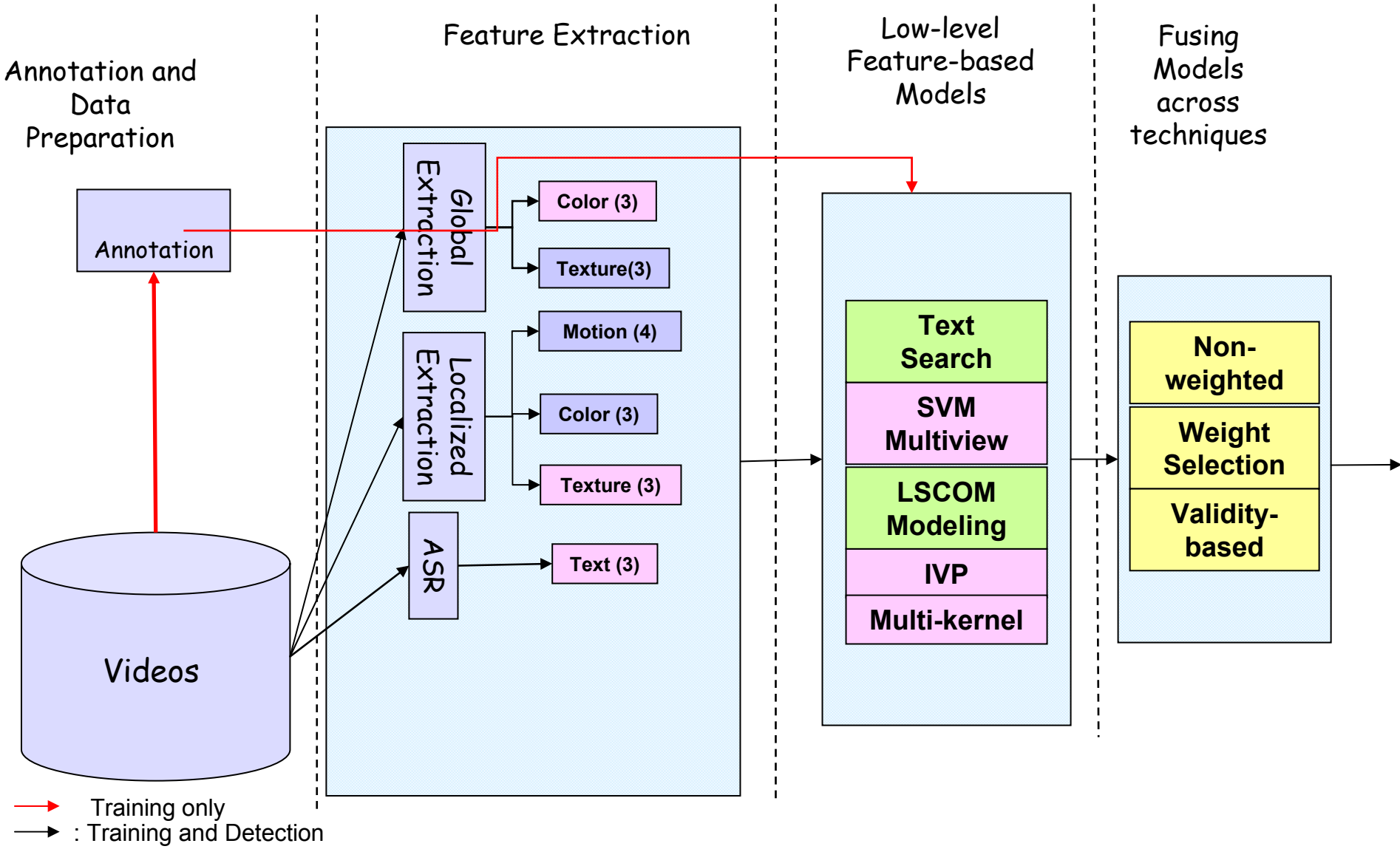
The IBM TRECVID Concept Detection System

Milind Naphade
Intelligent Information Analytics Group
IBM Thomas J Watson Research Center

Team: Milind Naphade, Dhiraj Joshi, Dipankar Datta,
Paul Natsev, Lexing Xie, Shahram Ebadoolahi, John Smith,
Alexander Haubold, Jelena Tesic, Joachim Seidl

The IBM TRECVID 2006 Concept Detection System



Feature Extraction

Visual

- Color Correlogram (166)
- Co-occurrence Texture (96)
- Color Moments (9)
- Wavelet Texture (12)
- Motion Magnitude & Direction (260)

