Quaero consortium at TRECVID 2011 Semantic Indexing Task

Bahjat Safadi, Nadia Derbas, Abdelkader Hamadi, Franck Thollard and Georges Quénot
UJF-LIG

Hervé Jégou
INRIA-Texmex

Tobias Gehrig, Hazım Kemal Ekenel and Rainer Stifelhagen
KIT

5 December 2011

Outline

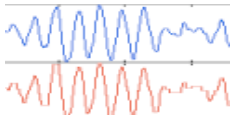
- TRECVID semantic indexing task
- Global system architecture
- Descriptors with optimization
- Classification
- Hierarchical fusion
- Conceptual feedback
- Re-ranking
- Submitted runs
- Conclusion

TRECVID 2011 semantic Indexing task

- Find concepts in video shots
- Train classifiers on development data using the collaborative annotations
- Predict on test data and send ranked lists of 2000 shots to NIST for evaluation
- (Inferred) Mean Average Precision metrics on ground truth produced by NIST using pooling of submissions

The classical classification pipeline

0101



0101

President Clinton is basking in some good news

0101

Descriptor extraction

Classification

Fusion

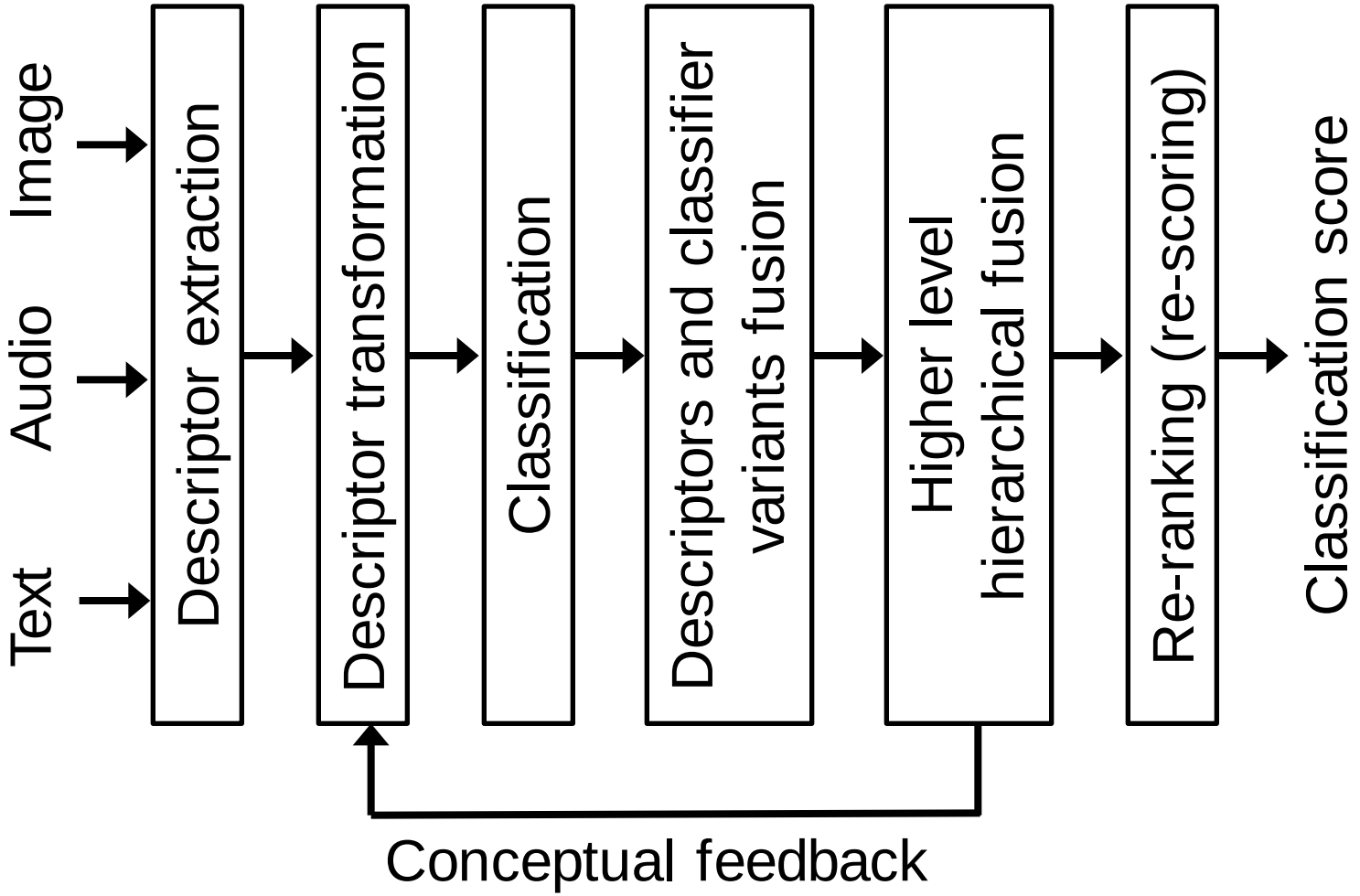
Discourse of President Bill Clinton

Signal

Semantics

Semantic gap

The Quaero classification pipeline



System features

- Use of a large number of descriptors and variants
- Descriptor optimization
- Use of classifier variants
- Late fusion of descriptor and classifier variants
- Further hierarchical late fusion
- Conceptual feedback
- Temporal re-ranking

