



MEDNARODNA
PODIPLOMSKA ŠOLA
JOŽEFA STEFANA

JOŽEF STEFAN
INTERNATIONAL
POSTGRADUATE SCHOOL

Data Mining and Knowledge Discovery Part III - Text, web and multimedia mining

Prof. Dr. Dunja Mladenić

Information and Communication
Technologies (ICT2), 2011/2012

www.risi.si

Overview

- Introduction
- Supervised Learning
- Semi-supervised Learning
- Unsupervised Learning
- References

©Dunja Mladenic

Text, web and multimedia mining

- "...finding **interesting** regularities in large **text, web or multimedia data...**" (Usama Fayad, adapted)
 - ...where **interesting** means: non-trivial, hidden, previously unknown and potentially useful
- finding regularities in web structure, web logs, web content
 - analysis and profiling of web customers based on web-server log files

Knowledge Discovery Process - phases

Following CRISP-DM methodology:

1. Business understanding
2. Data understanding
3. Data preparation
4. Modeling
5. Evaluation
6. Deployment

First steps to data modeling


- Data representation in a suitable format
 - feature vectors are commonly used
 - for each data point (example), each feature has one value from a predefined set of possible values
 - features generation and feature selection may be applied

transformation or combination

feature subset selection

Illustrative example – cartoon descriptions

Bob the builder



Vehicles characters = yes
Human characters = yes

Features:

- vehicle characters [yes, no]
- human characters [yes, no]

Feature vector = [1, 1]

Basic approaches to modeling using machine learning methods

When to apply different approaches?

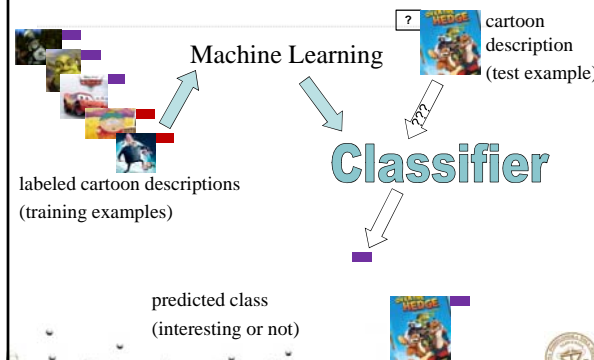
- **Supervised learning** (classification)
 - ...given cartoon descriptions and corresponding labels of interestingness for children, the goal is to find rules which can map/predict interestingness of a new cartoon based on its description
- **Semi-supervised learning** (transduction, active learning)
 - ... given cartoon descriptions and corresponding labels interestingness for children **for only a few cartoons**, leverage these to find the most probable interestingness label for arbitrary cartoons
- **Unsupervised learning** (clustering, decompositions)
 - ...given only cartoon descriptions, find groups of similar cartoons

Supervised learning

Assign an object to a given finite set of classes:

- Medical diagnosis
 - ...assign diagnosis to a patient
- Credit card applications or transactions
 - ...assign credit score to an applicant
- Fraud detection in e-commerce
 - ...decide about fraud or non-fraud event in a business process
- Financial investments
 - ...decide whether to *buy* or *sell* or *hold* on a stock exchange
- Spam filtering of e-mails
 - ...decide if an email is a spam or a regular email
- Recommending articles in a newspaper
 - ...decide if an article fits the user profile
- Semantic/linguistic annotation
 - ...assign semantic or linguistic annotation to a word or phrase

Recommending cartoons



Supervised learning

Given: a set of labeled examples represented by feature vectors
Goal: build a model approximating the target function which would automatically assign right label to a new unlabeled example

- Feature values:
 - discrete (eg., eyes_color ∈ {brown, blue, green})
 - continuous (eg., age ∈ [0..200])
 - ordered (eg., size ∈ {small, medium, large})
- Values of the target function – labels:
 - discrete (classification) or continuous (regression)
 - exclude each other (eg., medical diagnosis) or not (eg., a single document content can talk about several topics)
 - have some predefined relations (taxonomy of document categories, e.g., DMOZ)

The target function can be

- represented in different ways (storing examples, symbolic, numerical, graphical,...)
- modeled by using different algorithms