Granularity

- Grid
- Global
- Compressed Domain

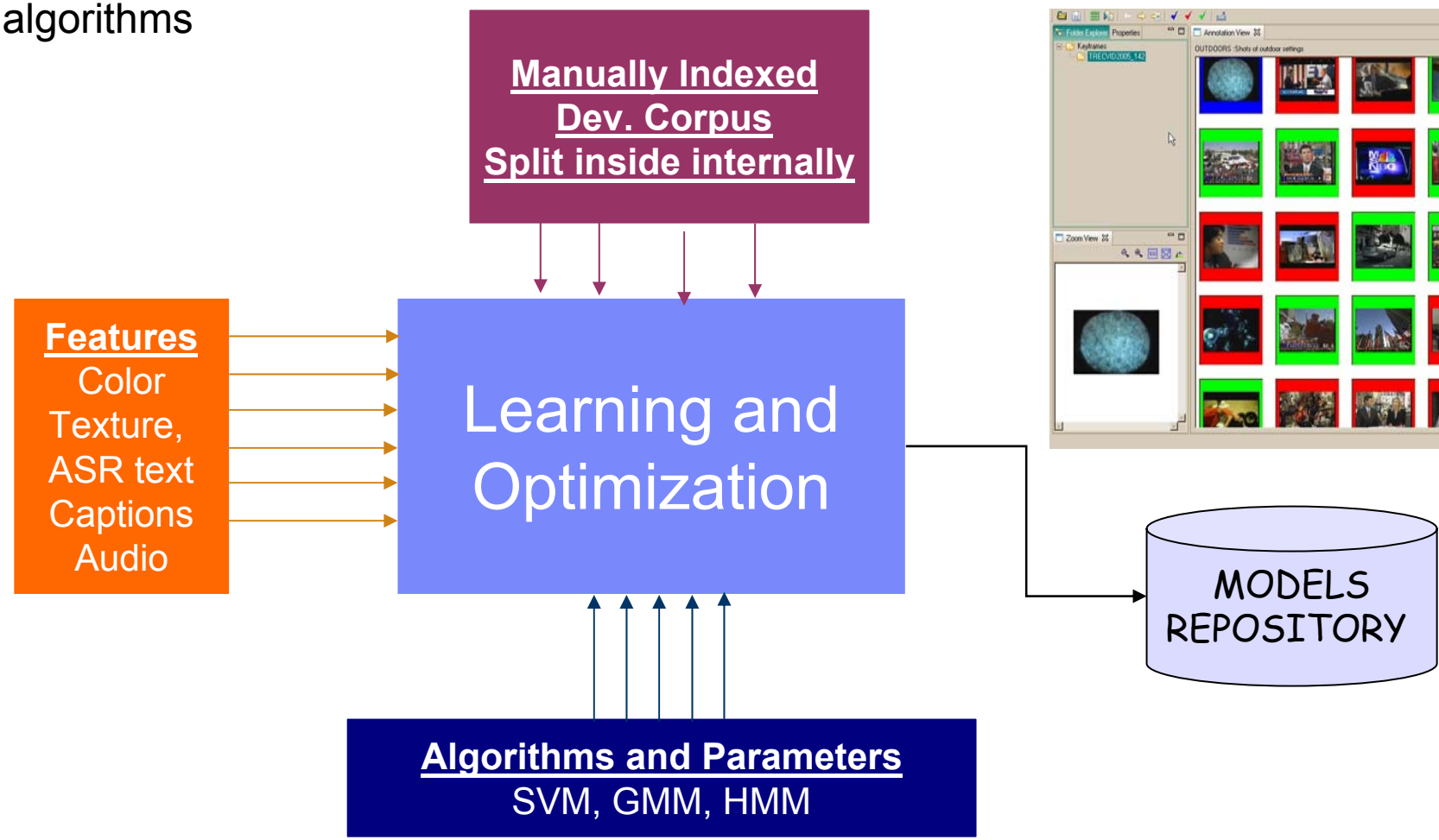
  Macro-block

ASR

- Text Search System

MARVEL MODELER

A tool for building models optimized over features, parameters and learning algorithms



Approach 1: Multiple Instantiations

- Consider multiple instantiations of learning problem
  - Different development corpus partitions
  - Different ground truth interpretations
  - Different learning algorithms
  - Different optimization schemes
- Fuse across the multiple instantiations using multiple normalization and simple fusion strategies

Reusing what we have – 2005 Models

- 2005 Models for 39 LSCOM-lite concepts using 5 visual features
- Run against 2006 data and combined using late fusion
- Development Corpus partitioned into 4 sets
- Uses SVM-light package and a range of gamma and C values for parameter optimization
- Uses the training set of 28055 images for training and validation set of 4400 images for validation and parameter optimization
- Uses a liberal interpretation of ground truth (annotation assumes positive when any annotator tags it positive) when multiple annotators inputs were available

Using Marvel :Modeler: 2006 Models

- 2006 Models for 39 LSCOM-lite concepts using 5 visual features
- Run against 2006 data and combined using late fusion
- Development Corpus partitioned into 3 sets
- Uses IBM implementation of SVM SMO and a range of gamma and C values for parameter optimization
- Uses the training set of 42000 images for training and validation
- Uses multiple interpretations of ground truth ranging from the most liberal to the most strict when multiple annotators inputs were available
- All new models built using Marvel Modeler using 7 parameter configurations for 5 features for each concept.
- Number of parameter configurations and features constrained by the time for the effort: 1 week

Multi-view Approach: Fusion

- Normalization
  1. Gaussian
  2. Sigmoid
  3. Range
  4. Rank
- Aggregation
  1. Average
  2. Weighed Average

Comparison between 2005 and 2006 SVM Models

- Older models built for TREC 2005
- Newer 2006 models built using Marvel Modeler
- Performance evaluated: 2005 Test Set
- Number of Concepts: 10
- Ground Truth: Provided by NIST
- MAP for 2005 models: 0.31
- MAP for 2006 models: 0.31
- MAP for fused 2005 and 2006 models: 0.37

20 % performance improvement fusing 2 views

Approach: Multi-kernel Learning

- Problem: Fusing multiple inputs: color moments, correlogram, texture …
- Late fusion
  1. Train SVM on each
  2. Perform weighted fusion on the prediction values

→ Equivalent to having kernel weights for each support vector

$$(1)\ y_j^* = \sum_i \eta_i K_j(\hat{x}, x_i)$$

$$(2)\ y* = \sum_j \mu_j y_j^*$$

$$= \sum_i \sum_j \eta_{ij} \mu_j K_j(\hat{x}, x_i)$$

- Alternative
  - Train one decision function for both the support vector weights and the kernel weights
  - … and make the support vector weights shared among kernels ?
- Advantages:
  - Decision + fusion learned in one pass
  - Less weights to learn and keep
  - Faster to evaluate on test data
  - … …

$$\hat{y} = \sum_j \sum_i \mu_j \eta_i K_j(\hat{x}, x_i)$$

Multiple Kernel Learning: Solution

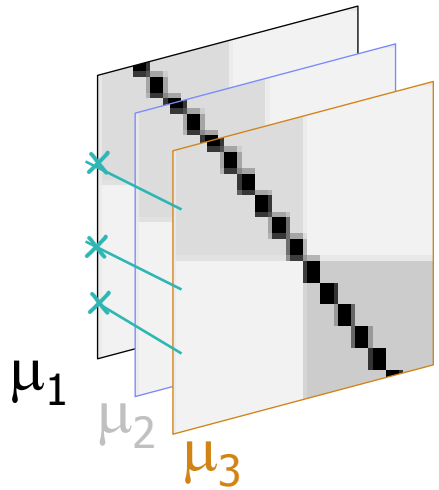
SVM

$$\hat{y} = \sum_i \eta_i K_j(\hat{x}, x_i)$$

support vectors

MKL

$$\hat{y} = \sum_j \sum_i \mu_j \eta_i K_j(\hat{x}, x_i)$$



Second-order cone programming

$$\min \quad \frac{\gamma^2}{2} - e^T\alpha$$

$$\text{s. t.} \quad \alpha^T D_y K_j D_y \alpha \leq \frac{\text{tr}(K_j)}{d}\gamma^2 \qquad j = 1, \ldots, k$$

[bach, lankriet, jordan2003]
[sedumi 2001]

Approach: Text Baseline

- IBM Text Search Engine for Shot-level ranking
  - JURU Search Engine used
  - No story level processing
  - Normalization of Text-based Run different than other runs
  - Fusion with visual models for generating multimodal runs
- Manual expansion from concepts to keywords
  - Potential use of LSCOM, CyC, WordNet to be explored
- Held Out Set Performance lower than Visual Models
  - Strength of approach is in combination hypothesis

Fusion Across Multiple Approaches

- Normalization
  1. Gaussian
  2. Sigmoid
  3. Range
  4. Rank
- Aggregation
  1. Average
  2. Weighed Average
- Weight Selection
  1. Validity-based

LSCOM Models

- Time limitation forced to build 70 LSCOM models
- Focused on frequent concepts that were also relevant
- Marvel Modeler leveraged for building models
- Same IBM colleague performed model building
- Context enforcement performed using manual mapping
- Few LSCOM-lite concepts targeted for context enforcement
  - Military Personnel
  - Waterscape
  - Airplane
- Resulted in 1 Type B Run mistakenly tagged Type A

