Active Learning for Multimedia

Georges Quénot
Multimedia Information Retrieval Group

September 24, 2007

Tutorial Outline

• Introduction / example
• TRECVID and evaluation
• Active learning principles
• Application categories
• Implementation aspects
• Some works in active learning
• A case study in the context of TRECVID
  – Part 1: Evaluation of active learning strategies
  – Part 2: TRECVID 2007 collaborative annotation
• Conclusion and perspectives
Active learning

Two meanings:

• **Human active learning**: when the teacher requires an active participation of the pupils not just that they passively listen.

• **Machine active learning**: supervised machine learning in which the learning system interacts with a teacher / annotator / oracle to get new samples to learn from.

We consider here only **machine** active learning.
Learning a concept from labeled examples

Raw data: need for a teacher / annotator / oracle / user → human intervention → high cost

Learning a concept from labeled examples

Full annotation: possibly optimal in quality but highest cost

Cats:

Cats?

Non cats:
Learning a concept from labeled examples

Partial annotation: less costly, possibly of similar quality but need to select “good” examples for annotation

Cats:

Cats?

Non cats:

Learning a concept from labeled examples

Incremental partial annotation: samples for annotation are selected on the basis of a class prediction using a learning system → relevance feedback or query learning

Cats:

Cats?

Non cats:
TRECVID and Evaluation

TRECVID
“High Level Feature” detection task

From NIST site:

- Text Retrieval Conference (TREC): encourage research in information retrieval by providing a large test collection, uniform scoring procedures, and a forum for organizations interested in comparing their results.


- High Level Feature (HLF) detection task: contribute to work on a benchmark for evaluating the effectiveness of detection methods for semantic concepts.
TRECVID 2006
“High Level Feature” detection task

• Find 39 concepts (High Level Features, LSCOM-lite) in 79484 shots (160 hours of Arabic, Chinese and English TV news).

• For each concept, propose a ranked list of 2000 shots.

• Performance measure: Mean (Inferred) Average Precision on 20 concepts.

• Distinct fully annotated training set: TRECVID 2005 collaborative annotation on development collection: 39 concepts on 74523 subshots, many of them being annotated multiple times.

• 30 participants. Best Average Precision: 0.192

20 LSCOM-lite features evaluated

<table>
<thead>
<tr>
<th>sports</th>
<th>animal</th>
</tr>
</thead>
<tbody>
<tr>
<td>weather</td>
<td>computer TV screen</td>
</tr>
<tr>
<td>office</td>
<td>US flag</td>
</tr>
<tr>
<td>meeting</td>
<td>airplane</td>
</tr>
<tr>
<td>desert</td>
<td>car</td>
</tr>
<tr>
<td>mountain</td>
<td>truck</td>
</tr>
<tr>
<td>waterscape/waterfront</td>
<td>people marching</td>
</tr>
<tr>
<td>corporate leader</td>
<td>explosion fire</td>
</tr>
<tr>
<td>police security</td>
<td>maps</td>
</tr>
<tr>
<td>military personnel</td>
<td>charts</td>
</tr>
</tbody>
</table>
LSCOM
Large Scale Concept Ontology for Multimedia

• LSCOM: 850 concepts:
  – What is realistic (developers)
  – What is useful (users)
  – What makes sense to humans (psychologists)

• LSCOM-lite: 39 concepts, subset of LSCOM.

• Annotation of 441 concepts on ~65K subshots of the TRECVID 2005 development collection.

• 33,508,141 concept × annotations → About 20,000 hours or 12 man × years effort at 2 seconds/annotation.

• Possibly the same efficiency using active learning with only a 2 or 3 man × years effort.

Frequency of hits by features
[from Paul Over and Wessel Kraaij, 2006]
Metrics: precision and recall
From relevant and non relevant sets

- Recall = \( \frac{\text{Retrieved and Relevant}}{\text{Relevant}} \) = \( \frac{\text{Corrects}}{\text{Relevant}} \)
- Precision = \( \frac{\text{Retrieved and Relevant}}{\text{Retrieved}} \) = \( \frac{\text{Corrects}}{\text{Retrieved}} \)
- F-measure = \( \frac{2 \times \text{Corrects}}{\text{Retrieved} + \text{Relevant}} \)
- Error rate = \( \frac{\text{False positives} + \text{False negatives}}{\text{Relevant}} \)
Metrics: Recall × Precision curves
From ranked lists

• Results ranked from most probable to least probable: more informative that just “relevant / non relevant”.
• For each \( k \): set \( Ret_k \) of the \( k \) first retrieved items
• Fixed set \( Rel \) of the relevant items
• For each \( k \): Recall(\( Ret_k, Rel \)), Precision(\( Ret_k, Rel \))
• Curve joining the (Recall, Precision) points with \( k \) varying from 1 to \( N = \) total number of documents.
• Interpolation : Precision = f(Recall) → Continuous curve

• “Standard” program: \texttt{trec_eval}
  (ranked lists, relevant sets) → RP curve, MAP, ...

Area under the curve: Mean Average Precision (MAP)
Active learning principles

Active learning

- Machine learning:
  - Learning from data.