Descriptors and variants

- Color (histograms), texture (Gabor, quaternionic wavelets), points of interest (SIFT, color SIFT, STIP), percepts, audio (MFCC statistics)...
- Use of spatial (grid-based, pyramid) variants
- Use of other variants: number of bins in histograms, SIFT sampling, histogram fuzziness)
- 15 different types, 47 final variants
- Produced by Quaero partners or shared with external groups

- Gain by fusing variants: 5-15 % (relative on MAP)

Descriptor optimization

- Power transformation (similar to Douze 2010):
  - Many descriptors are histogram-based
  - χ^2 distance is more optimal but more complex to compute and Euclidian transformation not possible
  - $x_i >> y_i$ problem: large component dominates but

$$(x_i - y_i)^2 / (x_i + y_i) \sim \left(\sqrt{x_i} - \sqrt{y_i}\right)^2$$

  - Use $x_i \leftarrow \sqrt{x_i}$ or, more generally $x_i \leftarrow x_i^{\alpha}$
  - α optimized by cross-validation
  - Optimal value close to 0.5 but sometimes quite different
  - Gain from 10% to 100% (not frequent)
  - Gain even for non histogram-based descriptors
  - $\rightarrow$ Euclidian distance becomes appropriate

Descriptor optimization

- PCA reduction
  - Possible only once Euclidian distance is appropriate
  - Significant reduction in the number of components
  - (Generally slight) simultaneous increase in performance
  - Number of kept component optimized by cross-validation on both criteria (good reduction with optimal performance)
  - Typical compression ratio 2:1 to 5:1, sometimes more
  - Gain from 0 to 15 %, typical 0 to 5%
  - Cumulated gain with power normalization

Use of multiple classifiers

- kNN
  - Linear combination of 0 and 1 according to the sample class and with a continuous and decreasing function of the distance to the nearest neighbors
  - Nearest neighbors computed only once whatever the number of target concepts → very efficient
- MSVM
  - Use of multiple SVMs to address the unbalanced data problem: late fusion of many classifiers, each with all the positive samples and with a different fraction of the negative samples (variant of Tahir and Kittler 2008).
  - Improves over regular SVM on highly imbalanced datasets
- MSVM is generally better than kNN but not always
- Late fusion of both almost always improves over the best one by 0 to 10%
- Tuning with kNN is relevant for MSVM

Hierarchical fusion

- Late fusion of descriptor and classifier variants: get the maximum from each descriptor type:
  - fuse spatial variants
  - then fuse other variants
  - finally fuse classification results from different classifiers
- Further hierarchical late fusion: fuse across different descriptors with similar types first:
  - all color together, all texture together ...
  - then all visual together, all audio together ...
  - finally everything together
- In all cases the exact form of the fusion function has not much effect
  - linear combination of scores is quasi-optimal
  - fusion is very prone to over-fitting

Conceptual feedback

- Idea: using the probability(-like) scores predicted on the 346 concepts for building a new descriptor
- Comparable to the percepts or attribute-base approaches
- Classifiers trained on this concept score descriptor have lower performance than the original ones but:
- Late fusion between the original scores and the scores computed from classification on these original scores yield a small (1-2%) improvement.
- Baseline and partial experiment, could be improved.