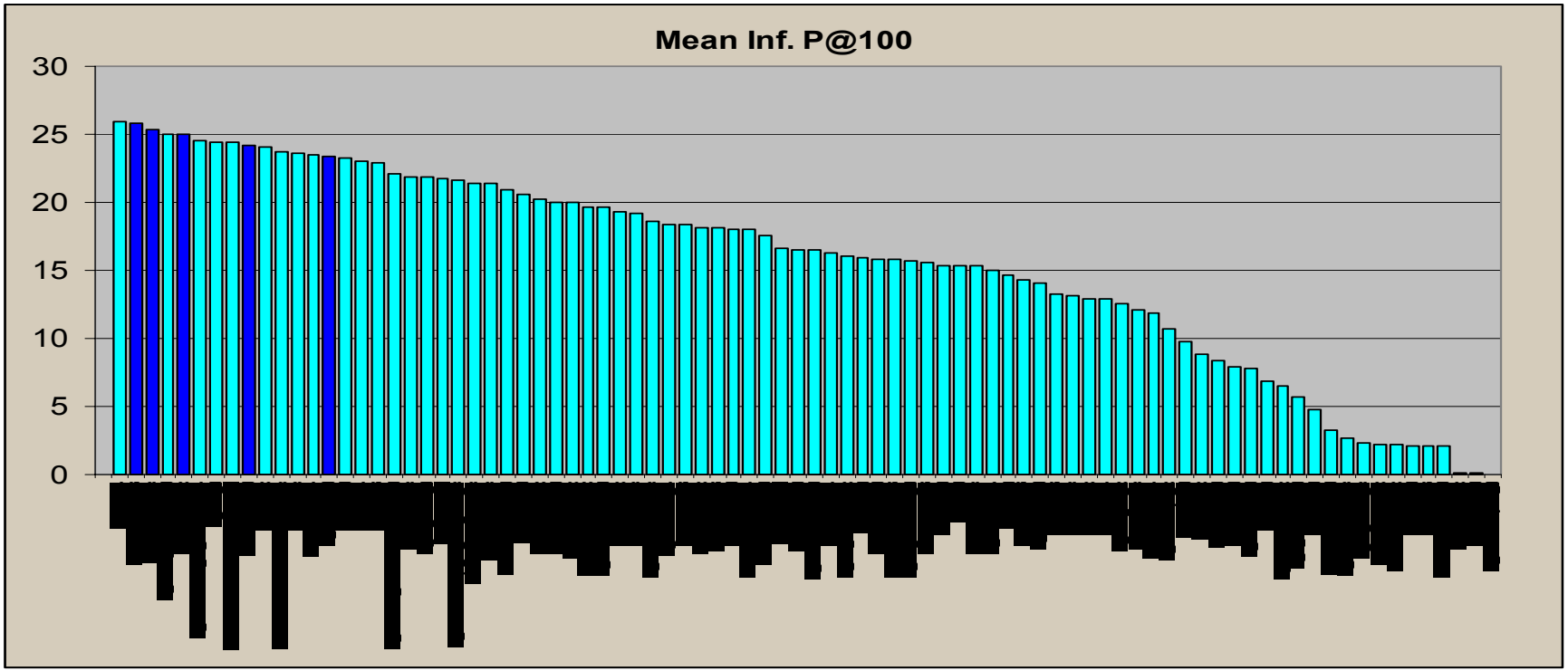
IBM Runs

Run Name	Type	Description
VB	A	Visual Baseline: Using 5 upto visual features and Multi-view SVM Models with naïve fusion
UB	A	Unimodal Baseline: Best of Visual Baseline and Text Baseline selected based on held out set performance
MBW	A	Fusion of Multi-view SVM Visual and Text Baselines
MBWN	A	Sigmoid Normalization and Decision Fusion of Multi-view SVM Visual and Text Baselines
MRF	A	Aggregating across all subsystems including Text Baseline, Visual Baseline Multi-kernel Linear machines, and Image Upsampling
MAAR	B	Aggregating across all subsystems including Text Baseline, Visual Baseline Multi-kernel Linear machines, Image Upsampling, and LSCOM context and using held out set for optimal selection

NIST Evaluation: Performance Summary

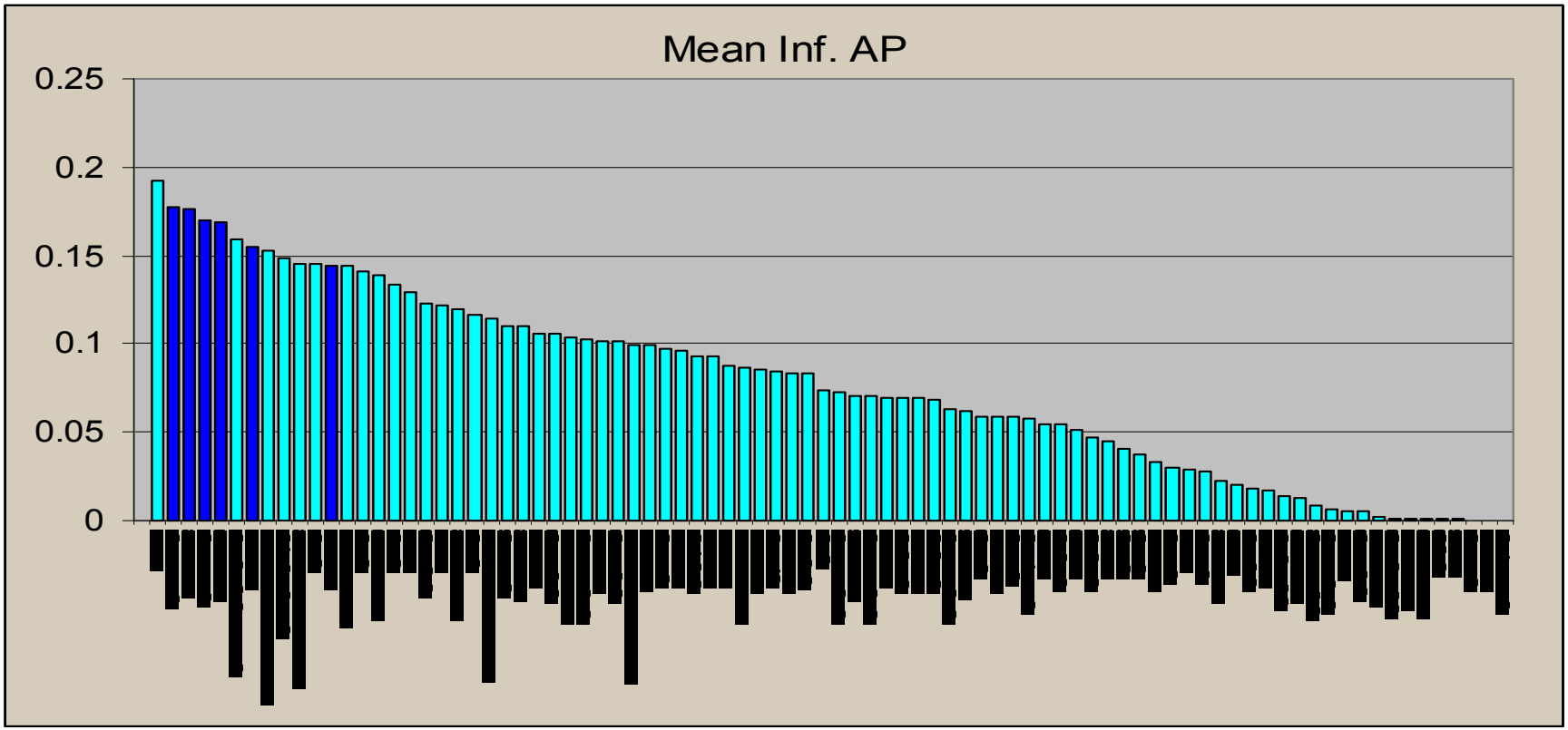
- All IBM runs except Visual Baseline buggy for 3 concepts
  - Submitted with Incorrect feature numbers (fnum)
  - Did not contribute to the pooling
- Mean Inferred Average Precision
  - Ranges from 0.145 (Visual only) to 0.1773 (Multimodal)
- NIST Returned Precision @100
  - Ranges from 22 (Visual Only) to 26 (Multimodal)
- Top performance for 7 of the 20 concepts
- Second highest MAP among all sites
- Top MIAP and IP@100 accounting for the bug
  - Excluding the 3 concepts that did not make it to the pool