- Supervised learning:
  - Learning from labeled data: human intervention.

- Incremental learning:
  - Learning from training sets of increasing sizes,
  - Algorithms to avoid full retrain of the system at each step.

- Active learning:
  - Selective sampling: select the “most informative” samples for annotation: optimized human intervention.

- Offline active learning: indexing (classification).
- Online active learning: search (relevance feedback).
Supervised learning

• A machine learning technique for creating a function from training data.
• The training data consist of pairs of input objects (typically vectors) and desired outputs.
• The output of the function can be a continuous value (regression) or a class label (classification) of the input object.
• The task of the supervised learner is to predict the value of the function for any valid input object after having seen a number of training examples (i.e. pairs of input and target output).
• To achieve this, the learner has to generalize from the presented data to unseen situations in a “reasonable” way.
• The parallel task in human and animal psychology is often referred to as concept learning (in the case of classification).
• Most commonly, supervised learning generates a global model that helps mapping input objects to desired outputs.

(http://en.wikipedia.org/wiki/Supervised_learning)

Supervised learning

• Target function: \( f : X \rightarrow Y \)
  \[ x \rightarrow y = f(x) \]
  – \( x \): input object (typically vector)
  – \( y \): desired output (continuous value or class label)
  – \( X \): set of valid input objects
  – \( Y \): set of possible output values

• Training data: \( S = (x_i, y_i)_{1 \leq i \leq I} \)
  – \( I \): number of training samples

• Learning algorithm: \( L : (X \times Y)^* \rightarrow Y^X \)
  \[ S \rightarrow f = L(S) \]

• Regression or classification system: \( y = [L(S)](x) = g(S, x) \)

\( (X \times Y)^* = \bigcup_{n \in \mathbb{N}} (X \times Y)^n \)
Model based supervised learning

- Two functions, “train” and “predict”, cooperating via a Model

- General regression or classification system:
  \[ y = [L(S)](x) = g(S,x) \]

- Building of a model (train):
  \[ M = T(S) \]

- Prediction using a model (predict):
  \[ y = [L(S)](x) = g(S,x) = P(M,x) = P(T(S),x) \]

---

Supervised learning
Classification problem

Training samples
\[ S = (x_i, y_i)_{1 \leq i \leq n} \]

\[ M = T(S) = T((x_i,y_i)_{1 \leq i \leq n}) \]

Testing samples
\[ x \]

\[ y = P(M,x) = P(T(S),x) \]
Supervised learning
Classification problem

Annotate

Class judgments

Training samples

Train

Model

Testing samples

Predict

Predicted classes

\[ y_i = A(x_i) \quad (A \Leftrightarrow f) \]

\[ C = (y_i)_{1 \leq i \leq l} \]

\[ M = T(S) = T((x_i, y_i)_{1 \leq i \leq l}) \]

\[ y = P(M, x) = P(T(S), x) \]

Incremental supervised learning
Classification problem

- Training set of increasing sizes \((I_k)_{1 \leq k \leq K}\):
  \[ S_k = (x_i, y_i)_{1 \leq i \leq I_k} \]
  \[ U_k = (x_i)_{1 \leq i \leq I_k} \]
  \[ C_k = (y_i)_{1 \leq i \leq I_k} \]

- Model refinement:
  \[ M_k = T(S_k) \]

- Prediction refinement:
  \[ y_k = P(M_k, x) \]
  \[ y = P(M_K, x) \]

- Possible incremental estimation \((k > 1)\):
  \[ M_k = T'(M_{k-1}, S_k - S_{k-1}) \]

- Useful for large data sets, model adaptation (concept drift), …
Incremental supervised learning
Classification problem

Training samples
\( S_k = (x_i, y_i)_{(1 \leq i \leq I_k)} \)

Train
Models
\( M_k = T(S_k) = T((x_i, y_i)_{(1 \leq i \leq I_k)}) \)

Testing samples
Predict
Predicted classes

\( x \)

\( y_k = P(M_k, x) = P(T(S_k), x) \)

\( y = P(M_k, x) = P(T(S_k), x) \)

Incremental supervised learning
Classification problem

Annotate

Class judgments
\( y_i = A(x_i) \quad (A \Leftrightarrow f) \)

\( C_k = (y_i)_{(1 \leq i \leq I_k)} \)

Training samples
\( U = (x_i)_{(1 \leq i \leq I)} \)

Train
Models
\( M_k = T(S_k) = T((x_i, y_i)_{(1 \leq i \leq I_k)}) \)

Testing samples
Predict
Predicted classes

\( x \)

\( y = P(M_k, x) = P(T(S_k), x) \)
Active learning basics

- Concept classification → “Semantic gap” problem.
- Improve classification performance?
  - Optimize the model and the train/predict algorithm.
  - Get a large training set: quantity, quality, ...
- Cost of corpus annotation:
  - Getting large corpora is (quite) easy and cheap (already there).
  - Getting annotations on it is costly (human intervention).
- Active learning:
  - Use an existing system and heuristics for selecting the samples to annotate → need of a classification score.
  - Annotate first or only the samples that are expected to be the most informative for system training → various strategies.
  - Get same performance with less annotations and/or get better performance with the same annotation count.