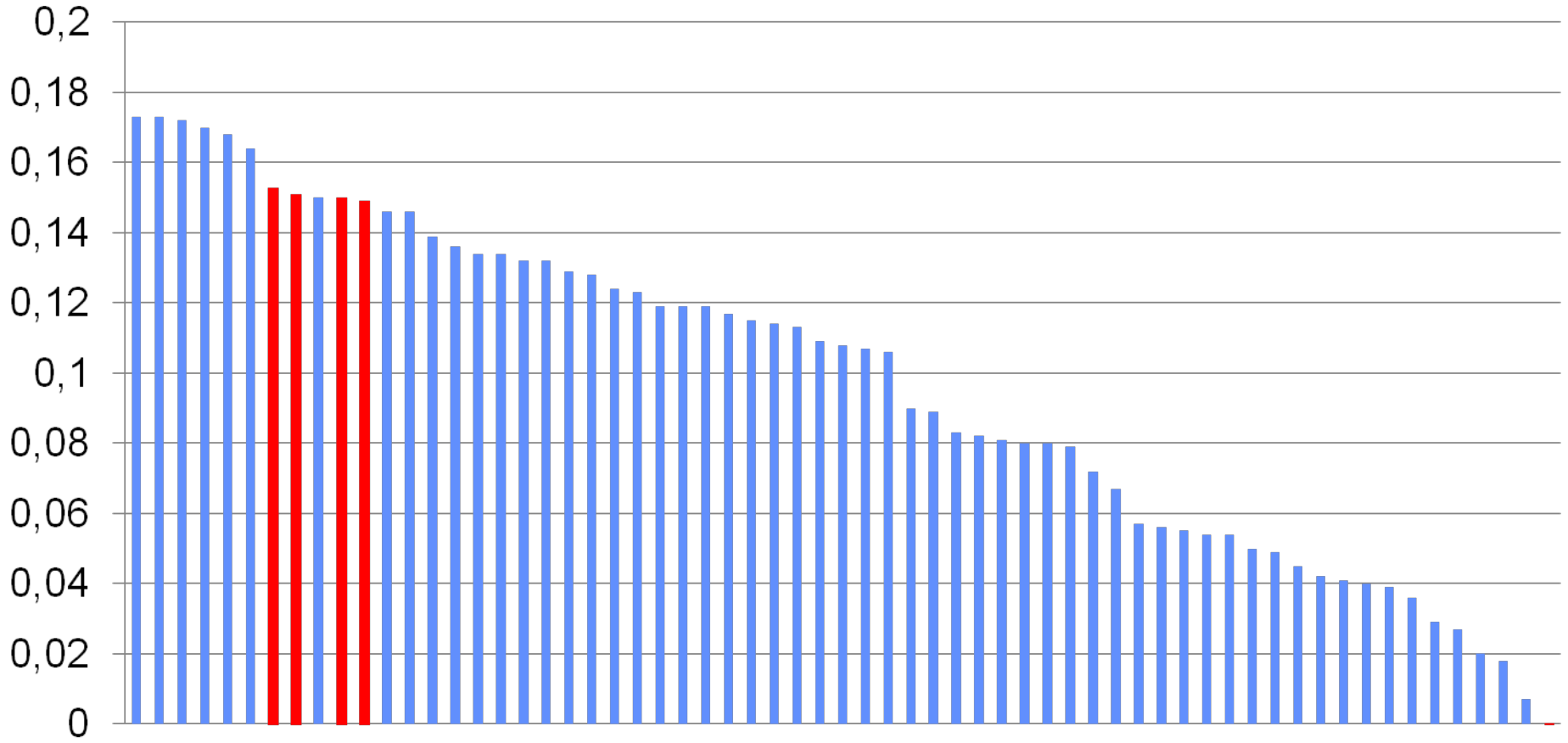
Temporal re-ranking

- Fact: shot within a video are semantically related, especially if they are close within the same video
- Idea: update shot scores according to neighbors' scores
- May be done globally (whole video) (Mérialdo 2009) or locally (window of a few shots) (Safadi 2010).
- Case of the full video:
  - Compute a global score for a whole video from the scores of all shots it contains (typically average or a variant)
  - Update the score of each shot using the global video shot (typically a linear combination or a variant)
  - Some parameters are tuned on a development set
- Gain from 5 to 15%
- Same effect if done lately, early or both, lately is simpler.

Submitted runs

- F_A_Quaero4_4: 0.1487
  - MAP weighted combination of all available descriptor/classifier combinations including the concept score feedback descriptor
- F_A_Quaero3_3: 0.1497
  - Flat and uniform combination of available descriptor/classifier combinations excluding the concept score feedback descriptor
- F_A_Quaero2_2: 0.1509 (+0.8%)
  - Optimized hierarchical combination of all available descriptor/classifier combinations excluding the concept score feedback descriptor
- F_A_Quaero1_1: 0.1528 (+1.3%)
  - Optimized hierarchical combination of all available descriptor/classifier combinations including the concept score feedback descriptor

Submitted runs



Conclusion

- Use of many descriptors, variants and classifiers
- Optimization of the descriptors
- Hierarchical fusion
- Conceptual feedback and temporal re-ranking
- Compute-intensive approach (no GPU optimization but use of the GRID'5000 facility)
- Many steps all bringing a modest improvement leads to a significantly improved global performance
- Complementary with approaches focusing on the best possible descriptor and the best possible machine learning?
- Multiple key frame was not used while a significant further improve can be expected (+12 to +15% reported by MediaMill)
- Audio was used (small contribution) but not ASR
- Improvements still possible