IBM Runs in Context of Overall Benchmark



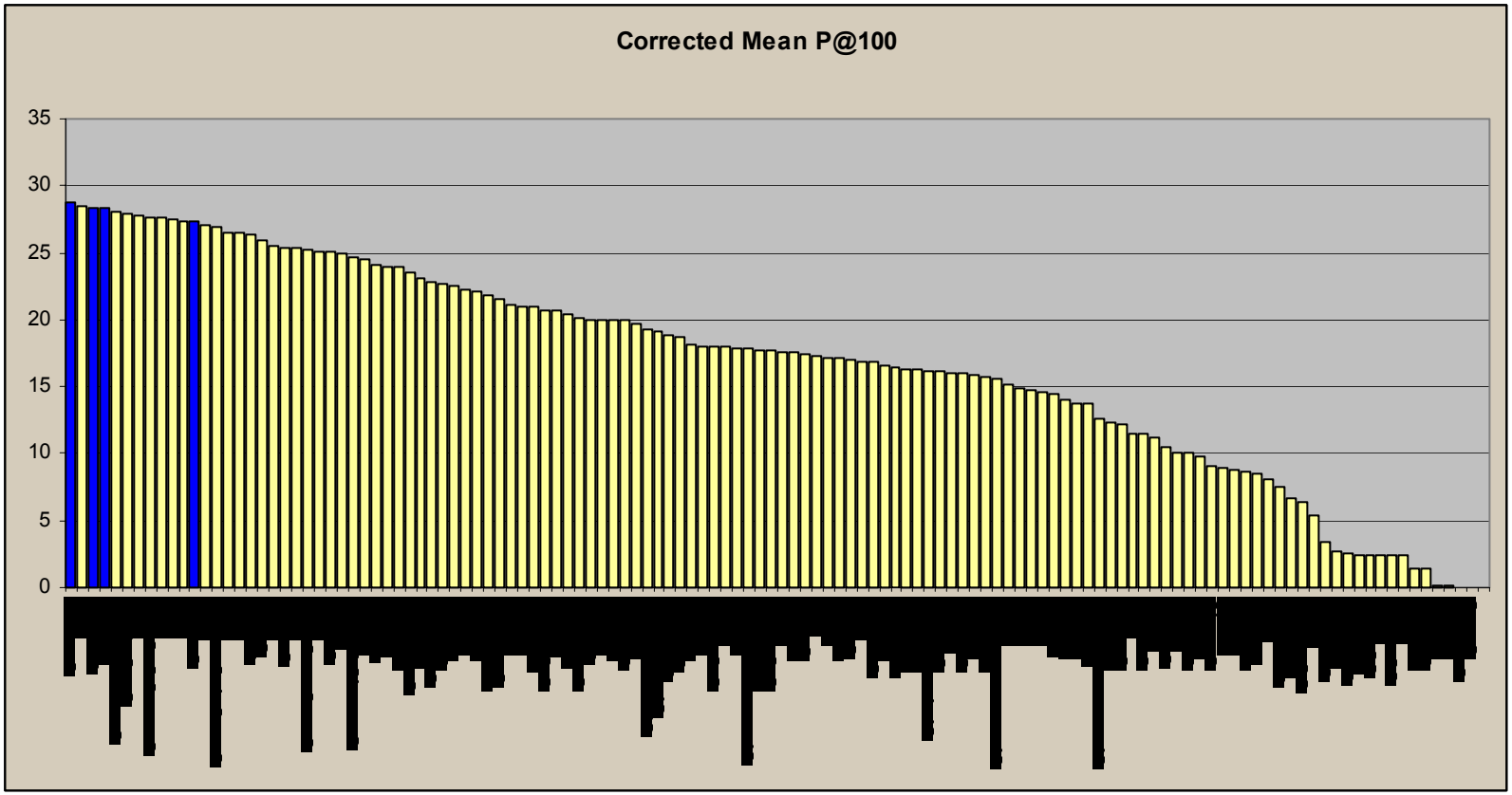
- IBM Runs returned near top performance with bug, top performance discounting bug
- NIST Returned P@100: Multimodal runs improve over Visual baseline by 10 %
- InfAP: Multimodal Runs improve over Visual baseline by 22 %
- IBM Runs have top performance for 7/20 concepts

