The **automated categorisation** (or classification) **of texts** into topical categories has a long history, dating back at least to 1960.

Until the late ’80s, the dominant approach to the problem involved *knowledge-engineering* automatic categorisers, i.e. manually building a set of rules encoding expert knowledge on how to classify documents.

In the ’90 s, a newer paradigm based on **Machine learning** has superseded the previous approach.

Document *categorization* (or *classification*) may be seen as the task of determining an *assignment* of value from {0,1} to each entry of the *decision matrix:*

	d_1	...	...	d_j	...	...	d_n
c_1	a_{11}	...	...	a_{1j}	...	...	a_{1n}
...	...	...	...	...	...	...	...
c_i	a_{i1}	...	...	a_{ij}	...	...	a_{in}
...	...	...	...	...	...	...	...
c_m	a_{m1}	...	...	a_{mj}	...	...	a_{mn}

where $C = \{ c_1,\ldots,c_m \}$ is set of predefined *categories*, and $D = \{ d_1,\ldots,d_n \}$ is a set of documents to be categorised. A value of 1 for a_{ij} is interpreted as a decision to file d_j under c_i, while value of 0 is interpreted as decision not to file d_j under c_i.

Observations:

• *Categories* are just symbolic labels. No additional knowledge of their “*meaning*” is available.

• The attribution of documents to categories should, in general, be attributed on the basis of the *content* of the documents, and not on the basis of *metadata.*

Applications:

Automatic indexing for Boolean information retrieval systems.

Document organisation: In general, all issues pertaining to document organisation and filing , be it for purposes of personal organisation or document repository structuring, may be addressed by automatic categorisation techniques.

Document filtering refers to the activity of categorising *dynamic*, rather than static, collection of documents, in the form of a stream of incoming documents dispatched in an asynchronous way by an information producer to an information consumer.

Word sense disambiguation: *Word sense disambiguation* refers to the activity of finding, given the occurrence in text of an ambiguous (i.e. polysemous or homonymous) word, the word sense the word refers to.

Yahoo!-style search space categorisation: Automatically categorising Web pages, or sites, into one or several of the categories that make up commercial hierarchical catalogues such as those embodied in Yahoo!,Infoseek, etc.

The machine learning approach relies on the existence of an initial corpus C_o = { d_1,…, d_s } of documents previously categorised under the same set of categories C = { c_1,…,c_m } with which the categoriser must operate. This means that the corpus comes with a correct decision matrix

	Training set				Test set			
	$\overline{d}_1$	…	…	$\overline{d}_g$	$\overline{d}_{g+1}$	…	…	$\overline{d}_s$
c_1	ca_{11}	…	…	ca_{1g}	$ca_{1(g+1)}$	…	…	ca_{1s}
…	…	…	…	…	…	…	…	…
c_i	ca_{i1}	…	…	ca_{ig}	$ca_{i(g+1)}$	…	…	ca_{is}
…	…	…	…	…	…	…	…	…
c_m	ca_{m1}	…	…	ca_{mg}	$ca_{m(g+1)}$	…	…	ca_{ms}

A value of 1 for ca_{ij} is interpreted as an indication from the expert to file d_j under c_i, while a value of 0 for it is interpreted as an indication from the expert not to file d_j under c_i.
- *training set* T_r = { d_1,…, d_g } .
- *test set* T_e = { d_{g+1},…, d_s } .

One may define the *generality* $gC_o(c_i)$ of category c_i relative to corpus C_o s the percentage of documents that belong to c_i, i.e.:

$$g_{Co}(c_i) = \frac{|\ \{\overline{d}_j \in Co \mid ca_{ij} = 1\}\ |}{|\ \{\overline{d}_j \in Co\}\ |}$$

Problems:

- Indexing and dimensionality reduction
- Feature selection
- Feature extraction
- Building a classifier
- Determining thresholds
- Evaluation

Indexing

Each document is usually represented by a vector of n weighted *index terms (words)*. Weights usually range between 0 and 1.

For determining the weight w_{jk} of term t_k in document d_j the standard *tf idf* weighting function is used, defined as

$$tfidf(t_k, d_j) \quad = \#(t_k, d_j) \cdot log \frac{|Tr|}{\#(t_k)}$$

where $\#(t_k,d_j)$ denotes the number of times t_k occurs in d_j, $\#(t_k)$ denotes the number of documents in T_r in which t_k occurs at least once (also known as the *document frequency* of term t_k), and $|\cdot|$ is the cardinality of a set.

Indexing

In order to make weights fill in the [0,1] interval and documents be represented by vectors of equal length, the weights resulting from *tf idf* are often normalised by cosine normalisation, given by:

$$w_{jk} \quad = \quad \frac{tfidf(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|} (tfidf(t_s, d_j))^2}}$$

where T is the set of all terms that occur at least once in T_r.

Dimensionality reduction (DR)

There are two quite distinct ways of viewing DR:

• *local dimensionality reduction*: for each category c_i, $r_i << r$ features are chosen in terms of which the classifier for category c_i will operate.

• *global dimensionality reduction*: $r_i << r$ features are chosen in terms of which the classifiers for all categories $C = \{ c_1,\ldots,c_m \}$ will operate.

A second, orthogonal distinction may be drawn in terms of what kind of features are chosen:

• *dimensionality reduction* ***by feature selection***: the chosen features are subset of the original r features;

• *dimensionality reduction* ***by feature extraction***: the chosen features are not subset of the original r features.