Short? **strative example**
 recommending cartoon for children

Illustrative example
 not interesting for children

Illustrative example

Recommending cartoon for children

Title	Characteristic words	Duration
Bob the builder	vehicles, human, Bob,..	10 mins
Pixar-Locomotion	vehicles, locomotive,..	5 mins
Ice age	animals, squirrel, ice,..	90 mins
Over the hedge	animals, neighborhood,..	60 mins
Cars	vehicles, car, race,..	90 mins

Target function

There is a trade-off between the expressiveness of a representation and the ease of learning

- The more expressive a representation, the better it will be at approximating an arbitrary function; however, more examples will be needed to learn an accurate function

Illustrative example

- Values of the target function: discrete labels (classification), exclude each other

Cartoon interesting for children: yes no

Possible data visualization

Possible Model for not interesting
(vehicles = no) and (human = yes)

Generalization

- Model must generalize the data to correctly classify yet unseen examples (the ones which don't appear in the training data)
- Lookup table of training examples is a consistent model that does not generalize
 - An example that was not in the training data can not be classified

Occam's razor:

- Finding a *simple* model helps ensure generalization

Algorithms for learning classification models

- Storing examples
 - Nearest Neighbour
- Symbolic
 - Decision trees
 - Rules in propositional logic or first order logic
- Numerical
 - Perceptron algorithm
 - Winnow algorithm
 - Support Vector Machines
 - Logistic Regression
- Probabilistic graphical models
 - Naive Bayesian classifier
 - Hidden-Markov Models

Nearest neighbor

- Storing training examples without generating any generalization
 - Simple, requires efficient storage
- Classification by comparing the example to the stored training examples and estimating the class based on classes of the most similar examples
 - Similarity function is crucial

Also known as:

- Instance-based, Case-based, Exemplar-based, Memory-based, Lazy Learning

Similarity/Distance

- For continuous features use Euclidian distance

$$Dist(e_1, e_2) = \sqrt{\sum_{i=1}^n (f_{1i} - f_{2i})^2}$$

$$e_k = \langle f_{k,1}, f_{k,2}, \dots, f_{k,n} \rangle$$

- For discrete features, assume distance between two values is 0 if they are the same and 1 if they are different (eg., Hamming distance for bit vectors).

To compensate for difference in units across features, scale all continuous values to the interval [0,1].

Nearest neighbor

K = 2
K = 5

Decision tree model

Linear Model

Support Vector Machine

- Learns a hyperplane in higher dimensional space
 - that separates the training data and
 - gives the highest margin
- Implicit mapping of the original feature space into higher dimensional space
 - mapping using so called kernel function (eg., linear, polynomial, ...)

Regarded as state-of-the-art in text document classification

SVM Demo

Naïve Bayes

Determine class of example e_k by estimating

$$P(c_i | e_k) = \frac{P(c_i)P(e_k | c_i)}{P(e_k)} = \arg \max_i P(c_i)P(e_k | c_i)$$

- $P(c_i)$ – estimate from the data using frequency: no. of examples with class c_k / no. of all examples
- $P(e_k | c_i)$ – too many possibilities (all combinations of feature values)
 - assume feature independence given the class

$$P(e_k | c_i) = \prod_{j=1} P(f_{kj} | c_i)$$

Naïve Bayes on text

$$P(C | Doc) = \frac{P(C) \prod_{W \in Doc} P(W | C)^{Freq(W, Doc)}}{\sum_{C'} P(C') \prod_{W_i \in Doc} P(W_i | C')^{Freq(W_i, Doc)}}$$

- Document is represented as a set of words W
- For binary classification, each classifier has two distributions: P(W|pos), P(W|neg)
- When having a large collection of binary classifiers (one per category) with unbalanced prior probability, consider only promising categories:
 - calculated P(pos|Doc) is high meaning that the classifier has P(W|pos) > 0 for at least some W from the document (otherwise, the prior probability is returned, P(neg) is about 0.90)