IBM Runs in Context with Overall Benchmark



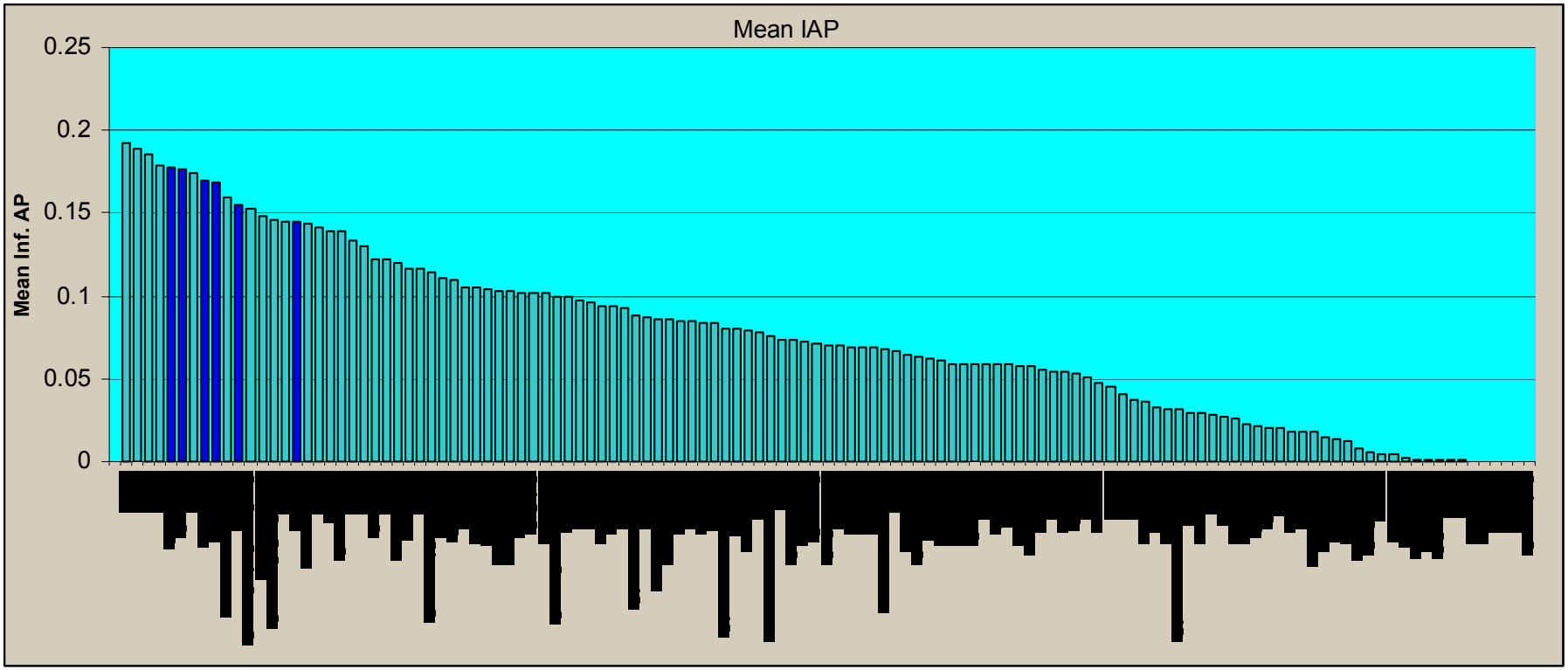
- IBM Runs returned near top performance with bug, top performance discounting bug
- NIST Returned P@100: Multimodal runs improve over Visual baseline by 10 %
- InfAP: Multimodal Runs improve over Visual baseline by 22 %
- IBM Runs have top performance for 7/20 concepts

IBM Runs in Context with Overall Benchmark



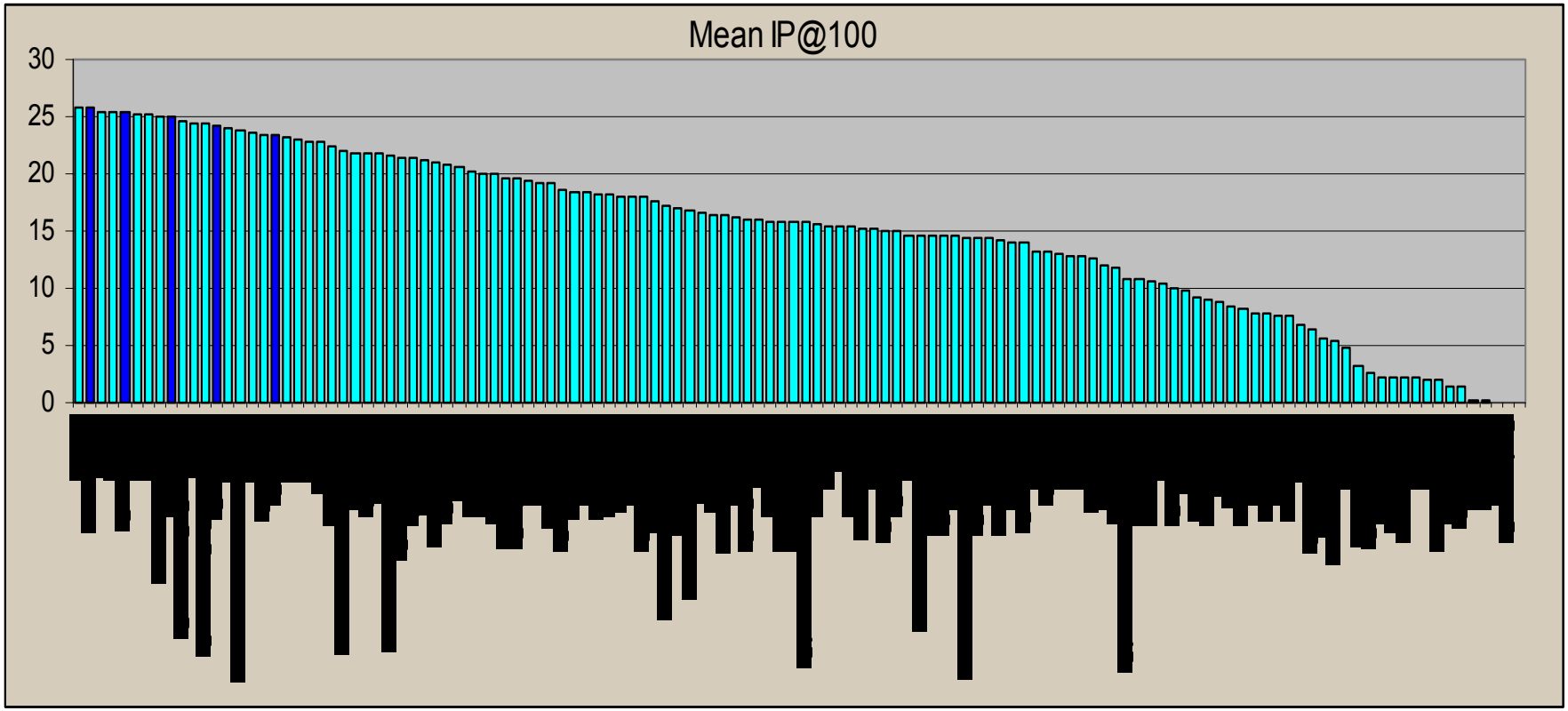
- IBM Runs returned near top performance with bug, top performance discounting bug
- NIST returned P@100: Multimodal runs improve over Visual baseline by 10 %
- InfAP: Multimodal Runs improve over Visual baseline by 22 %
- IBM Runs have top performance for 7/20 concepts

IBM Runs in Context with Overall Benchmark



- IBM Runs returned near top performance
- NIST returned P@100: Multimodal runs improve over Visual baseline by 10 %
- InfAP: Multimodal Runs improve over Visual baseline by 22 %
- IBM Runs have top performance for 7/20 concepts

