

Joint Visual-Text Modeling for Multimedia Retrieval

Presented by: *Giri Iyengar*

JHU CLSP Summer Workshop 2004

Team Update, August 4th 2004

Team

- Undergraduate Students
 - Desislava Petkova (Mt. Holyoke), Matthew Krause (Georgetown)
- Graduate Students
 - Shaolei Feng (U. Mass), Brock Pytlik(JHU), Pavel Ircing(U. West Bohemia), Paola Virga (JHU)
- Senior Researchers
 - Pinar Duygulu, Bilkent U., Turkey
 - Giri Iyengar, IBM Research
 - Sanjeev Khudanpur, CLSP, JHU
 - Dietrich Klakow, Uni. Saarland
 - R. Manmatha, CIIR, U. Mass Amherst



Outline

- TRECVID description
- Our retrieval models
- Models for Image Annotation
- Models for Auto-Illustration
- Next Steps

TRECVID Corpus and Task

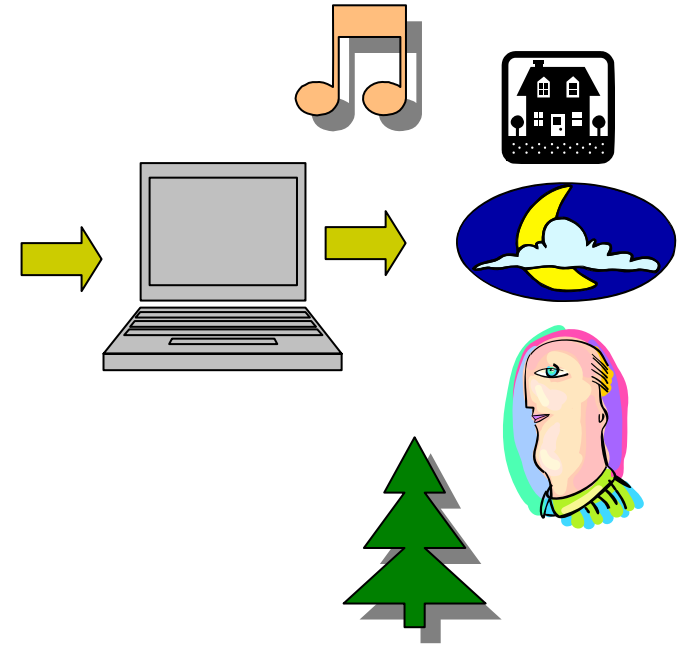
□ Corpus

- Broadcast news videos used for Hub4 evaluations

□ Tasks

- Shot-boundary detection
- News Story segmentation (multimodal)
- Concept detection ("Annotation")
- Search task

Concept Detection Task



- NIST provides list of benchmark concepts
- Sites build models & NIST evaluates

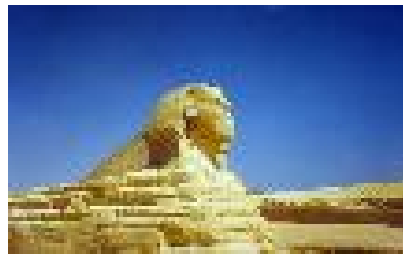
Search Task

- Find video clips ("shots") *visually* relevant to a multimedia statement of information need



Find shots containing
the Mercedes Logo (star)

Find shots with
the Sphinx





Concept Detection vs. Search

- The essential difference is
 - When are the topics made available?
 - How much training data is available?



Alternate (development) Corpus

- COREL photograph database
 - 5000 high-quality photographs with captions
- Task
 - Annotation

Outline

- TRECVID description
- Our retrieval models
 - Linear
 - Log-Linear
- Models for Image Annotation
- Models for Auto-Illustration
- Next Steps

Retrieval Model I: $p(q|d)$

$$p(q_w, q_v | d_w, d_v)$$
$$= p(q_w | d_w, d_v) \times p(q_v | d_w, d_v)$$

Baseline. Standard text-retrieval

$$p(q_w | d_w, d_v)$$
$$= \lambda_w p(q_w | d_w) + (1 - \lambda_w) p(q_w | d_v)$$

Retrieval Model I: $p(q|d)$

$$p(q_w, q_v | d_w, d_v) =$$
$$[\lambda_w p(q_w | d_w) + (1 - \lambda_w) p(q_w | d_v)] \times$$
$$[\lambda_v p(q_v | d_w) + (1 - \lambda_v) p(q_v | d_v)]$$