Example of Naïve Bayes classifier

	A	B	C	D	E
w1	1	1	1	0	0
w2	0	0	0	0	1
w3	1	0	1	0	0
w4	0	0	0	1	1
w5	1	1	0	0	0

- Estimate model parameters from data.
 - P(pos) = 2/4 = 0.5; P(neg) = 2/4 = 0.5
 - P(w1|pos) = 2/2 = 1; P(w1|neg) = 0/2 = 0
 - P(w2|pos) = 0/2 = 0; P(w2|neg) = 1/2 = 0.5
 - P(w3|pos) = 1/2 = 0.5; P(w3|neg) = 0/2 = 0
 - P(w4|pos) = 0/2 = 0; P(w4|neg) = 2/2 = 1
 - P(w5|pos) = 1/2 = 0.5; P(w5|neg) = 0/2 = 0
- Calculate probability for each class using the model on A.
 - P(pos|A) = P(pos) * [P(w1|pos)*P(w3|pos) * P(w5|pos)] / sum_c = 0.5 * [1 * 0.5 * 0.5] / 0.125 = 0.125 / 0.125 = 1
 - P(neg|A) = P(neg) * [P(w1|neg)*P(w3|neg) * P(w5|neg)] / sum_c = 0.5 * [0 * 0 * 0] / 0.125 = 0 / 0.125 = 0
- Classify A returning the most probable class **pos**

Generative Probabilistic Models

- Assume a simple (usually unrealistic) probabilistic method by which the data was generated
- Each class value has a different parameterized generative model that characterizes it
- Training:** Use the data for each category to estimate the parameters of the generative model for that category.
 - Maximum Likelihood Estimation (MLE):** Set parameters to maximize the probability that the model produced the given training data
 - If M_k denotes a model with parameter values λ and D_k is the training data for the k th class, find model parameters for class k (λ_k) that maximize the likelihood of D_k :

$$\lambda_k = \operatorname{argmax}_{\lambda} P(D_k | M_{\lambda})$$
- Testing:** Use Bayesian analysis to determine the category model that most likely generated a specific test instance.

Semi-supervised learning

Similar to supervised learning except that

- we have examples and only some of them are labeled
- we may have a human available for a limited time to provide labels of examples
 - ...this corresponds to the situation where all the cartoons in our collection have descriptions, but only a few have label
 - ...and occasionally we have a human for a limited time to respond the questions about the cartoons



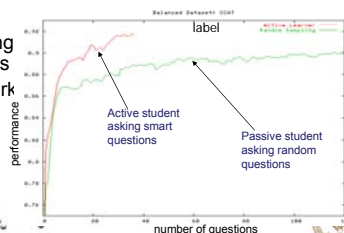
MEDNARODNA
PODIPLOMSKA ŠOLA
JOŽEFA STEFANA

JOŽEF STEFAN
INTERNATIONAL
POSTGRADUATE SCHOOL

Active Learning

Active Learning

- We use this methods whenever hand-labeled data are rare or expensive to obtain
- Interactive method
 - Teacher → Data & labels → passive student
 - Teacher ← query → active student
- Requests only labeling of "interesting" objects
- Much less human work needed for the same result compared to arbitrary labeling examples



Unsupervised learning

- Given is a set of examples
- The goal is: to cluster the examples into several groups based on some similarity measure
 - examples inside the group should be similar while examples between the groups should be different

Similarity measure plays a crucial role:

Cosine $Cos(d_1, d_2) = \frac{d_1 \cdot d_2}{\|d_1\| \|d_2\|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$ Jaccard $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$

Minkowsky (norm k) $L_k(d_1, d_2) = \|d_1 - d_2\|_k = \left(\sum_{i=1}^n (w_{1i} - w_{2i})^k \right)^{1/k}$

Manhattan (k=1), Euclidean (k=2)

Clustering methods

- Hierarchical**
 - agglomerative – at each step merge two or more groups
 - divisive – at each step break the selected group into two or more groups
- Non hierarchical**
 - requires specification of the number of clusters
 - optimization of the initial clustering (e.g., maximize similarity of examples inside the same group)
- Geometrical**
 - map multidimensional space into two- or three-dimensional (e.g., principal component analysis)
- Graph-theoretical**

K-Means clustering algorithm

- Given:**
 - set of examples (e.g., TFIDF vectors of documents),
 - distance measure (e.g., cosine)
 - K** (number of groups)
- For each of K** groups initialize its centroid with a random document
- While** not converging
 - Each document is assigned to the nearest group (represented by its centroid)
 - For each group calculate new centroid (group mass point, average document in the group)