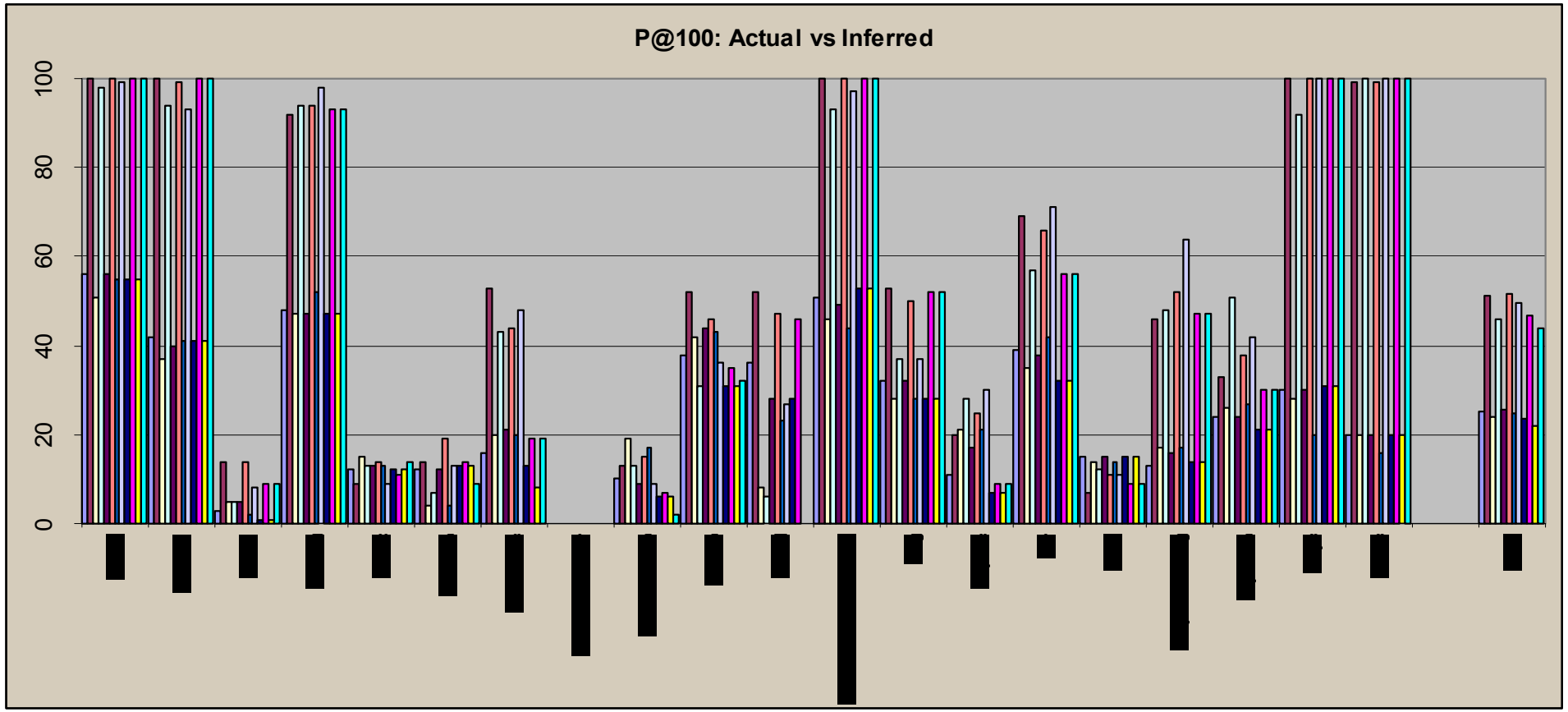
IBM Runs in Context with Overall Benchmark



- IBM Runs returned near top performance
- NIST returned P@100: Multimodal runs improve over Visual baseline by 10 %
- InfAP: Multimodal Runs improve over Visual baseline by 22 %
- IBM Runs have top performance for 7/20 concepts

But Was this Analysis Conclusive?

- Random Sampling of the Pool raises questions about conclusiveness
- Actual P@100 Range: 44 to 52
- NIST Returned P@100 Range: 22 to 26
- Absolute Numbers Matter: So Relative Ordering may not be enough
- Performance discrepancy significant for 15 of the 20 concepts

