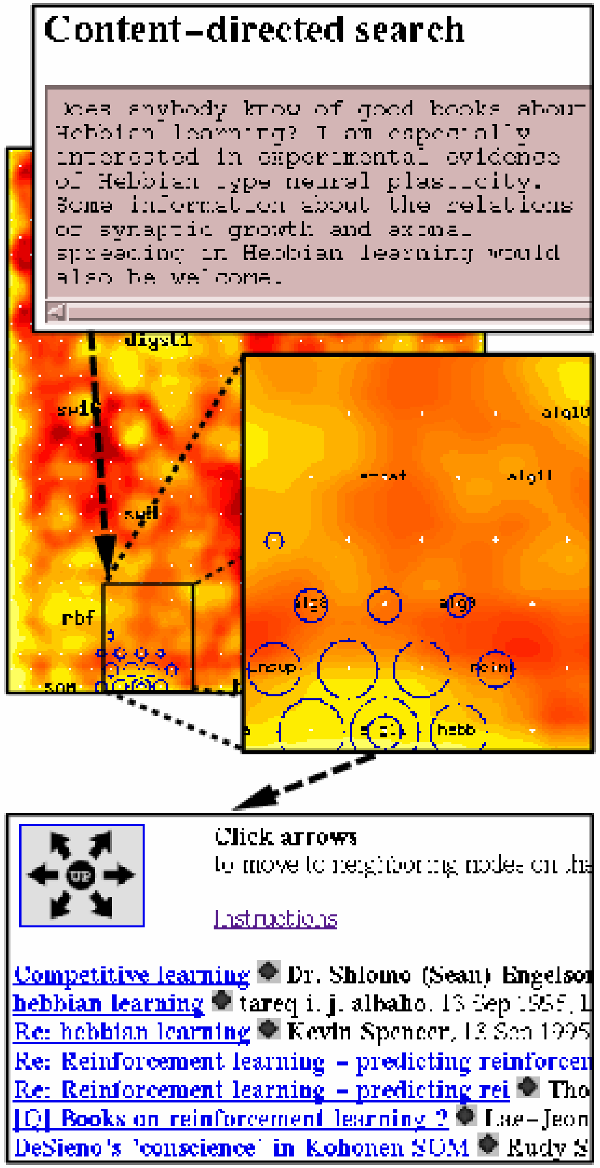
Brain usage Was:Re: Function of sleep in brain
Re: Brain usage Was:Re: Function of sleep in b
Re: Brain usage Was:Re: Function of sleep in b
Re: Brain usage Was:Re: Function of sleep in b
Re: Brain usage Was:Re: Function of sleep in b
Re: BRAIN AND CONSCIOUSNESS
Re: BRAIN AND CONSCIOUSNESS

neural nets application in paint m
Re: GO programming question u
Fuzzy Min-Max Neural Network
Critics on Fuzzy and Neural Net
Re: Critics on Fuzzy and Neural
Fuzzy Neural Net References N
Re: Fuzzy Neural Net Referenc
Re: Fuzzy Neural Net Referenc

Neural nets & finance
Re: Neural nets & finance
Re: Neural nets & finance
Neural nets & bankruptcy predict
Re: Neural nets & finance
Re: Neural nets & bankruptcy pr
Re: NN applications?
...

Sample search

- A new document or any document’s description can be used for finding related documents.
- The circle on the map denote the location of the most representative document for the question.



Content-directed search

Does anybody know of good books about
Hebbian learning? I am especially
interested in experimental evidence
of Hebbian type neural plasticity.
Some information about the relations
of synaptic growth and axonal
spreading in Hebbian learning would
also be welcome.