Example of k-means clustering

	A	B	C	D	E
w1	1	1	1	0	0
w2	0	0	0	0	1
w3	1	0	1	0	0
w4	0	0	0	1	1
w5	1	1	0	0	0

- Randomly select two examples, e.g., A, D to be representatives of two clusters I: A, II: D
- Calculate similarity of other examples to the them
B, I= 0.82, B, II= 0, C, I= 0.82, C, II= 0, E, I= 0, E, II= 0.7
- Assign examples to the most similar cluster
I: (A,B,C) II: (D,E)
- Calculate the cluster centroid
I: 1,0,0.67,0,0.67 II: 0,0.5,0,1,0
- Calculate similarity of all the examples to the centroids A, I= 0.88, A, II= 0, B, I= 0.77, B, II= 0, C, I= 0.77, C, II= 0, D, I= 0, D, II= 0.82, E, I= 0, E, II= 0.87
- Assign examples to the most similar cluster
I: (A,B,C) II: (D,E)
- Repeat steps 3-5 until the clustering got stabilized

K=2

Latent Semantic Indexing

- LSI is a statistical technique that attempts to estimate the hidden content structure within documents:
 - ...it uses linear algebra technique Singular-Value-Decomposition (SVD)
 - ...it discovers statistically most significant co-occurrences of terms

LSI Example

	d1	d2	d3	d4	d5	d6
cosmonaut	1	0	1	0	0	0
astronaut	0	1	0	0	0	0
moon	1	1	0	0	0	0
car	1	0	0	1	1	0
truck	0	0	0	1	0	1

Original document-term matrix

	d1	d2	d3	d4	d5	d6
Dim1	-1.62	-0.60	-0.04	-0.97	-0.71	-0.26
Dim2	-0.46	-0.84	-0.30	1.00	0.35	0.65

Rescaled document matrix, Reduced into two dimensions

	d1	d2	d3	d4	d5	d6
d1	1.00					
d2	0.8	1.00				
d3	0.4	0.9	1.00			
d4	0.5	-0.2	-0.6	1.00		
d5	0.7	0.2	-0.3	0.9	1.00	
d6	0.1	-0.5	-0.9	0.9	0.7	1.00

Correlation matrix

High correlation although d2 and d3 don't share any word



MEDNARODNA
PODIPLOMSKA ŠOLA
JOŽEFA STEFANA

JOŽEF STEFAN
INTERNATIONAL
POSTGRADUATE SCHOOL

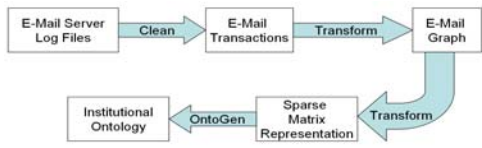
Social networks

Social networks

- Social networks can be also potential source of data for machine learning and building semantic structures
 - ...conceptually they share similar underlying structure as text – namely, the underlying distribution is generated by power-law
- In the next slides we show how social networks can be modeled using unsupervised techniques

Analysis of e-mail graph

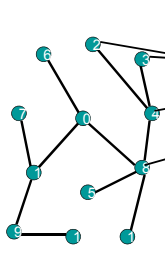
- An e-mail graph can be analyzed in the following 5 major steps:
 1. Starting with log files from an e-mail server where the data include information about e-mail transactions with the fields: **sender** and the **list of receivers**.
 2. After cleaning we get the data in the form of **e-mail transactions** which include e-mail addresses of **sender** and **receiver**.
 3. From a set of **e-mail transactions** we construct a **graph** where vertices are e-mail addresses connected if there is a transaction between them
 4. **E-mail graph** is transformed into a **sparse matrix** allowing to perform data manipulation and analysis operations
 5. **Sparse matrix** representation of the graph is analyzed with **ontology learning** tools producing an **ontological structure** corresponding to the **organizational structure** of the institution where e-mails came from.



Graph transformation into a set of sparse matrix

- Graph with N vertices is transformed into $N \times N$ sparse matrix where:
 - ... X th row represents information for X th vertex
 - ... X th row has nonzero components for:
 - X th vertex itself and
 - X th vertex's neighbors on the distance D (e.g. 1, 2, 3)
 - Intuitively, X th row represents numerically "neighborhood" of the X th vertex within the graph:
 - X th element in the X th row has weight 1
 - ...elements representing neighbors have lower weights relative to the distance (d) from the X th vertex ($1/(2^d)$)
 - (e.g. 1, 0.5, 0.25, 0.125, ...)

