

# Language Technology: Document Categorization

Walter Daelemans

walter.daelemans@ua.ac.be

## Document Categorization

- Assign to a document one (or small number) of a set of predefined labels
- Can be anything:
  - Easy: which documents contain a word with three e's?
  - Complex:
    - which documents contain an optimistic view on the future of world economy?
    - which documents are funny?

## Applications

- automatic indexing for Boolean IR systems
- web search engines (grouping search results, web directories, ...)
- document organization (knowledge management)
- document filtering / routing (dynamically)
  - e.g. spam filters, email filters, news feeds, narrowcasting
- document annotation (semantic web)
- module in NLP system: context determination, word sense disambiguation
- authorship / gender / personality attribution
- (sub)-language identification

## Classification vs. Retrieval

- Different goals
  - **Classification**: Labeling / filtering documents
  - **Retrieval**: Finding documents given a query
- Different information
  - **Classification**: document is bag of words + class
  - **Retrieval**: document is bag of words
- Information Retrieval with “closed vocabulary” (meta-information)

The screenshot shows a Yahoo! Shopping search results page for 'Apple Laptops'. The page includes a search bar, navigation tabs (Home, Clothing, Electronics, Computers, Home & Garden, Mother's Day, More, My Lists), and a list of search results. The results are filtered by 'Apple' and sorted by 'Top Results'. The first result is 'The Official Apple iPod Store' with a price of \$599. Other results include 'Free Apple Store Coupons', 'HP DV5000Z Entertainment Laptop', and 'Buy Toshiba Laptop from \$599'. The page also features a 'Refine Results for Apple Laptops' section with a price range filter and a 'Compare Side by Side' button.

## Text is a special kind of data

- Direct entry, OCR (.99 accuracy), Speech Recognition output (.50-.90 accuracy), ...
- What we have:
  - Characters, character n-grams, words, word n-grams, lay-out, counts, lengths, ...
  - Document = Bag of Words
- What we want:
  - Meaning
- Bridging the gap:
  - Tagging, lemmatization, phrase chunking, grammatical relations, ... (Language Technology)

## Bags of words

- Meaning of a document is represented by the words occurring in them
- Word order and position in the document are not relevant
- Given a document vector type of length  $n$  (where  $n$  is the number of different words in the *document collection*), a specific document is represented by a binary vector, according to the presence or absence of words
- Instead of 1-0, numeric weights representing word importance can be used

## Classification

- Allows use of supervised machine learning techniques
  - Objective measurement of quality
- Fits current conception of NLP (everything, even parsing, can be seen as a cascade of classification problems)
- Classifiers are interesting plug-in modules in other systems

## Evaluation

### Confusion table

System/Truth	yes	no
yes	a	b
no	d	c

Precision =  $a / a+b$  (P)

Recall =  $a / a+d$  (R)

F-score ( $\beta = 1$ ) =  $(\beta+1) P R / \beta (P + R)$

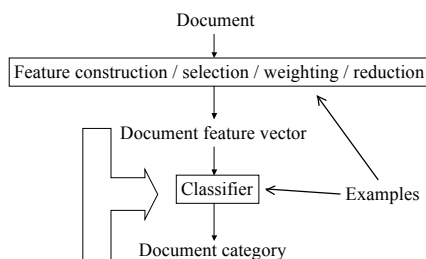
Accuracy =  $a+c / a+b+c+d$

Error rate =  $b+d / a+b+c+d$

## Deductive Approach

- Hand-crafted rules
  - if** (interest and rate) or tax **then** FINANCIAL  
*CONSTRUE* (Reuters)
- Knowledge acquisition bottleneck
  - OK for simple problems (mail sorting)
- Updating for new documents / categories is expensive
- Higher precision-recall (?)
- “*The human tendency to produce classifiers that ‘make sense’ causes them to miss effective corpus-specific language patterns*” (Jamie Callan)

## Inductive (Machine Learning) Approach



## Document Categorization Components

- Document representation
  - vector of  $n$  weighted terms (bag of words)
- Feature selection (Dictionary Creation)
  - single-word
  - phrases
  - positive and negative correlation with category
- Feature weighting
- Categorizer

## Feature selection

- All words except function words (stop list)
- Lemmatization / stemming
- Frequency-based (discard low and high frequency)
- NPs
- Multi-word clusters
- Word-tag combinations

## Feature weights

- $tf-idf = \#(t_k, d_j) \log |Tr|/\#(t_k)$

Term frequency:

How many times does term occur in document?

(inverse) Document frequency:

in how many documents does term occur?

The more a term occurs in a document, the more informative, in the more documents it occurs the less informative

## Dimensionality Reduction

- By feature selection (TSR: Term Space Reduction)
  - document frequency  $\#(t_k)$
  - information-theoretic  $X(t_k, c_i)$ 
    - Information gain (expected mutual information)
    - Chi-square
    - Correlation Coefficient
- By feature construction
  - Singular Value Decomposition
  - Latent Semantic Indexing

## Classifier Construction

- Class systems
  - Binary (= unary?)
  - Multiclass (n-ary) (may be ordinal)
  - Multilabel (may be hierarchical)
- Types of function
  - $F_{ci}: D \rightarrow [0,1]$  (threshold needed, set on validation set)
  - $F_{ci}: D \rightarrow \{0,1\}$
  - $F: D \rightarrow C$
- Some Methods
  - Naïve Bayes (term independence assumption)
  - Memory-Based (= example-based)
  - Rocchio (closeness to centroids of positive / negative cases)
  - Classifier combination

## Learning Methods

- Lazy (k-nn)
- Eager
  - Statistical
    - Linear (threshold) models
    - Linear logistic models
  - Naïve Bayes
  - Support Vector Machines
  - Prototype-based (Rocchio)
  - Rule-based
    - Decision trees, decision lists, rule induction, weighted rules, ...