Feature selection

Given fixed $r'<<r$, techniques for feature selection (or Term Space Reduction -TSR-) purport to select, from the original set of r features, the r' terms that, when used for document indexing, yield the smallest reduction in effectiveness with respect to the effectiveness that would be obtained by using full-blown representations.
Global TSR is usually tackled by keeping the $r_i'<<r$ terms that score highest according to a predetermined numerical function that measures the “importance” of the term for the categorisation task.

Function	Denoted by	Mathematical form
Document frequency	$\#(t_k, c_i)$	$P(t_k, c_i)$
Information gain	$IG(t_k, c_i)$	$P(t_k, c_i) \cdot log \frac{P(t_k, c_i)}{P(c_i) \cdot P(t_k)} + P(\bar{t}_k, c_i) \cdot log \frac{P(\bar{t}_k, c_i)}{P(c_i) \cdot P(\bar{t}_k)}$
Chi-square	$\chi^2(t_k, c_i)$	$\frac{g \cdot [P(t_k, c_i) \cdot P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i) \cdot P(\bar{t}_k, c_i)]^2}{P(t_k) \cdot P(\bar{t}_k) \cdot P(c_i) \cdot P(\bar{c}_i)}$
Correlation coefficient	$CC(t_k, c_i)$	$\frac{\sqrt{g} \cdot [P(t_k, c_i) \cdot P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i) \cdot P(\bar{t}_k, c_i)]}{\sqrt{P(t_k) \cdot P(\bar{t}_k) \cdot P(c_i) \cdot P(\bar{c}_i)}}$
Relevancy score	$RS(t_k, c_i)$	$log \frac{P(t_k \mid c_i) + d}{P(\bar{t}_k \mid \bar{c}_i) + d}$

Feature extraction

Given a fixed $r' << r$, feature extraction purports to synthesize, from the original set of r features, a set of r new features that maximises the obtained effectiveness.

• Term Clustering. *Term clustering a*ims at grouping words with high degree of pairwise semantic relatedness into clusters, so that the clusters (or their centroids) may be used instead of the terms as dimensions of the vector space.
• Latent Semantic Indexing. This technique compresses vectors representing either documents or queries into other vectors of lower-dimensional space whose dimensions are obtained as combinations of the original dimensions by looking at their patterns of co-occurrence.

Building a classifier

The inductive construction of classifier for am category c_i *in C* usually consists of two different phases:

1. the definition of a function CSV_i : *D to* [0, 1] that, given document *d*, returns a *categorisation status value* for it, i.e. number between 0 and 1 that represents the evidence for the fact that *d* should be categorised under *ci*.
2. the definition of a *threshold* t_i such that $CSV_i(d) >= t_i$ is interpreted as a decision to categorise *d* under c_i, while $CSV_i(d) < t_i$ is interpreted as a decision *not* to categorise *d* under c_i.

Regarding the first issue, we can distinguish two main ways to build a classifier:

• *parametric*. According to this approach, training data are used to estimate parameters of a probability distribution. Example: Naive Bayes Classifier.
• *non-parametric*: profile-based, and example-based.

Building a classifier: Non-Parametric Methods

• *Profile-based* (or *linear*) *classifier* is basically a classifier which embodies an explicit, or declarative, representation of the category on which it needs to take decisions. The learning phase consists then on the extraction of the profile of the category from the training set. **Rocchio's classifier** is defined by the formula:

$$w_{yi} \;=\; \beta \cdot \sum_{\{\vec{d}_j \mid ca_{ij}=1\}} \frac{w_{yj}}{|\{\vec{d}_j \mid ca_{ij}=1\}|} + \gamma \cdot \sum_{\{\vec{d}_j \mid ca_{ij}=0\}} \frac{w_{yj}}{|\{\vec{d}_j \mid ca_{ij}=0\}|}$$

where Ò + Í =1, Ò >= 0, Í >= 0 and w_{yj} is the weight that term t_y has in document d_j. In this formula, Ò and Í are control parameters that allow setting the relative importance of positive and negative examples. In general, the Rocchio classifier rewards the closeness of a document to the centroid of the positive training examples, and its distance from the centroid of the negative training examples.

• *Example-based*: k nearest neighbors.

Setting the thresholds

CSV thresholding or **SCUT**. In this case the threshold τ_i is value of the CSV_i function.

Proportional thresholding or **PCUT**. The aim of this policy is to set the threshold ô$_i$ so that the test set generality $gTe(c_i)$ of category c_i is as close as possible to its training set generality $gTr(c_i)$. This idea encodes the quite sensible principle according to which both in training set and test set the same percentage of documents of the original set should be classified under c_i.

Measures of efectiveness

Classification effectiveness is measured in terms of *precision* (Pr) and *recall* (Re).

Precision wrt c_i (Pr_i) is defined as the conditional probability $P(ca_{ix}=1 \mid a_{ix}=1)$, i.e. as the probability that if random document d_x is categorised under c_i, this decision is correct.

Analogously, recall wrt c_i (Re_i) is defined as the conditional probability $P(a_{ix}=1 \mid ca_{ix}=1)$, i.e. as the probability that, if random document d_x should be categorised under c_i ,this decision is taken.

$$Pr = \frac{\sum_{i=1}^{m} Pr_i}{m} \qquad Re = \frac{\sum_{i=1}^{m} Re_i}{m}$$