Supervised learning
Classical approach

(flowchart showing the process from training samples to predicted classes)
Active learning classification

Active learning basics

- Incremental process:
  - Needs at least one classification system (several for some strategies).
  - Small increments are better → compromise with system re-training cost.
  - “Cold start” problem: needs at least a few sample for each class to bootstrap or start with a “random” or cluster-based strategy.
  - True incremental learning (actual model adaptation) is possible but not necessary.

- Use for classification system training (offline)
- Use for corpus annotation (offline)
- Use during search (relevance feedback, online)
Active learning strategies

• Query by committee (Seung, 1992): choose the samples which maximize the disagreement amongst systems.
• Uncertainty sampling (Lewis, 1994): choose the most uncertain samples, tries to increase the sample density in the neighborhood of the frontier between positives and negatives → improve the system's precision.
• Relevance sampling: choose the most probable positive samples, tries to maximize the size of the set of positive samples (positive samples are most often sparse within the whole set and finding negative samples is easy).
• Choose the farthest samples from already evaluated ones, tries to maximize the variety of the evaluated samples → improve the system's recall.
• Combinations of these, e.g. choose the samples amongst the most probable ones and amongst the farthest from the already evaluated ones.
• Choose samples by groups which maximize the expected global knowledge gain (Souvanavong, 2004).

Simulated active learning

• How efficient is the active learning approach?
• Experiment on strategies and problem parameters.
• Simulated (artificial) active learning:
  – Use of a fully annotated training set.
  – Simulate incremental annotations of the training set using various strategies.
  – Use a distinct testing set (if possible with posterior contents) for concept learning evaluation (not for corpus annotation evaluation)
  – Analyze the effect of various parameters.
• Some reasonable assumptions, e.g. the order in which the annotations are done by the evaluators does not significantly influence their judgments.
Application categories

Application: concept learning

• Most popular application.
• Offline use: mainly used for classifier training, not for interaction with a user.
• Goals:
  – Increase the learning performance for a given annotation cost or
  – Reduce the annotation cost for a given learning performance or
  – Seek for a best annotation cost versus learning performance compromise.
• Evaluation:
  – Simulated active learning.
  – Distinct development and test collection.
  – Mean Average Precision performance metrics.
  – MAP as a function of the annotated fraction of the development set.
  – Comparison with different AL strategies and/or parameter values.
• It does work: huge effects reported in a variety of areas
Application: corpus annotation

- Growing application.
- Offline use: used for corpus annotation, not for interaction with a user.
- Principles:
  - A fraction of the corpus is manually annotated.
  - The remainder of the corpus is automatically annotated using a classifier trained using the manually annotated part.
  - The classifier is only temporarily used for the corpus annotation, not a goal.
- Goals:
  - Increase the full corpus annotation quality for a given manual annotation cost or
  - Reduce the manual annotation cost for a given corpus annotation quality or
  - Seek for a best manual annotation cost versus full annotation quality compromise.
- Evaluation:
  - Simulated active learning.
  - Same collection for development and test.
  - Error rate performance metrics.
  - Error rate as a function of the manually annotated fraction.
  - Comparison with different AL strategies and/or parameter values.
- It does work: significant effects reported in several areas.

Application: search (relevance feedback)

- Popular application.
- Online use: used for interaction with a user.
- Principles:
  - The user information need is considered as a concept to be learnt.
  - An incremental supervised learning system is trained using user feedback.
  - The classifier is only temporarily used for the search task, not a goal.
- Goals:
  - Increase the search result quality for a given number of feedback cycles or
  - Reduce the number of feedback cycles for a given search result quality or
  - Combination of both.
- Evaluation:
  - Simulated active learning and user interaction.
  - Same collection for development and test.
  - MAP on the last result list performance metrics.
  - MAP as a function of the number of feedback cycles.
  - Comparison with different AL strategies and/or parameter values.
- No random baseline (or baseline is no feedback): only comparison between strategies.
Implementation aspects

Online versus offline active learning

• Relevance feedback: everything is online.

• Concept learning and corpus annotation:
  – Offline relatively to the final use but online relatively to the teacher.
  – The teacher may have to wait for the system to select new samples for annotation.
  – The system may have to wait for the teacher to do new annotations.

• The AL iteration cycle has to be optimized
  – Fast system training → possible use of a simplified learning system.
  – Efficient user work → annotate several samples in a series.
Active learning iteration cycle

- Compromise about the step (or chunk) size
  - The smaller the better for AL efficiency.
  - The larger the better for training time cost and teacher work efficiency.

- Interlacing of training and annotating phases for different concepts
  - Better for user work: possibly continuous activity.
  - Still need for a compromise about the step or chunk size.
  - Consider the relation between annotation time and training time → annotation driven classifier retraining.
  - Largely error prone because of frequent changes in the concepts to be annotated.