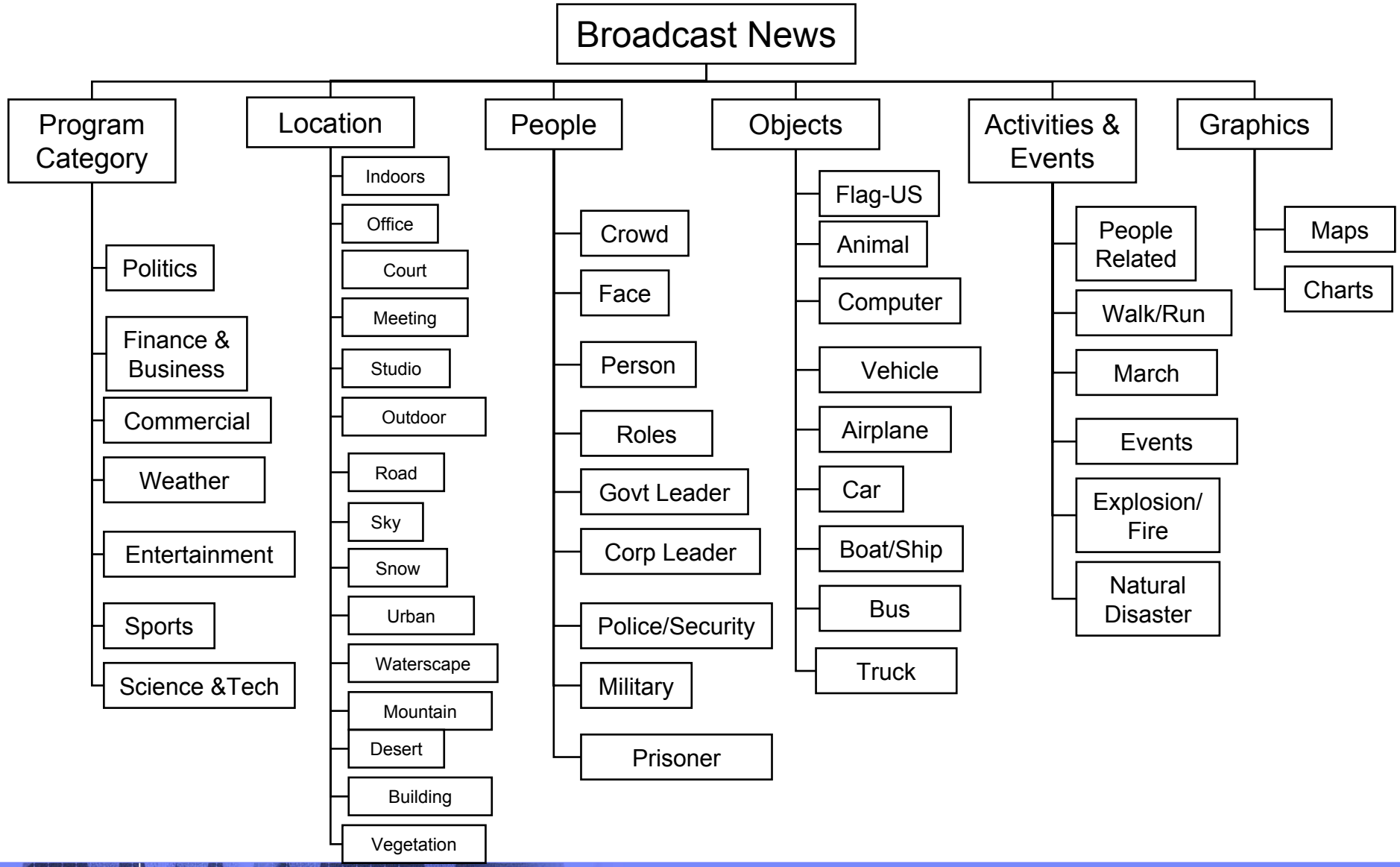


Observations

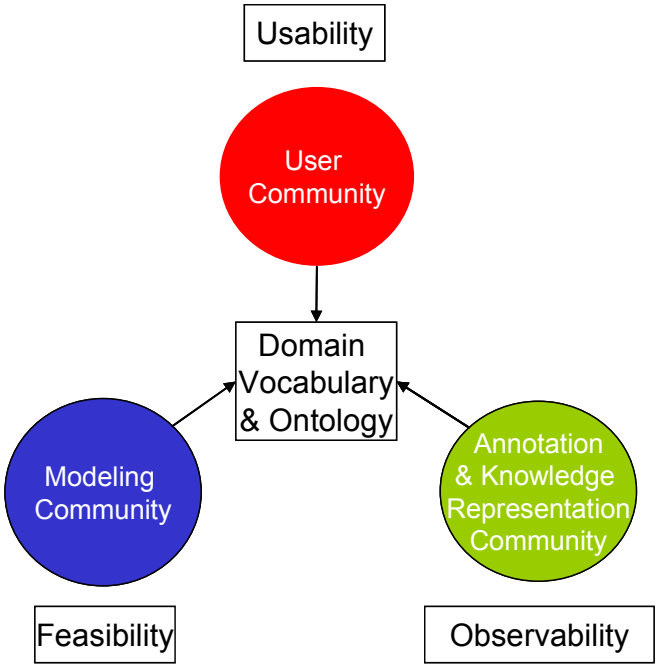
- Visual Baseline created by leveraging Marvel Modeler Asset
- Text+Visual improve performance by 10 % over Visual-only
- Context helps when underlying contributors are robust
- Need more work on event and object detection
- Normalization & multimodal fusion leads to re-ranking

  Significant improvement in concepts such as Airplane (3x better)
- LSCOM provides large untapped potential
  - **Quality is Key**
  - **Once Acceptable Quality guaranteed, Quantity is game changer**

From LSCOM-lite to LSCOM

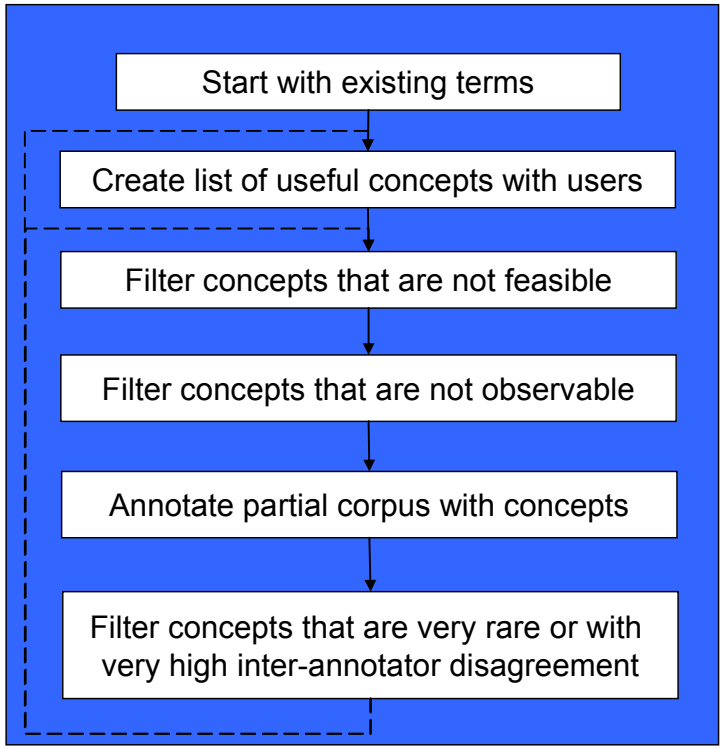


Goal and Vision



LSCOM

Workflow



Deliverables

- 1000+ concept lexicon
- Annotated corpus
- 39 Use Cases and 250 + Queries
- Ontology
- Experimental Evaluation