Retrieval Model II: $p(q|d)$

- We want to estimate $p(q_w, q_v, d_w, d_v)$
- Assume we can only estimate pairwise marginals reliably. E.g.,

$$\sum_{d_w, q_w} p(q_w, q_v, d_w, d_v) = p(d_v, q_v)$$

- Setting this as a Maximum Entropy problem with 6 constraints, after 1 iteration of GIS we get (Log-Linear Model)

$$p(q_w, q_v | d_w, d_v) \propto p(q_w | d_w)^{\lambda_1} p(q_w | d_v)^{\lambda_2} p(q_v | d_w)^{\lambda_3} p(q_v | d_v)^{\lambda_4}$$

Outline

- TRECVID description
- Our retrieval models
- Models for Image Annotation-- $p(q_w | d_v)$
 - Relevance Models
 - Machine Translation Models
 - Graphical Models
- Models for Auto-Illustration-- $p(q_v | d_w)$
- Next Steps

Component $p(q_w | d_v)$: Image Annotation

$$p(q_w | d_v) = \sum_c p(q_w | c) p(c | d_v)$$

- Build models for $p(c | d_v)$ using Relevance Models, MT & HMMs
- Given textual query, get a distribution over a concept vocabulary ($p(q_w | c)$)
 - E.g Run text retrieval over training corpus and build this model from top R documents
 - Baseline: Rule-based system

Relevance Models

- Analogous to Cross Lingual IR
 - Retrieve documents in a language (images) with queries in a different language (English text)
- Cross Media Relevance Model (CMRM)
 - Discrete model - Visual vocabulary discretized.

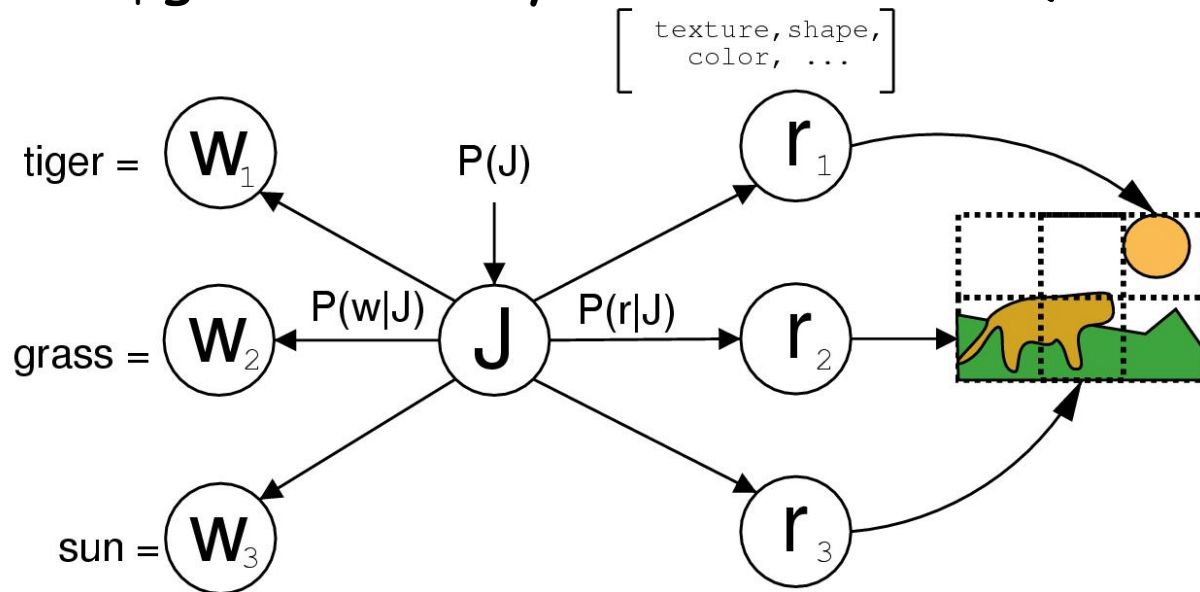
$$P(c, \vec{v}) = \sum_J P(J) P(c | J) \prod_{v \in \vec{v}} P(v | J)$$

- Continuous Relevance Model.
 - $P(r|J)$ is estimated using a kernel density function.

$$P(c, \mathbf{r}) = \sum_J P(J) P(c | J) \prod_{i=1}^m P(r_i | J)$$

Continuous Relevance Model

- A generative model
- Concept words w_j generated by an i.i.d. sample from a multinomial
- Features r_i generated by a multi-variate (Gaussian) de



Normalized Continuous Relevance Models

- Normalized CRM
 - Pad annotations to fixed length. Then use the CRM.
 - Similar to using a Bernoulli model (rather than a multinomial for words).
 - Accounts for length (similar to length of document in text retrieval).

