Text Categorization

- Text categorization (TC - aka text classification) is the task of buiding text classifiers, i.e. sofware systems that classify documents from a domain D into a given, fixed set $C = \{c_1,\ldots,c_m\}$ of categories (aka classes or labels)

- TC is an approximation task, in that we assume the existence of an 'oracle', a target function that specifies how docs ought to be classified.
- Since this oracle is unknown, the task consists in building a system that 'approximates' it

Text Categorization

- We will assume that categories are symbolic labels; in particular, the text constituting the label is not significant. No additional knowledge of category 'meaning' is available to help building the classifier

- The attribution of documents to categories should be realized on the basis of the content of the documents. Given that this is an inherently subjective notion, the membership of a document in a category cannot be determined with certainty

Single-label vs Multi-label TC

- TC comes in two different variants:
  - Single-label TC (SL) when exactly one category should be assigned to a document
    - The target function in the form $\Phi : D \rightarrow C$ should be approximated by means of a classifier $\Phi' : D \rightarrow C$
  - Multi-label TC (ML) when any number {0,…,m} of categories can be assigned to each document
    - The target function in the form $\Phi : D \rightarrow P(C)$ should be approximated by means of a classifier $\Phi' : D \rightarrow P(C)$
- We will often indicate a target function with the alternative notation $\Phi : D \times C \rightarrow \{-1,+1\}$.

  Accordingly, a document d_j is called a positive example of c_i if $\Phi(d_j,c_i)=+1$, and a negative example of c_i if $\Phi(d_j,c_i)=-1$

Category and document pivoted categorization

- We may want to apply a ML classifier in to alternative ways:
  - Given $d_j \in D$, find all the category c_i under which it should be filed (document-pivoted categorization - DPC)
  - Given $c_i \in C$, find all the documents d_j that should be filed under it (category-pivoted categorization - CPC)
- The distinction may be important when the sets C or D are not available in their entirety from the start
  - DPC is suitable when documents become available one at a time (e.g. in e-mail filtering)
  - CPC is suitable when a new category c_{m+1} is added to C after a number of docs have already been classified under C (e.g. in patent classification)

"Hard" and "soft" categorization

- Fully automated classifiers need to take a 'hard' binary decision for each pair $\langle d_j, c_i \rangle$
- Semi-automated, 'interactive' classifiers are instead obtained by allowing 'soft' (i.e. real-valued) decisions:
  - Given $d_j \in D$ a system might rank the categories in C according to their estimated appropriateness to d_j (category ranking task)
  - Given $c_i \in C$ a system might rank the documents in D according to their estimated appropriateness to c_i (document ranking task)

"Hard" and "soft" categorization

- Such ranked lists would be of great help to a human expert in charge of taking the final categorization decision
  - They can thus restrict to the items at the top, rather than having to examine the entire set
- Semi-automated classifiers are useful especially in critical applications where the effectiveness of an automated system may be significantly lower than that of a human expert

Application of TC

TC has been used in a number of different applications

- Automatic indexing for Boolean IR
- Document organization
- Document filtering (e-mail filtering, spam filtering)
- Categorization of web pages into hierarchical catalogues

Automatic indexing for Boolean IR

- The application that spawned most of the early research in TC is that of automatic document indexing for use in Boolean IR systems
- Each document is assigned one or more keywords belonging to a controlled dictionary. Usually, this is performed by trained human indexers, thus a costly activity
- If the entries in the controlled dictionary are viewed as categories, document indexing is an instance of TC
- This is a multi-label task, and document-pivoted categorization is used
- This form of automated indexing may also be viewed as a form of automated metadata generation (or ontology learning), which is going to be very important for the 'Semantic Web'

Document Organization

- Many issues pertaining to document organization and filing, be it for purposes of personal organization or document repository structuring, may be addressed by automatic categorization techniques.
- Possible instances are:
  - Classifying 'incoming' articles at a newspaper
  - Grouping conference papers into sessions
  - Assigning a paper to a review to the more appropriate expert reviewer
  - Classifying patents for easing their later retrieval

Document filtering

- Document filtering (DF) is the categorization of a dynamic stream of incoming documents dispatched in an asynchronous way by an information consumer
- A typical example is a newsfeed (the information producer is a news agency and the information consumer is a newspaper). In this case, the DF system should discard the documents the consumer is not likely to be interested in
- A DF system may be installed
  - At the producer end - to route the info to the interested users only
  - At the consumer end – to block the delivery of info deemed interesting to the consumer

Other applications

- Author (or author's gender) identification for documents of disputed paternity [deVel01,Koppel02,Diederich03]
- Automatic identification of text genre [Finn02,Lee&Myaeng02,Liu03] or Web page genre [MeyerZuEissen&Stein04]
- Polarity detection (aka 'sentiment classification')[Pang02,Turmey02,Kim&Hovy04]
- Multimedia document categorization through caption analysis [Sable&Hatzivassiloglu99]
- Speech categorization through speech recognition+TC [Myers00,Schapire&Singer00]
- Automatic survey coding [Giorgetti&Sebastiani03]
- Text-to-speech synthesis for news reading [Alias02]
- Question type classification for question answering [Li&Roth02,Zhang&Lee03]

Machine Learning (ML) & TC

- In the '80s, the typical approach used for the construction of TC system involved hand-crafting an expert system consisting of a set of rules, one per category, of the form
  - If <DNF formula> then <category> else ¬<category>
  - Where <DNF formula> is a disjunction of conjunctive clauses
- The drawback of this "manual" approach is the knowledge acquisition bottleneck: since rules must be manually defined, building a classifier is expensive, and if the set of categories is updated or the classifier is ported to a different domain, other manual work has to be done.