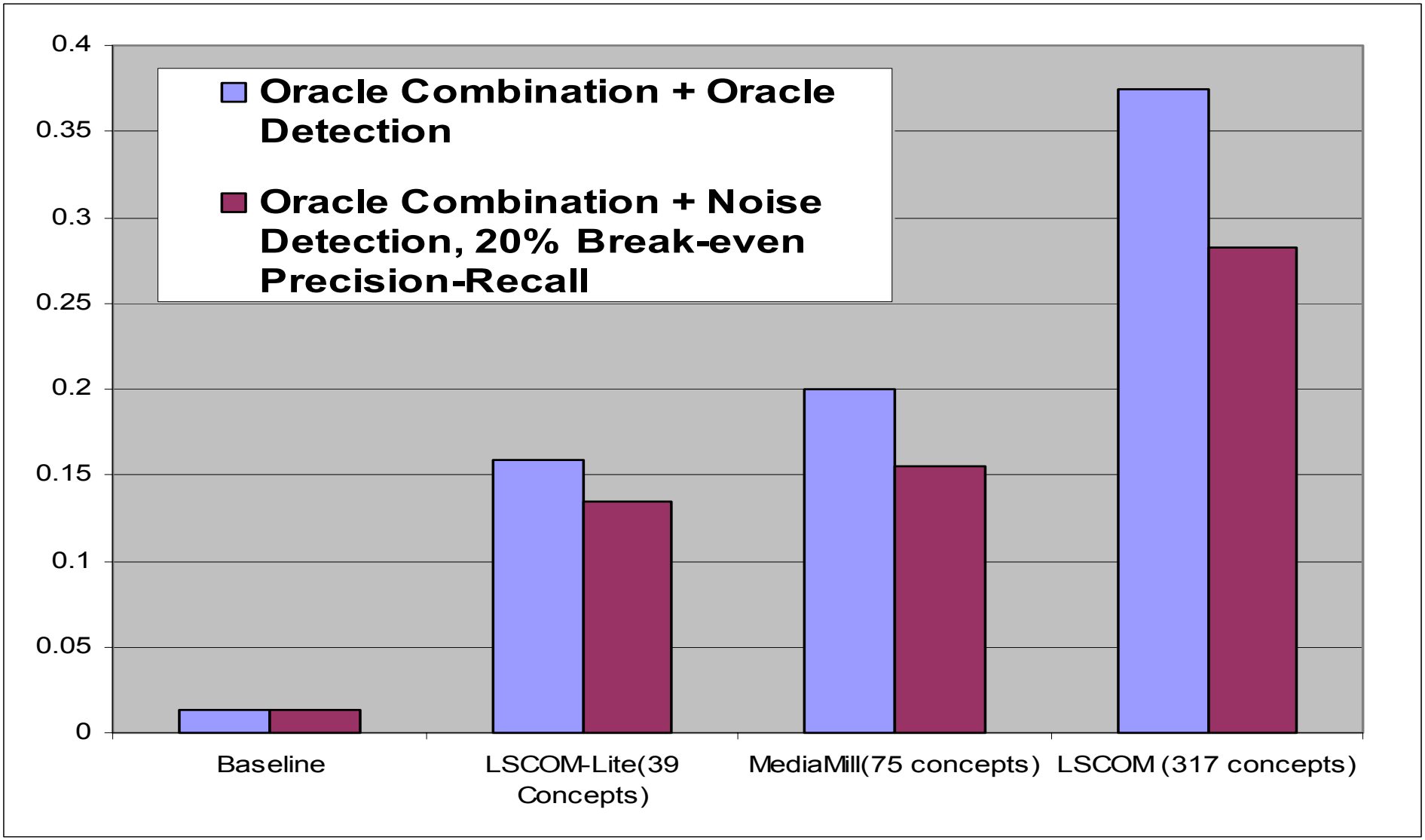
Impact

- Largest annotated video corpus
- Leveraged at TRECVID and other fora
- LSCOM mapped into openCyC and ResearchCyC
- Dissemination at various fora for optimizing utilization leading to collaboration opportunities

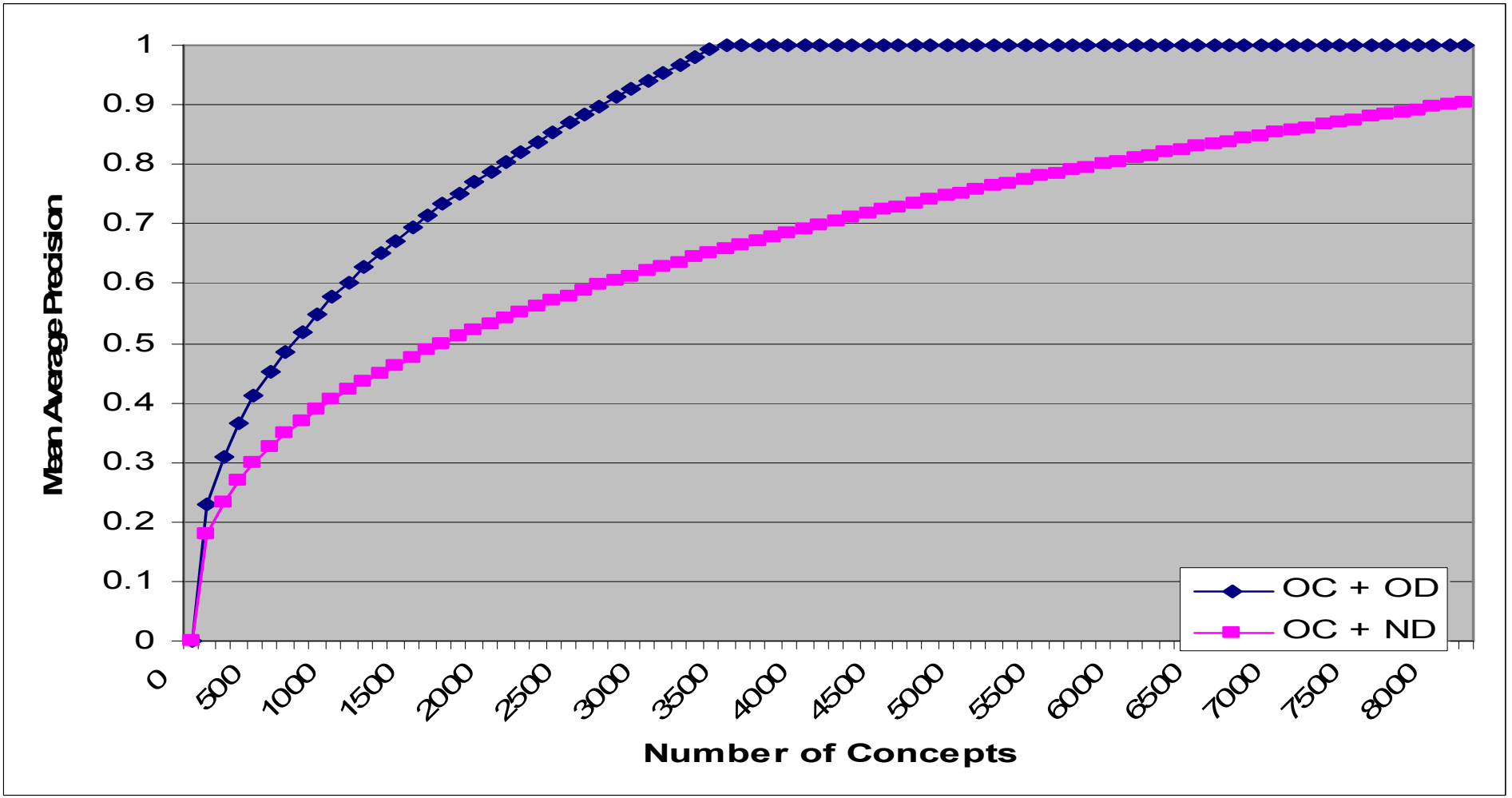
What is LSCOM?

- 1000+ concepts that describe broadcast news from the intelligence analyst perspective
- An annotated corpus of 61901 shots (80 hours) of broadcast news video (3 languages, 6 channels) for 449 concepts
- Compilation of 39 use cases and 250+ TRECVID style Queries that represent analyst requirements
- Mapping of LSCOM concepts and subsequent expansion using CyC (packaged in OpenCyC and ResearchCyC releases)
- Initial results on modeling 300 of the annotated concepts

Evaluation Results



Extrapolating MAP by # concepts:



How many concepts do we need? 3K-5K