Graph transformation into sparse matrix (example)



	0	1	2	3	4	5	6	7	8	9	10	11
0	1	0.5			0.25	0.25	0.5	0.25	0.5	0.25		0.25
1	0.5	1			0.25	0.5	0.25	0.5	0.25	0.5	0.25	
2			1	0.25	0.5				0.25			
3			0.25	1	0.5				0.25			
4	0.25		0.5	0.5	1	0.25			0.5			0.25
5	0.25			0.25	1				0.5			0.25
6	0.5	0.25					1		0.25			
7	0.25	0.5						1		0.25		
8	0.5	0.25	0.25	0.25	0.5	0.5	0.25		1			0.5
9	0.25	0.5						0.25		1	0.5	
10		0.25									0.5	1
11	0.25				0.25	0.25			0.5			1

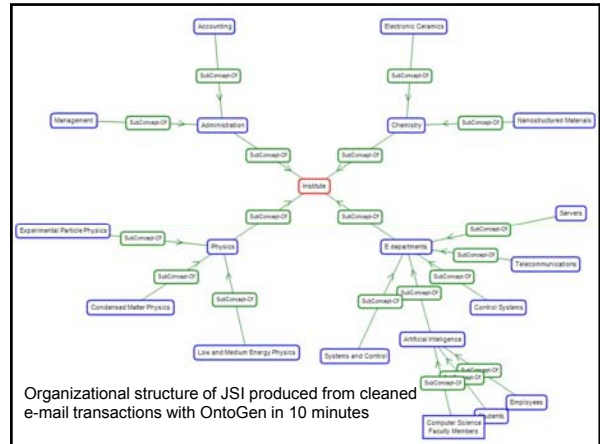
Transforming Graph into Matrix

Data used for experimentation

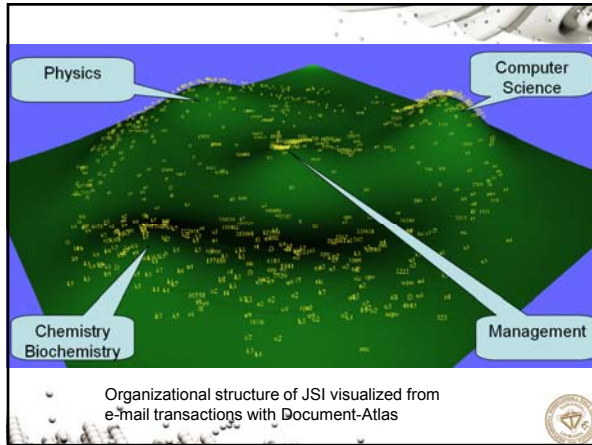
- The data is the collection of log files with e-mail transactions from local e-mail spam filter software Amavis (<http://www.amavis.org/>):
 - Each line of the log files denotes one event at the spam filter software
 - We were interested in the events on successful e-mail transactions
 - ...having information on **time**, **sender**, and **list of receivers**
 - An example of successful e-mail transaction is the following line:
 - 2005 Mar 28 13:59:05 patsy amavis[33972]: (33972-01-3) Passed CLEAN, [217.32.164.151] (193.113.30.29) <john.nj.davies@bt.com> -> smarko.grobelnik@ijs.si> Message-ID: <21DA6754A9238B48B92F39637EF307FD0D4781C8@i2km41-lkdy.domain1.sytemhost.net>, Hits: -1.668, 6389 ms

Some statistics about the data

- The log files include e-mails for 19 months:
 - ...this sums up to **12.8Gb** of data.
 - After filtering out successful e-mail transactions it remains **564Mb**
 - ...which contains approx. **2.7 million** of successful e-mail transactions used for further processing
 - The whole dataset contains references to approx. **45000** e-mail addresses
 - ...after the data cleaning phase the number is reduced to approx. **17000** e-mail addresses
 - ...out of which **770** e-mail addresses are internal from the home institution (with local domain name)



Organizational structure of JSI produced from cleaned e-mail transactions with OntoGen in 10 minutes



Organizational structure of JSI visualized from e-mail transactions with Document-Atlas