Parallel annotation of several concepts on one shot / key frame: TRECVID 2003
Parallel annotation of one concepts on several shot / key frame: TRECVID 2005 and 2007

User effects

Annotation errors or ambiguities:

- Inconsistencies of up to 3% between two different annotators have been reported for concept annotations in video shots even in good conditions (TRECVID collaborative annotation 2005). Even worse with more than two annotators.
- Actual ambiguities (how many stairs to make a stairway?).
- True human errors: many possible causes (e.g. helicopter as an airplane or fail to notice a change in concept to annotate).
- Significant impact: false positives and false negatives really hurts system performance.
- Active learning makes things worse because of frequent context changes.
User effects

Active learning can help:

• Avoid full double or triple annotations to remove errors or solve easy ambiguities.
• Ask for a second opinion only for those samples that were misclassified with a strong confidence → Active cleaning.
• Ask for a third opinion only if the second opinion is inconsistent with the first one.
• Use only consistent samples for retraining.
• Could be evaluated using simulated active learning with a multiply fully annotated corpus (e.g. LSCOM – TRECVID 2005).
• Need to arbitrate between new annotations and reannotations.

Use of a fully featured search system

• Beyond the use of a simple classifier.
• Use of a general purpose content-based search system.
• The concept to be annotated is the query: search by active learning but not limited to relevance feedback.
• Multi-criteria search (video example, not exhaustive):
  – Keywords from Automatic Speech Recognition.
  – Image examples.
  – Already trained and indexed (other) concepts.
  – Visual similarity to already found positive samples.
  – Temporal closeness to already found positive samples.
• Mostly useful for sparse concepts (general case).
• Possible solution to the cold start problem (but may require some system training from other sources).
• Not reported yet but promising.
Use of a fully featured search system

Game-based collaborative annotation

The ESP Game is a two-player game. Each time you play you are randomly paired with another player whose identity you don't know. You can't communicate with your partner, and the only thing you have in common with them is that you both saw the same image. The goal is to guess what your partner is thinking on each image. Once you both type the same word(s), you get a new image. Each time you type a word or phrase, you must press enter on your keyboard to submit it to the game. You can type as many guesses as you want, and as soon as a single guess of yours is equal to a guess that your partner has made, you get a new image. You have two and a half minutes to agree on 15 images.

Game images have taboo words, which you can't use, nor can you use any plural, singular, or word related to a taboo word. If one of the taboos for an image is the name of a color, you cannot use any other color as a guess. If you feel that an image is too hard, you can ask to pass by clicking the yellow pass button on the lower right corner. Clicking the button will generate a message on your partner's screen, letting them know that you want to pass. You cannot pass on an image until both you and your partner have hit the pass button.
Game-based collaborative annotation

The ESP Game

As seen on CNN and newspapers around the world!

31,280,732

Labels collected since October 5, 2003
(This number is updated every 12 hours)

Labeling an image means associating word descriptions to it, as shown below. Computer programs can yet determine the contents of arbitrary images, but the ESP game provides a novel method of labeling them: players get to have fun as they help us determine their contents. If the ESP game is played as much as other popular online games, we estimate that all the images on the Web can be labeled in a matter of weeks!

Done

Game-based collaborative annotation

Peekaboom: How to Play - Mozilla Firefox

The Basics: Peeking and Booming

You and a random partner take turns "peeking" and "booming." While one of you is peeking, the other is booming. The booming player (Boom) gets an image along with a word related to the image, and the peeking player (Peek) gets no image (see Figure below). Booming consists of clicking parts of the image; when Boom clicks a part of the image, it is revealed to Peek. The object of the game is for Peek to type the word associated to the image — from their perspective, the game consists of a slowly revealing image, which has to be named. From Boom's perspective, the game consists of clicking on areas of the image so that Peek can guess the word associated to it. Once Peek guesses the correct word, the two of you move on to the next image and switch roles.

Done
Features (not exhaustive)

• Low-level visual features:
  – Color: color histograms or color moments in various color spaces.
  – Texture: Gabor or wavelets transforms.
  – Motion: motion vectors or statistics on motion vectors.
  – Local and global features: Principal Component Analysis for data dimension reduction and noise cleaning.

• Low and intermediate audio features:
  – Word vector representations from Automatic Speech Recognition (ASR).
  – Music / noise / gender detection.
  – Mel Frequency Cepstral Coefficients.

• Intermediate features:
  – Output from further preprocessing: text categories, …

Some works in Active Learning
Queries and Concept Learning  
[Dana Angluin, 1988]

- Queries to instructors for concept learning tasks.
- Queries from the system to a human being (the opposite of a queries in an information retrieval system).
- Problem: identify an unknown set $L^*$ from a finite or countable hypothesis space $L_1, L_2, ...$ of subsets of a universal set $U$.
- The system has access to oracles that can answer specific kinds of queries about the unknown concept $L^*$: membership, equivalence, subset, superset, disjointness, exhaustiveness.
- Majority vote strategy: Identification of the target set in $L_1, ..., L_N$ in $\left\lfloor \log_2 N \right\rfloor$ steps.
- Not easily transposable for multimedia indexing because indexing at the sample level usually supports only the membership query type.