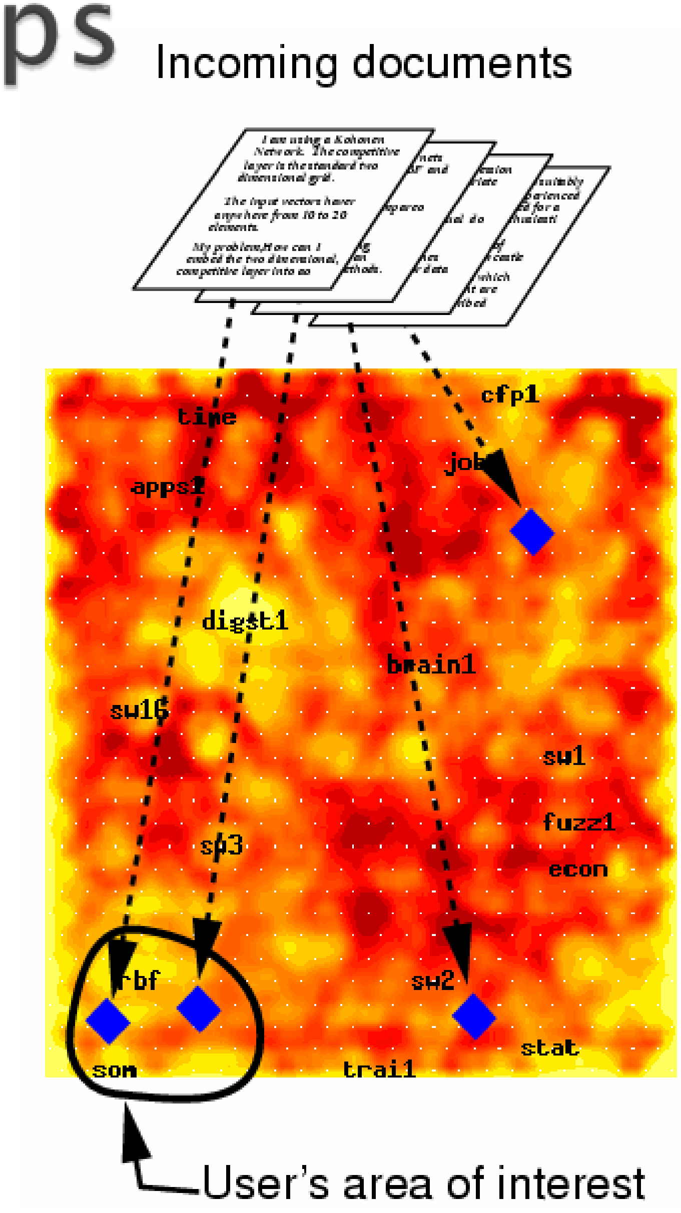
Click arrows
to move to neighboring nodes on the

Instructions

Competitive learning ● Dr. Shlomo (Sean) Engelson
hebbian learning ● tareq i. j. albaho. 13 Sep 1995, 1
Re: hebbian learning ● Kevin Spencer, 13 Sep 1995
Re: Reinforcement learning - predicting reinforcem
Re: Reinforcement learning - predicting rei ● Tho
[Q] Books on reinforcement learning ? ● Lae-Jeon
DeSieno's 'conscience' in Kohonen SOM ● Rudy S

How to use the Maps

- As filter that notifies the user of interesting documents.
- As a searching engine.
- A new index method.


Incoming documents
I am using a Kohonen Network. The competitive layer is the standard two dimensional grid.
The input vectors have anywhere from 10 to 20 elements.
My problem,How can I embed the two dimensional, competitive layer into an
cfp1
time
job
apps1
digst1
brain1
sw16
sw1
fuzz1
sp3
econ
rbf
sw2
stat
som
trai1
User's area of interest

Automatic Web Page Summarization

Automatic web page text summarization

- The goal is to construct automatically summaries of a natural-language document [Hahn00] .
- In many case the web pages only contain few words and the page could contain non-textual elements (e.g. video, pictures, audio, etc.) [Amitay00] .
- In text summarization research, there are three major approaches [Mani99] :
  - paragraph based.
  - sentence based.
  - Using natural language cues in the text.

Types of summaries

- Purpose
  - Indicative, informative, and critical summaries
- Form
  - Extracts (representative paragraphs/sentences/phrases)
  - Abstracts: “a concise summary of the central subject matter of a document” [Paice90].
- Dimensions
  - Single-document vs.. multi-document
- Context
  - Query-specific vs.. query-independent

Genres

- headlines
- outlines
- minutes
- biographies
- abridgments
- sound bites
- movie summaries
- chronologies, etc.

[Mani and Maybury 1999]

What does summarization involve?

- Three stages (typically)
  - content identification
  - conceptual organization
  - realization

Kupiec et al. 95

- Uses Bayesian classifier:

$$P(s \in S \mid F_1, F_2, \dots F_k) = \frac{P(F_1, F_2, \dots F_k \mid s \in S) P(s \in S)}{P(F_1, F_2, \dots F_k)}$$

  - Assuming statistical independence:

$$P(s \in S \mid F_1, F_2, \dots F_k) = \frac{\prod_{j=1}^{k} P(F_j \mid s \in S) P(s \in S)}{\prod_{j=1}^{k} P(F_j)}$$