## Empirical Comparison

- Yang & Liu (1999, SIGIR)
- Direct and indirect comparison
  - Best methods: memory-based, regression, neural nets
  - Rocchio and Naïve Bayes perform worst
- Much more experimentation is needed !

## TROS-case (Martijn Spitters, 1999)

- TROS database of newspaper articles (4000 hierarchically organized categories)
- Bag of words representation (binary vector)
- Feature selection
  - $X^2$  statistic
  - correlation coefficient

## TROS case

- Classifier
  - One classifier per domain (e.g. genetics, music)
  - Positive and Negative set (closest negatives) of documents
  - 100, 200, 400 best selected features
  - Stemming doesn't help
    - e.g. "gedood" (assaults) vs. "dood" (general)

## TROS case

- Classifier
  - Bi-grams
    - Tony Blair, Hoge Raad, dodelijke aanslagen, ...
  - Relevancy Signatures (Riloff)
    - e.g. subject+passive-verb, infinitive+prep+np
  - Tri-grams
    - e.g. in vitro fertilisatie, officier van justitie

	# features	TIMBL			C5.0	
		Recall	Precision	F-score	F-score	F-score
Genetics	100	0.77	0.52	0.62	0.54	
	200	0.87	0.49	0.63	0.53	
	400	0.90	0.49	0.64	0.44	
Music	100	0.89	0.68	0.77	0.74	
	200	0.93	0.62	0.75	0.72	
	400	0.96	0.56	0.71	0.77	

	TIMBL	Stemming	Bigrams	Terms+Bigrams	Signatures	k = 5
F-score	0.62	0.60	0.68	0.71	0.76	0.78
	0.77	0.75	0.73	0.75	0.78	0.80

## SPAM filtering

- Unsolicited Bulk E-mail
- Detecting spam mail
  - Non-existing sender
  - Prevent mail from known "spam" sites
  - Text categorization based on content and form

## SPAM filters

- Should be adaptive (changing trends in spamming)
- Should have high precision (and useful recall)
- Preferably tuned toward individual user

## Simple Bayesian Approach

- “lingspam” email database, 293 emails
- 243 non-spam emails (prior probability 0.83)
- 50 spam emails (0.17)
- The word **cash**
  - 5 times in non-spam (0.02)
  - not in remaining 238 non-spam (0.98)
  - 18 times in spam (0.36)
  - not in remaining 32 spam (0.64)

## Conditional probabilities

word	with spam + / -	with non-spam + / -
cash	0.36 / 0.64	0.02 / 0.98
linguistics	0.02 / 0.98	0.47 / 0.53
you	0.94 / 0.06	0.45 / 0.55
english	0.02 / 0.98	0.32 / 0.68

## Test Document 1

- Did you read the new linguistics text book by Kay?
- Probability spam:  $0.94 \times 0.02 = \mathbf{0.0188}$
- Probability non-spam:  $0.47 \times 0.45 = \mathbf{0.2115}$
- Class: non-spam

## Test Document 2

- Do you want to earn some easy cash?
- Probability spam:  $0.94 \times 0.36 = \mathbf{0.3384}$
- Probability non-spam:  $0.02 \times 0.45 = \mathbf{0.009}$
- Class: spam

## Test Document 3

- You will never earn any cash with expertise in english linguistics.
- Probability spam:  $0.94 \times 0.36 \times 0.02 \times 0.02 = \mathbf{0.000135}$
- Probability non-spam:  $0.02 \times 0.45 \times 0.32 \times 0.47 = \mathbf{0.00135}$
- Class: non-spam

## SPAM features

- Capitalization, spacing exclamation marks
  - Advertisers S H O U T !
- Typical terms and phrases
  - opportunity, money, sex
- Amounts, checkboxes, arrows, decoration ...

## Gender attribution

- <http://www.bookblog.net/gender/genie.html>
- Koppel / Argamon / Shimonì 2003
- Expectation: there is no difference between *written* language of men and women (as opposed to spoken language).
- British National Corpus (fiction and non-fiction)
- Approach:
  - Extract linguistics features (e.g. with shallow parsing)
  - Machine Learning for classification
- Predictability: ~ 80%

## Results

- Most important differences
  - Use of pronomina (woman use more) and some types of noun modification (men use more).
  - Even in formal language use
    - Women are more “involved” in their language use (interpersonal relations)
    - Men are more “informative” in their language use (referring to objects)
  - Strong correlation of male language use with non-fiction and female language use with fiction.

## Results

- “Male” words  
*a, the, that, these, one, two, more, some*
- “Female” words  
*I, you, she, her, their, myself, yourself, herself*
- Are these differences cognitive or sociological?

## Document classification from unlabeled data

- Nigam et al. 2000 (Machine Learning)
- Induce classifier (Naïve Bayes) from small number of hand-labeled documents
- Apply classifier to large number of unlabeled documents (probabilistic class assignment)
- Retrain until convergence (EM)

## Document classification from unlabeled data

- Reduction of classification error up to 30%
- Why does it work?
  - Joint probability distribution over words
  - Unlabeled data → Gaussian mixture components, associated with labels using labeled data
- Why does it sometimes fail?
  - Distributions are not always Gaussian