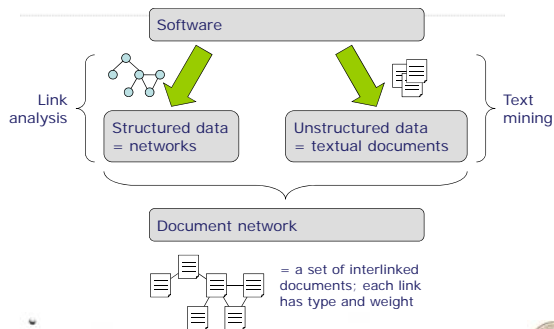
MEDNARODNA
PODIPLOMSKA ŠOLA
JOŽEFA STEFANA

JOŽEF STEFAN
INTERNATIONAL
POSTGRADUATE SCHOOL

Cross-modal analysis

www.mss.si

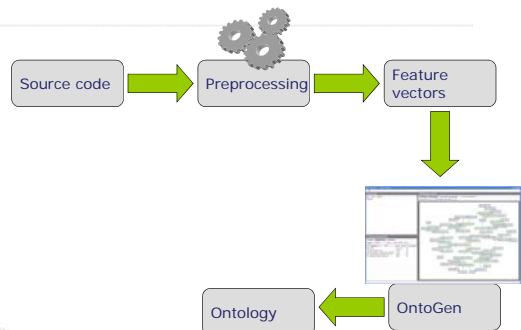
Software Mining



[Grčar, Mladenić, Grobelnik, 2007]



Structuring extracted knowledge





MEDNARODNA
PODIPLOMSKA ŠOLA
JOŽEFA STEFANA

JOŽEF STEFAN
INTERNATIONAL
POSTGRADUATE SCHOOL

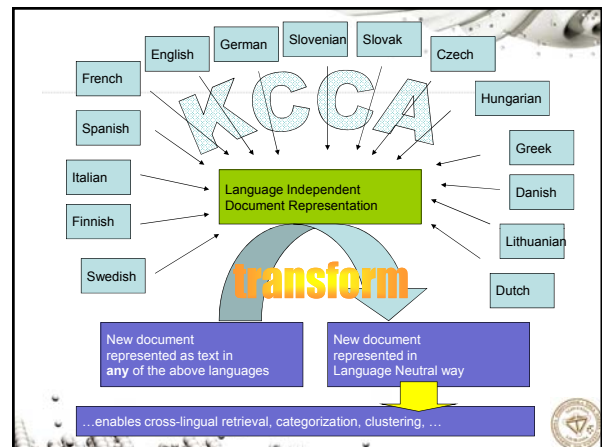
Multilingual data

Multilingual data

- Text in several natural languages
- Perform machine learning and retrieval on textual data regardless the language differences
- Approach:
 - Machine Translation (on sentence level)
 - Multilingual lexicon (on word level)
 - Mapping into semantic space (on word level, eg., KCCA)

KCCA to handle multilingual data

- KCCA enables representing documents in a "language neutral way"
- Intuition behind KCCA:
 1. Given a parallel corpus (such as Acquis)...
 2. ...first, we automatically identify language independent semantic concepts from text,
 3. ...then, we re-represent documents with the identified concepts,
 4. ...finally, we are able to perform cross language statistical operations (such as retrieval, classification, clustering...)



Input for KCCA

- On input we have set of aligned documents:
 - For each document we have a version in each language
- Documents are represented as bag-of-words vectors

The Output from KCCA

- **The goal:** find pairs of *semantic dimensions* that co-appear in documents and their translations with high correlation
 - *Semantic dimension* is a weighted set of words.
- These pairs are pairs of vectors, one from e.g. English bag-of-words space and one from German bag-of-words space.

The Algorithm – Theory

Formally the KCCA solves:

$$\max_{(x,y)} \text{Corr}(\langle x, \text{[img alt="English flag icon"]} \rangle, \langle y, \text{[img alt="German flag icon"]} \rangle)$$

- x, y – semantic directions for English and German
- $(\text{[img alt="English flag icon"]}, \text{[img alt="German flag icon"]})$ is a pair of aligned documents

Examples of Semantic Dimensions from Acquis corpus: English-French (1/2)

Most important words from semantic dimensions automatically generated from 2000 documents:

Veterinary, Transport

DIRECTIVE, DECISION, VEHICLES, AGREEMENT, EC, VETERINARY, PRODUCTION, HEALTH, MEAT

DIRECTIVE, DECISION, VEHICULES, PRESENTE, RESIDUS, ACCORD, PRODUITS, ANIMAL

NOMENCLATURE, COMBINED, COLUMN, GOODS, TARIFF, CLASSIFICATION, CUSTOMS

NOMENCLATURE, COMBINEE, COLONNE, MARCHANDISES, CLASSEMENT, TARIF, TARIFAIRES

EMBRYOS, ANIMALS, OVA, SEMEN, ANIMAL, CONVENTION, BOVINE, DECISION, FEEDINGSTUFFS

EMBRYONS, ANIMAUX, OVULES, CONVENTION, SPERME, EQUIDES, DECISION, BOVINE, ADDITIFS

SUGAR, CONVENTION, ADDITIVES, PIGMEAT, PRICE, FEEDINGSTUFFS, SEED

SUCRE, CONVENTION, PORC, ADDITIF, PRIX, ALIMENTATION, SEMENCES, DECISION

EXPORT, LICENCES, LICENCE, REFUND, VEHICLES, FISHERY, CONVENTION, CERTIFICATE, ISSUED

EXPOSITION, CERTIFICATS, CERTIFICATE, RECHER, VEHICULES, T, CONVENTION

Export Licences Agriculture Veterinary

Examples of Semantic Dimensions from Acquis corpora: English-Slovene (2/2)

Most important words from semantic dimensions automatically generated from 2000 documents :

Agriculture

OLIVE, OIL, AID, SUGAR, PRICE, STATE, MILK, LICENCES, OR, EXPORT, INTERVENTION

OLJA, OLJNEGA, POMOCI, SLADKORJA, POMOC, OLJK, SLADKOR, ALI, DOVOLJENJA

NOMENCLATURE, COLUMN, COMBINED, GOODS, TARIFF, CLASSIFICATION, STATEMENT, QUOTA

NOMENKLATURO, STOLPCU, NOMENKLATURE, KOMBINIRANO, KOMBINIRANE, CARINSKI, BLAGA

QUOTAS, TARIFF, SEED, CUSTOMS, COLUMN, ENERGY, INVOKED, ATOMIC, QUOTA, OPENING

KVOT, TARIFNE, SEMENA, KVOTE, TARIFNIH, CARINSKI, ATOMSKO, ENERGIJO, ODPRTJU

DESIGNATIONS, GEOGRAPHICAL, INDICATIONS, EURATOM, PROTECTED, ECSC, NAMES, ORIGIN

OZNACB, EURATOM, GEOGRAFSKI, POREKLA, ESPJ, ZASCI, ENIH, OZNACBE, IMEN, REGISTER

WINE, WINES, ALCOHOL, DRINKS, DISTILLATION, POULTRYMEAT, ICEWINE, ANALYSIS

VINO, VINA, VIN, VINSKEM, VINSKI, ALKOHOL, NAMIZNEGA, DESTILACIJO, DESTILACIJE

Wine Agriculture protection Energy

Applications of KCCA

- **Cross-lingual document retrieval:** retrieved documents depend only on the meaning of the query and not its language.
- **Automatic document categorization:** only one classifier is learned and not a separate classifier for each language
- **Document clustering:** documents should be grouped into clusters based on their content, not on the language they are written in.
- **Cross-media information retrieval:** in the same way we correlate two languages we can correlate text to images, text to video, text to sound, ...

Example of cross-lingual information retrieval on Reuters news corpus using KCCA

stock exchange

ns

de

[Bib Search]

Document Name

0.311658

REPUBLIC OF IRELAND: Country's Irish Stock Exchange listing cancelled.

0.311615

REPUBLIC OF IRELAND: Country's Irish Stock Exchange listing cancelled.

0.241181

MACEDONIA: ONE COMPANY TRADING: Stock transfer and trade.

0.240322

BORSE ANDERT PLANE FUR OPTIONSSCHEINHANDEL.