Example

If	((wheat & farm)	**or**	
	(wheat & commodity)	**or**	
	(bushels & export)	**or**	
	(wheat & tonnes)	**or**	
	(wheat & winter & :soft))	**then**	WHEAT else : WHEAT

Induction

- Since the early '90s, the machine learning approach to the construction of TC systems has become dominant.
- A general inductive process automatically builds a classifier for a category c by 'observing' the characteristics of a set of documents previously classified belonging (or not) to c_i, by a domain expert.
- This is an instance of Supervised Learning (SL).

Advantages of the SL approach

- The engineering effort goes towards the construction, not of a classifier, but of an automatic builder of classifiers (learner)
- If the set of categories is updated, or if the system is ported to a different domain, all that is need is a different set of manually classified documents
- Domain expertise (for labeling), and not knowledge engineering expertise, is needed. Easier to characterize a concept extensionally than intentionally.
- Sometimes the preclassified documents are already available
- The effectiveness achievable nowadays by these classifiers rivals that of hand-crafted classifiers and that of human classifiers

Training Set and Test Set

- The ML approach relies on the application of a train-and-test approach to a labeled corpus $Tr=\{d_1,\dots,d_{|S|}\}$, i.e. a set of documents previously classified under $C=\{c_1,\dots,c_m\}$.
- The value of the function $\Phi : D \times C \rightarrow \{-1,+1\}$ are known for every pair $\langle d_j,c_i \rangle$. Tr then constitutes a 'glimpse' of the ground truth.
- We assume that pair $\langle d_j,c_i \rangle$ are extracted according to a probability distribution $P(d_j,c_i)$
- For evaluation purposes, a new set Te is usually provided which has elements extracted from the same pair distribution $P(d_j,c_i)$ used for elements in Tr.

Model Selection and Hold-out

- Most of the time, the learner is parametric. These parameters should be optimized by testing which values of the parameters yield the best effectiveness.
- Hold-out procedure
  1. A small subset of Tr, called the validation set (or hold-out set), denoted Va, is identified
  2. A classifier is learnt using examples in Tr-Va.
  3. Step 2 is performed with different values of the parameters, and tested against the hold-out sample
- In an operational setting, after parameter optimization, one typically re-trains the classifier on the entire training corpus, in order to boost effectiveness (debatable step!)
- It is possible to show that the evaluation performed in Step 2 gives an unbiased estimate of the error performed by a classifier learnt with the same parameters and with training set of cardinality |Tr|-|Va|<|Tr|

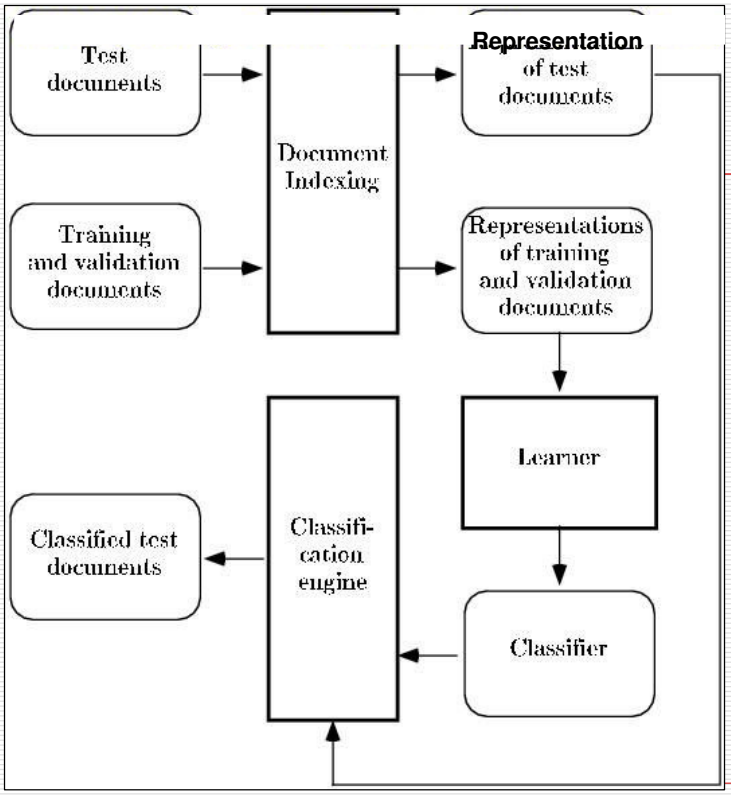
K-fold Cross Validation

- An alternative approach to model selection (and evaluation) is the K-fold cross-validation method
- K-fold CV procedure
  - K different classifiers $h_1,h_2,\ldots,h_k$ are built by partitioning the initial corpus Tr into k disjoint sets $Va_1,\ldots,Va_k$ and then iteratively applying the Hold-out approach on the k-pairs $\langle Tr_i = Tr\text{-}Va_i, Va_i \rangle$
  - Effectiveness is obtained by individually computing the effectiveness of $h_1,\ldots,h_k$, and then averaging the individual results
- The special case k=|Tr| of k-fold cross-validation is called leave-one-out cross-validation

The TC Process

The text categorization process consists of the following phases

- Document Indexing. This take as input the training, validation, and test documents, and outputs internal representations for them. Techniques of traditional IR (and information theory)
- Classifier Learning. This takes as input the representations of the training and validation sets and outputs a classifier. Techniques of ML.
- Classifier Evaluation. This takes as input the results of the classification of the test set, and is mostly accomplished by evaluation techniques belonging to both the IR and the ML tradition.



Architecture of a text classification system