Query by Committee  
[H.S. Seung et al, 1992]

- Mostly cited in literature.
- Committee of students (learning programs).
- Queries from the system to a human being (≠ queries in IR).
- The next query is chosen according to the principle of maximal disagreement.
- Parametric models with continuously varying weights
- Teacher: $\sigma_0(X)$  
  $X$: input vector (output space is $\{-1,+1\}$)
- Student: $\sigma(W;X)$  
  $W$: weight vector of the student function
- The training set is built up one sample at a time: $S_P = (X_t, \sigma_t)(1 \leq t \leq P)$
- Version space: set of all $W$ which are consistent with the training set: $W_P = \{ W : \sigma(W;X_t) = \sigma_t, 1 \leq t \leq P \}$
Query by Committee
[H.S. Seung et al, 1992]

- Flat prior distribution $P_0(W)$:
  \[ P(W | S_p) = \frac{1}{V_p} \text{ if } W \in W_p, \ 0 \text{ otherwise} \text{ with } V_p = \text{volume}(W_p) \]

- Information gain: $I_{P+1} = -\log\left(\frac{V_{P+1}}{V_P}\right)$

- Choose the $X_{P+1}$ that maximizes the information gain (not trivial)

- Two test applications: high-low game and perceptron learning of another perceptron.

- Query by committee learning:
  - Asymptotically finite information gain: the volume consistent with the observation in the parameter space is divided by a fixed finite factor.
  - Generalization error decreases exponentially.

- Random sampling:
  - Asymptotically null information gain.
  - Generalization error decreases with an inverse power law.

---

Query by Committee
[H.S. Seung et al, 1992]

- Suggestion of a criteria for a good query algorithm: asymptotically finite information gain.

- Closer to the multimedia indexing problem (membership only queries) but assumptions that
  - The actual teacher function can be reached by a given $W_0$.
  - The next sample can be chosen arbitrarily in the input space.
  - The parameter space does not vary with the number of samples → correct for a perceptron with a fixed architecture but not for classifiers in which the number of parameters is adjusted to or depends upon the size of the training set (e.g. Support Vector Machines).
Uncertainty sampling
[David Lewis and William Gale, 1994]

- A sequential Algorithm for Training Text Classifiers.
- Membership queries (from system to human, again).
- Use of a probabilistic classifier.
- Algorithm:
  1. Create an initial classifier
  2. While teacher is willing to label examples
     (a) Apply the current classifier to each unlabeled example
     (b) Find the $b$ examples for which the classifier is least certain
         of class membership
     (c) Have the teacher label the subsample of $b$ examples
     (d) Train a new classifier on all labeled examples
- Really close to the multimedia indexing/retrieval problem

Uncertainty sampling
[David Lewis and William Gale, 1994]

- Newswire classification task, use of simulated active learning.
- 319,463 training documents, 51,991 test documents, 10 categories.
- Cold start with 3 randomly chosen positive examples.
- Comparison between:
  - Random sampling (3+7),
  - Relevance sampling (3+996), increment by 4,
  - Uncertainty sampling (3+996), increment by 4,
  - Full annotation (3+319,463).
- The uncertainty sampling reduced by as much as 500-fold the
  amount of training data that would have to be manually classified to
  achieve a given level of effectiveness.
- Uncertainty sampling performs better than relevance sampling.

<table>
<thead>
<tr>
<th></th>
<th>uncertainty</th>
<th>random</th>
<th>relevance</th>
<th>full</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>0.453</td>
<td>0.107</td>
<td>0.248</td>
<td>0.409</td>
</tr>
</tbody>
</table>
SVM active learning
[Simon Tong and Edward Chang, 2001]

- Support Vector Machine Active Learning for Image Retrieval.
- Relevance feedback for learning a “query concept”.
- Select the most informative images to query a user.
- Quickly learn a boundary that separates the images that satisfy the user query concept from the rest of the dataset.

Algorithm:
- Cold start with 20 randomly selected images.
- Iterations with uncertainty sampling: display the 20 images that are the closest to the SVM boundary.
- Final output with relevance sampling: display the 20 images that are the farthest to the SVM boundary (on the positive side).

- Significantly higher search accuracy than traditional query refinement schemes after just three or four rounds of relevance feedback.

Active learning for CBIR
[Cha Zhang and Tsuhan Chen, 2002]

- An active learning framework for Content Based Information Retrieval
- Indexing phase and Retrieval phase.
- Annotation of multiple attributes on each object.
- Indexing via uncertainty sampling based active learning.
- Uncertainty is estimated via the expected knowledge gain.
- Each object (either in the database or from the query) receives a probability associated to each feature: 0 or 1 if annotated, computed probability from the trained classifier otherwise.
- Retrieval via semantic distance between query objects and objects in the database: attribute probabilities are used as a feature vector.