Kupiec et al. 95

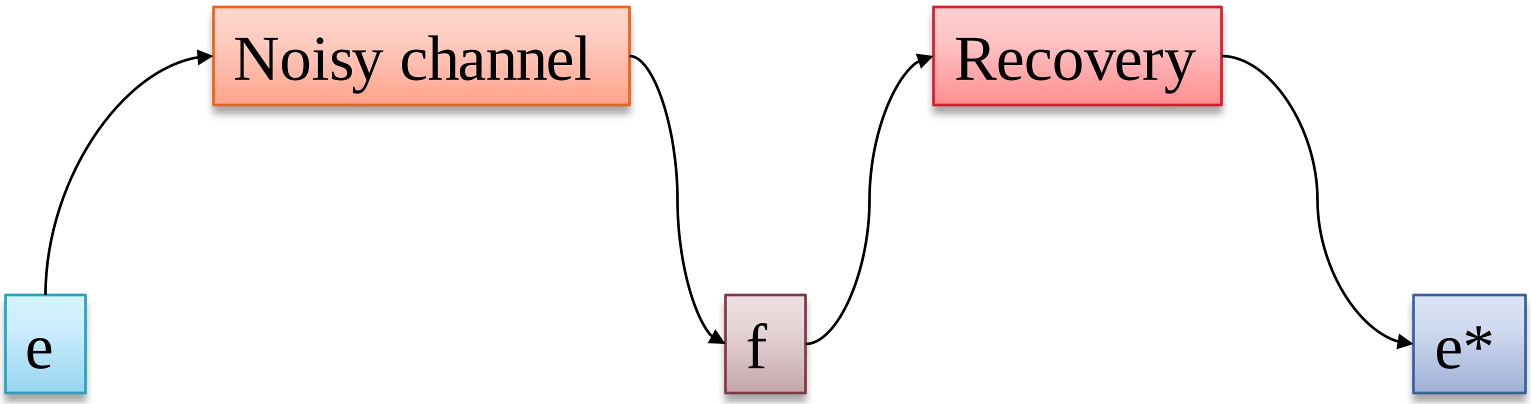
- Performance:
  - For 25% summaries, 84% precision
  - For smaller summaries, 74% improvement over Lead

Osborne 02

- Maxent (loglinear) model – no independence assumptions
- Features: word pairs, sentence length, sentence position, discourse features (e.g., whether sentence follows the “Introduction”, etc.)
- Maxent outperforms Naïve Bayes

Language modeling

- Source/target language
- Coding process



Language modeling

- Source/target language
- Coding process

$$\boxed{e^* = \underset{e}{argmax}\ p(e|f) = \underset{e}{argmax}\ p(e)\ .\ p(f|e)}$$

$$p(E) = p(e_1).p(e_2|e_1).p(e_3|e_1e_2)\ldots p(e_n|e_1\ldots e_{n-1})$$

$$p(E) = p(e_1).p(e_2|e_1).p(e_3|e_2)\ldots p(e_n|e_{n-1})$$

Summarization using LM

- Source language:
  - full document
- Target language:
  - summary

Berger & Mittal 00

- Gisting (OCELOT)

$$\boxed{g^* = \underset{g}{argmax}\ p(g|d) = \underset{g}{argmax}\ p(g)\ .\ p(d|g)}$$

- content selection (preserve frequencies)
- word ordering (single words, consecutive positions)
- search: readability & fidelity

Berger & Mittal 00

- Limit on top 65K words
- word relatedness = alignment
- Training on 100K summary+document pairs
- Testing on 1046 pairs
- Use Viterbi–type search
- Evaluation: word overlap (0.2–0.4)
- transilingual gisting is possible
- No word ordering

Berger & Mittal 00

Sample output:

Audubon society atlanta area savannah georgia chatham and local birding savannah keepers chapter of the audubon georgia and leasing

Extraction of key-text component from web pages

Extraction of key-text components from web pages

- The key-text components are parts of an entire document
- **Key-text:** *A paragraph, phrase and inclusive a word*, that contain **significant information** about a particular topic, from the *web site user point of view*.
- A **web site keyword** is *"a word or possibly a set of words that make a web page more attractive for an eventual user during his visit to the web site'"* [Velasquez05b].

Extraction of key-text components from web pages

- The assumption is that there exists a ***correlation*** *between the* ***time that the user spent in a page*** *and his/her* ***interest*** *in its content* [Velasquez04b] .
- Usually, the **keywords** in a web site have been related with the “**most frequently used words”.**
- In [Buyukkokten01] a method to extract keywords from a huge set of web pages is introduced.

Summary

- The **vector space model** is a recurrent *method to represent a document as a feature vector*.
- Because the set of words used in the construction of the web site could be a lot, it is necessary to apply a **stop word cleaning** and **stemming process**.
- A web page content is different to a common document, in fact, a web page contain **semi-structured text**, i.e., with tags that give additional information about the text component.
- Also a page could contain *pictures, sounds, movies, etc.*
- Sometime the page text content is a **short text** or inclusive a **set of unconnected words**.