ILLUSTRATIVE EXAMPLES ON TEXT MINING

©Dunja Mladenic

Representing textual data

Having a set of documents, represent each as a feature vector:

- divide text into units (eg., words), remove punctuation, (remove stop-words, stemming,...)
- each unit becomes a feature having numeric weight as its value (eg., number of occurrences in the text - referred to as term frequency or TF)

Commonly used weight is TFIDF: $TFIDF(w) = tf(w) * \log\left(\frac{N}{df(w)}\right)$

- $tf(w)$ – term frequency (no. of occurrences of word w in document dokumentu)
- $df(w)$ – document frequency (no. of documents containing word w)
- N – no. of all documents

Textual data - example

Bob the builder is an animated movie having a tractor as one of the main characters. The stories are on Bob and his friends facing challenges and jointly solving them, such as, repair a roof or save Bob's cat from a tall tree...

Pixar has several short animated movie products interesting for children. Locomotion is one of them showing train engine and a train wagon as two characters that face a challenge of crossing.

Simpsons family movie include two children facing challenges of daily life in a family. Sometimes sarcastic ...

	bob	builder	children	animated	movie	character	friend	vehicle
Bob	1	1	0	1	1	1	1	1
Pixar	0	0	1	1	1	1	0	0
Simpsons	0	0	1	0	1	0	0	0

Representation calculation

A Bob the builder is an animated movie having a tractor as one of the main characters.

B Pixar has several short animated movie products for children.

C Simpsons' family movie include two children facing challenges of daily life in a family.

TFIDF			log...	DF	TF	Feature				
A	B	C		A	B	C				
0.18	0.18	0	0.18	2	1	1	0	animated		
0.48	0	0	0.48	1	1	0	0	bob		
0.48	0	0	0.48	1	1	0	0	builder		
0	0	0.48	0.48	1	0	0	1	challenges		
0.48	0	0	0.48	1	1	0	0	characters		
0	0.18	0.18	0.18	2	0	1	1	children		
0	0	0.48	0.48	1	0	0	1	facing		
0	0	0.95	0.48	1	0	0	2	family		
0	0	0.48	0.48	1	0	0	1	include		
0	0	0.48	0.48	1	0	0	1	life		
0.48	0	0	0.48	1	1	0	0	main		
0	0	0	0	3	1	1	1	movie		
0	0.48	0	0.48	1	0	1	0	pixar		
0	0.48	0	0.48	1	0	1	0	products		
0	0.48	0	0.48	1	0	1	0	simpson's		
0	0.48	0	0.48	1	0	1	0	short		
0.48	0	0	0.48	1	1	0	0	tractor		

Classification calculation using k-Nearest neighbor

$$\text{Cos}(d_1, d_2) = \frac{\sum_i x_{1i} x_{2i}}{\sqrt{\sum_i x_{1i}^2} \sqrt{\sum_i x_{2i}^2}}$$

$\text{Cos}(A,B) = (0.18 * 0.18) / (1.0883 * 0.9931) = 0.0324 / 1.0809 = 0.03$

$\text{Cos}(A,C) = 0.0 / (1.0883 * 1.3625) = 0.0 / 1.4828 = 0.0$

$\text{Norm}(A) = \text{sqrt}(0.18^2 + 0.18^2 + 0.48^2 + 0.48^2 + \dots) = 1.0883$
 $\text{Norm}(B) = 0.9931$
 $\text{Norm}(C) = 1.3625$

Predict class(A) = class(B)

Feature
animated
bob
builder
challenges
characters
children
facing
family
include
life
main
movie
pixar
products
simpson's
short
tractor



MEDNARODNA
PODIPLOMSKA ŠOLA
JOŽEFA STEFANA

JOŽEF STEFAN
INTERNATIONAL
POSTGRADUATE SCHOOL

References

© Dunja Mladenić

Reading seminar assignment

- M. Grobelnik, D. Mladenić. Automated knowledge discovery in advanced knowledge management. Journal of Knowledge management 9:5, 132-149, 2005.
- Tom M. Mitchell . Mining Our Reality, Science:326, December 2009.
- M. Grobelnik et al., Machine Learning Techniques for Understanding Context and Process, Context and Semantics for Knowledge Management, 127-148, 2011.
- Tom M. Mitchell et al. Populating the Semantic Web by Macro-Reading Internet Text, ISWC-2009.

© Dunja Mladenić

Practical seminar assignment

Blaz.Fortuna@ijs.si teaching assistant

For individual assignment description, data and the code to play with see (ICT3)

- <http://www.blazfortuna.com/Pages/Teaching.aspx>

©Dunja Mladenic