- Weighted sum with low-level features.
- Experiments on a database of 3D objects: discriminate aircrafts from non aircrafts.
- Performance increases with the number of annotated objects.
- Active learning outperforms random sampling based learning.
Partition sampling

[Fabrice Souvannavong et al, 2004]

- Partition sampling for active video database annotation.
- Focus on the simultaneous selection of multiple samples.
- Select samples such that their contribution to the knowledge gain is complementary and optimal.
- Partition the pool of uncertain sample using the k-means clustering technique and select one sample in each cluster → the samples are both mostly uncertain and far from each other.
- Practical implementation:
  - HS color histograms and Gabor energies on keyframes
  - Latent Semantic Analysis (LSA) to capture local information
  - $k$-Nearest Neighbors ($k$NN) classification

Use of TRECVID 2003 development data and annotation

The task is corpus annotation, not concept learning → the development and the test set are identical → the performance measure is the error rate on the whole set

Comparison between
  - Random sampling
  - Greedy maximization of the error reduction
  - Partition sampling

- The partition sampling is significantly better (up to 30%) than greedy AL strategy only when a small fraction of the corpus is annotated.
- No significant difference after the annotation of about 1/6th of the corpus (no more “far” uncertain samples?).
- 0.5 % error rate after the annotation of half of the corpus against 2 % for random sampling: ~4-fold error reduction.
Partition sampling
[Fabrice Souvannavong et al, 2004]

History or instability sampling

- Active Learning with History-Based Query Selection for Text Categorization [Davy and Luu, 2007].
- Select the sample which have the most erratic label assignments.
- Similar to query by committee where the committee members are the classifiers of the $k$ previous iterations.
- History uncertainty sampling: average the uncertainty on the $k$ previous iteration.
- Use of class distributions: works with multiple classes, all possible classes are annotated at once when a sample is selected for annotation.
- History Kullback-Leibler Divergence (KLD): average on The KLD between average distribution and committee member distributions.
- Improvement over both uncertainty sampling and history uncertainty sampling.
History or instability sampling
[Davy and Luu, 2007]

A case study in the context of TRECVID

Part 1

Evaluation of active learning strategies
TRECVID “High Level Feature” detection task

- Fully annotated training set: TRECVID 2005 collaborative annotation on development collection.
- Distinct testing set: TRECVID 2006 test collection
- TRECVID 2006 HLF task and metrics:
  - Find 39 concepts (High Level Features, LSCOM-lite) in 79484 shots (146328 subshots, 160 hours of Arabic, Chinese and English TV news).
  - For each concept, propose a ranked list of 2000 shots.
  - Performance measure: Mean (Inferred) Average Precision on 20 concepts.

Classification system used

- Networks of SVM classifiers for multimodal fusion [Ayache, 2006].
- Combination of early and late fusion schemes.
- Local low level visual features: color, texture and motion on 20×13 patches of 32×32 pixels.
- Global low level visual features: color, texture and motion.
- Intermediate local visual features (percepts): 15 classes (sky, greenery, face, building, ...).
- Intermediate textual categories (percepts): 103 classes (Reuters categories on ASR transcriptions).
- Global performance slightly above median in TRECVID 2006.
Active learning evaluations

• Use of simulated active learning.
• The training set is restricted to the shots that contain speech → 36014 samples.
• Default step size: 1/40th of the training set → 900 samples.
• Cold start with 10 positive and 20 negative all randomly selected.
• Evaluation of:
  – Strategies: random, relevance and uncertainty sampling
  – Relation with concept difficulty
  – Effect of the step size
  – Training set size
  – Finding rates for positive samples
  – Precision versus recall compromise

Three evaluated strategies

• Significant level of fluctuations: smooth increase would be expected.
• Probably due to the progressive inclusion of particularly good or particularly bad positive or negative examples.
• Observed in many other works → average with many different cold start random selections
Three evaluated strategies

- **Random sampling** shows a continuous increase in performance with the size of the sample set with a higher rate near the beginning.
- The maximum performance is reached only when 100% of the sample set is annotated.

![Graph showing performance of different sampling strategies](image)

Three evaluated strategies

- **Relevance sampling** is the best one when a small fraction (less than 15%) of the dataset is annotated.
- It gets very close to the best random sampling performance with the annotation of only about 12.5% of the whole sample set. The performance increases then very slowly.

![Graph showing performance of different sampling strategies](image)
Three evaluated strategies

• **Uncertainty sampling** is the best one when a medium to large fraction (15% or more) of the dataset is annotated.

• It gets slightly over the best relevance and random sampling performances with the annotation of only about 15% of the whole sample set. The performance does not increase afterwards.

Relation with concept difficulty

“Easy” concepts:
Weather (0.454), Sport (0.301) and Maps (0.217).
Relation with concept difficulty

“Moderately difficult” concepts:
Military (0.0985), Car (0.0771), Waterscape-Waterfront (0.0755), Charts (0.0708), Meeting (0.0671), Flag-US (0.0634), Desert (0.0557) and Explosion-Fire (0.0548).

Relation with concept difficulty

“Difficult” concepts:
Computer-TV-screen (0.0411), Truck (0.0355), Mountain (0.0329), People-Marching (0.0284), Police-Security (0.0257), Airplane (0.0206), Animal (0.0058), Office (0.0027) and Corporate-Leader (0.0000).
Finding positive and negative samples

- Number of positive samples found along iterations.
- Relevance sampling founds positives more rapidly but this is not related to better performance, except close to the beginning.

Effect of the step size

- Uncertainty sampling with different step sizes (relatively to the training set size).
- Not surprisingly: the smaller, the better.
Effect of the step size

- Random sampling with different step sizes (relatively to the training set size).
- The step size should have no effect: only fluctuations are seen.

![Graph](image)

Effect of the corpus size

- A single step size is considered: 1/20th of the corpus size.
- Three corpus sizes: 36K, 18K and 9K samples.
- Use of first half and first quarter of the full corpus, not of a random sub selection.
- Asymptotic values for linear and random sampling:

<table>
<thead>
<tr>
<th>Corpus size</th>
<th>9K</th>
<th>18K</th>
<th>36K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear sampling</td>
<td>0.030</td>
<td>0.045</td>
<td>0.090</td>
</tr>
<tr>
<td>Random sampling</td>
<td>0.045</td>
<td>0.070</td>
<td>0.090</td>
</tr>
</tbody>
</table>

- Linear sampling is significantly worse than random sampling.
The three strategies on the 36K corpus

- Uncertainty sampling is the best strategy when a medium to large fraction (15% or more) of the dataset is annotated.
- Relevance sampling is the best strategy when a small fraction (less than 15%) of the dataset is annotated.
- Optimal annotation size: ~7K samples

The three strategies on the 18K corpus

- Uncertainty sampling is the best strategy when a medium to large fraction (20% or more) of the dataset is annotated.
- Relevance sampling is the best strategy when a small fraction (less than 20%) of the dataset is annotated.
- Optimal annotation size: ~5K samples
The three strategies on the 9K corpus

• Relevance sampling is always the best strategy (not enough sample for the uncertain sampling strategy to finally get better?).
• Optimal annotation size: ~3.5K samples

Better values on the whole corpus?

• Both uncertainty and relevance sampling often perform better than random and linear sampling even when the whole set is annotated: why?
• Most concepts are sparse:
  – All positive samples are kept but only a fraction of the negative samples are kept,
  – These are chosen first among those predicted as most relevant or most uncertain.
• Simulated active learning can improve system performance even when the corpus is fully annotated by improving the selection of the negative samples (some advanced learning algorithms already include something equivalent).
Precision versus recall compromise

- Mean Average Precision does not capture everything.
- Precision @ N has quite often a more practical meaning.
- Recall x precision curves for the three strategies when 20% of the corpus is annotated.
- Relevance sampling is more “recall oriented”.
- Uncertainty sampling is more “precision oriented”.
- Statistical significance unsure.
Part 1 conclusion (1)

• Evaluation of active learning strategies using simulated active learning.
• Use of TRECVID 2005/2006 data and metrics.
• Three strategies were compared: relevance sampling, uncertainty sampling and random sampling.
• For easy concepts, relevance sampling is the best strategy when less than 15% of the dataset is annotated and uncertainty sampling is the best one when 15% or more of the dataset is annotated (with 36K samples).
• Relevance sampling and uncertainty sampling are roughly equivalent for moderately difficult and difficult concepts.

Part 1 conclusion (2)

• The maximum performance is reached when 12 to 15% of the whole dataset is annotated (for 36K samples).
• The optimal fraction to annotate depends upon the size of the training set: it roughly varies with the square root of the training set size (25 to 30% for 9K samples).
• Random sampling is not the worst baseline, linear scan is even worse.
• Simulated active learning can improve system performance even on fully annotated training sets.
• Uncertainty sampling is more “precision oriented”.
• Relevance sampling is more “recall oriented”.
• “Cold start” not investigated yet.
A case study in the context of TRECVID

Part 2

TRECVID 2007 collaborative annotation

TRECVID 2007 Collaborative annotation

- Follows TRECVID 2003 and 2005 collaborative annotations.
- Annotations are done by TRECVID participants.
- A tool is provided to the annotators. It is Web-based since 2005.
- Images are displayed to the annotators one concept at a time, one or several images at a time.
- The user marks each image as either positive, negative or unsure (default is negative).
- Cold start using TRECVID 2005 annotations (same concepts but significantly different collection contents).
- Annotation driven active learning.
- Implementation of active cleaning.
TRECVID 2007 Collaborative annotation

Sequential annotation interface

TRECVID 2007 Collaborative annotation

Parallel annotation interface
Annotation driven active learning

- Two engines running in parallel:
  - The web-based annotation engine,
  - The active learning sample ranking engine.

- Active learning is continuously running, cycling on the 36 concepts (in approximately 18 hours).

- The next concept to retrain is the one that received the highest number of annotations since its last training.

- When a user connects it is asked to annotate the concept with the fewer number of annotations in total; when he has annotated at least 100 samples, another concept is proposed to him.

- Active learning is transparent to the user except that he has to switch quite often from one concept to another.

TRECVID 2007 collaborative annotation

- 21,532 subshots to annotate with 36 concepts.
- 32 participating teams.
- Each team was asked to annotate 3% of the subshots x concepts.
- About 92% once-equivalent annotation.
- Some annotated several times due to active cleaning:

<table>
<thead>
<tr>
<th>Pass</th>
<th>Annotated</th>
<th>% Annotated</th>
<th>Negative</th>
<th>Skipped</th>
<th>Positive</th>
<th>% Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pass 1</td>
<td>641223</td>
<td>82.7</td>
<td>578299</td>
<td>13163</td>
<td>49761</td>
<td>7.76</td>
</tr>
<tr>
<td>Pass 2</td>
<td>46864</td>
<td>6.05</td>
<td>11904</td>
<td>7478</td>
<td>27482</td>
<td>58.6</td>
</tr>
<tr>
<td>Pass 3</td>
<td>21987</td>
<td>2.84</td>
<td>9383</td>
<td>4040</td>
<td>8564</td>
<td>39.0</td>
</tr>
<tr>
<td>Pass 4</td>
<td>1492</td>
<td>0.19</td>
<td>324</td>
<td>940</td>
<td>228</td>
<td>15.3</td>
</tr>
<tr>
<td>Synthesis</td>
<td>641223</td>
<td>82.7</td>
<td>578683</td>
<td>15348</td>
<td>47192</td>
<td>7.36</td>
</tr>
</tbody>
</table>
**TRECVID 2007 collaborative annotation**

- About two-month effort.
- Main effort during two weeks (second an third weeks).
- First week not open to public to guarantee a small step size during the first iterations.

![Daily annotations in the collaborative annotation project](image)

Finding positive and negative samples

- Similar to experiments with simulated active learning on TRECVID 2005-2006 data.

![Evolution of the fraction of positive samples](image)
Finding positive and negative samples

- Evolution by concept is very variable.
- A few do worst than random close to the beginning.

![Graph showing evolution of positive samples against annotated samples for 36 concepts.]

Evolution of the fraction of positive samples found with the fraction of annotated samples for the 36 concepts individually.

Annotation driven active learning

- Small step size at the beginning: training driven
- Larger step size afterwards: annotation driven

![Graph showing evolution of positive samples for the "Animal" concept with marking of active learning occurrences.]

Evolution of the fraction of positive samples found with the fraction of annotated samples for the “Animal” concept with marking of the active learning iteration occurrences.
Part 2 conclusion

- Active learning based collaborative annotation.
- Actual active learning, not simulated active learning.
- Heterogeneous cold start (TV 2005 → TV 2007).
- Similar behavior in the finding of positive samples though the strategies and conditions differ.
- Neighbor sampling (select shots just before and just after a positive shot) significantly improve the finding rate.
- Difficult to quantify but active cleaning significantly improve the annotation quality.
- Significant global benefit compared to random sampling but possibly not as high as could have been due to the small collection size.
- Annotation used by TRECVID participants to the HLF detection task.

Global conclusion and perspectives
Global conclusion

- Active learning greatly improve the annotation cost versus system performance quality.
- Moderate additional cost in complexity.
- Main applications: classifier training, corpus annotation and relevance feedback during search.
- Main strategies: relevance sampling, uncertainty sampling and sample clustering (partition sampling) plus combinations of them including evolving strategies.
- Integration with classification techniques: SVM active learning.
- Other parameters: cold start, step size, user effects, concept difficulty, concept frequency, ...
- Practical implementation: organization of the human-system iteration cycle.
- Annotation driven active learning.

Perspectives (1)

- More work on strategies:
  - New strategies: instability, hybrid relevance-uncertainty (e.g. target probability at 0.75 or with an evolving value), ...
  - Classifier principle related strategies,
  - Characterization of strategy efficiency with the application and target concepts.
- More work on features:
  - Not directly related to active learning but very relevant to system performance.
- Adaptation of the strategies to the problem context, to concept frequency and difficulty:
  - Relevance sampling for sparse concepts,
  - Uncertainty sampling for frequent concepts,
  - Frequency and difficulty are not related.
- Variable strategies and step sizes:
  - Switch from relevance sampling to uncertainty sampling,
  - Increasing step size.
Perspectives (2)

• Active learning for annotation cleaning:
  – Human annotation is imperfect and annotation errors really hurt system performance,
  – Optimize the cost of every annotation: compare the benefit of correcting an annotation error to the benefit of getting new annotations,
  – Obvious strategy: double check the wrongly predicted but could certainly be improved.

• Concept derivation and active learning:
  – Use or relation between concepts, e.g. women are humans,
  – Derive generics from specifics,
  – Look for specifics within generics,
  – Which to annotate first? Which relations to use?
  – Not specific to active learning but possibly specific strategies.

• Ontology annotation and active learning: similar to concept derivation but full use of the ontology structure.

Perspectives (3)

• Use of a fully featured search system instead of a simple classification system:
  – Possible solution for the cold start,
  – Improve the finding rate of positive samples,
  – Rely on previous work: capitalizing knowledge.

• Application to local (frame by frame or region) annotation:
  – Need for annotation at the frame (versus shot) level,
  – Need for annotation at the region (versus image) level,
  – Need for better locating the concepts in the document,
  – Need for system training when locality is exploited,
  – Local annotation is very costly but can be very rewarding.
  – Active learning based prediction with manual correction.
  – Significant benefit can be expected.