Extensions









- New model.
$$P(c, \vec{r}) = \sum_J P(J) P(c | J) \prod_{r \in \vec{r}} P(r | w, J)$$
- How does one estimate $P(r | w, J)$?
- Discrete case
 - Maybe probabilities from IBM translation model 1?
 - Or estimate using images with word w as annotation.
- Continuous Case
 - A little trickier. Not clear yet.

Extensions

- Information Retrieval
 - Query based stuff usually works better
 - Pseudo-Relevance feedback, query expansion
 - Run an initial (text) ASR retrieval
 - Create text/image models with the returned top ranked documents
 - Similarly, run an initial image retrieval
 - Create text/image models with the returned top ranked documents

Retrieval example

Query: Bill_Clinton

CRM				
Norm alized -CRM				

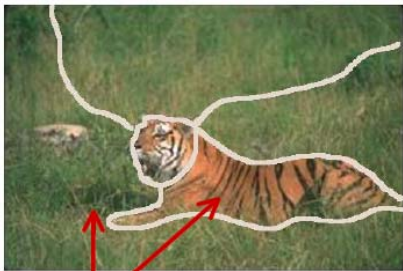
Retrieval Results on Video data

Query length	1 word	2 words	3 words
Number of queries	107	431	402
Relevant images	6649	12553	11023
Precision at 5 retrieved key frames			
CRM	0.36	0.33	0.42
normalized CRM	0.49	0.47	0.58
Mean Average Precision			
CRM	0.26	0.19	0.25
normalized CRM	0.30	0.26	0.32

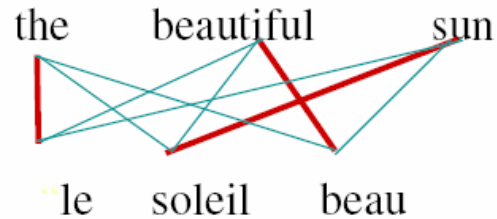
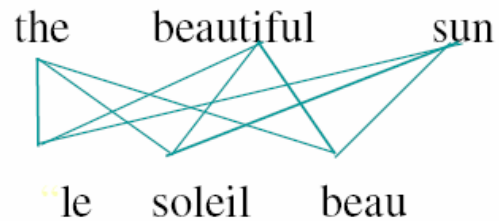
Data: 6 hrs of TRECVID03 – 5200 keyframes.

Training – 3470 keyframes. Test – 1730 keyframes.

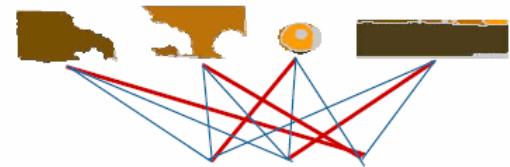
Machine Translation Approach



?
tiger grass cat



"sun sea sky"



"sun sea sky"

$$\Pr(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} \Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) \quad \Pr(\mathbf{w}|\mathbf{v}) = \sum_{\mathbf{a}} \Pr(\mathbf{w}, \mathbf{a}|\mathbf{v})$$

Annotation Results (Corel set)



field foals horses mare
tree horses foals mare field



birds grass plane zebra
grass tusks water plane
zebra



flowers leaf petals stems
flowers leaf petals grass
tulip



bear polar snow
beach polar grass bear snow



mountain sky snow water
sky mountain water
clouds snow



people pool swimmers water
swimmers pool people water sky



jet plane sky
sky plane jet tree clouds



people sand sky water
sky water beach people hills

Using word co-occurrences

- Model 1 with word co-occurrence

$$p_1(c_2 | J) = \sum_{c_1} p_0(c_1 | J) p(c_2 | c_1)$$

- Normalization

$$p_2(c | J) \sim \log(N/k) p_1(c | J)$$

N : number of images in the training data

k : number of times concept c appears in training data

Evaluating annotation performance

Predict 5 words with the highest frequency, and compare with actual annotations



Actual keywords

grass tige cat



Predicted words

cat horse grass water forest

Prediction performance = $1 / N \sum_I (\# \text{correct in } I) / (\# \text{ actual annotations in } I)$

Recall(c) = (number of times c is correctly predicted) / (number of times c appears as an annotation word)

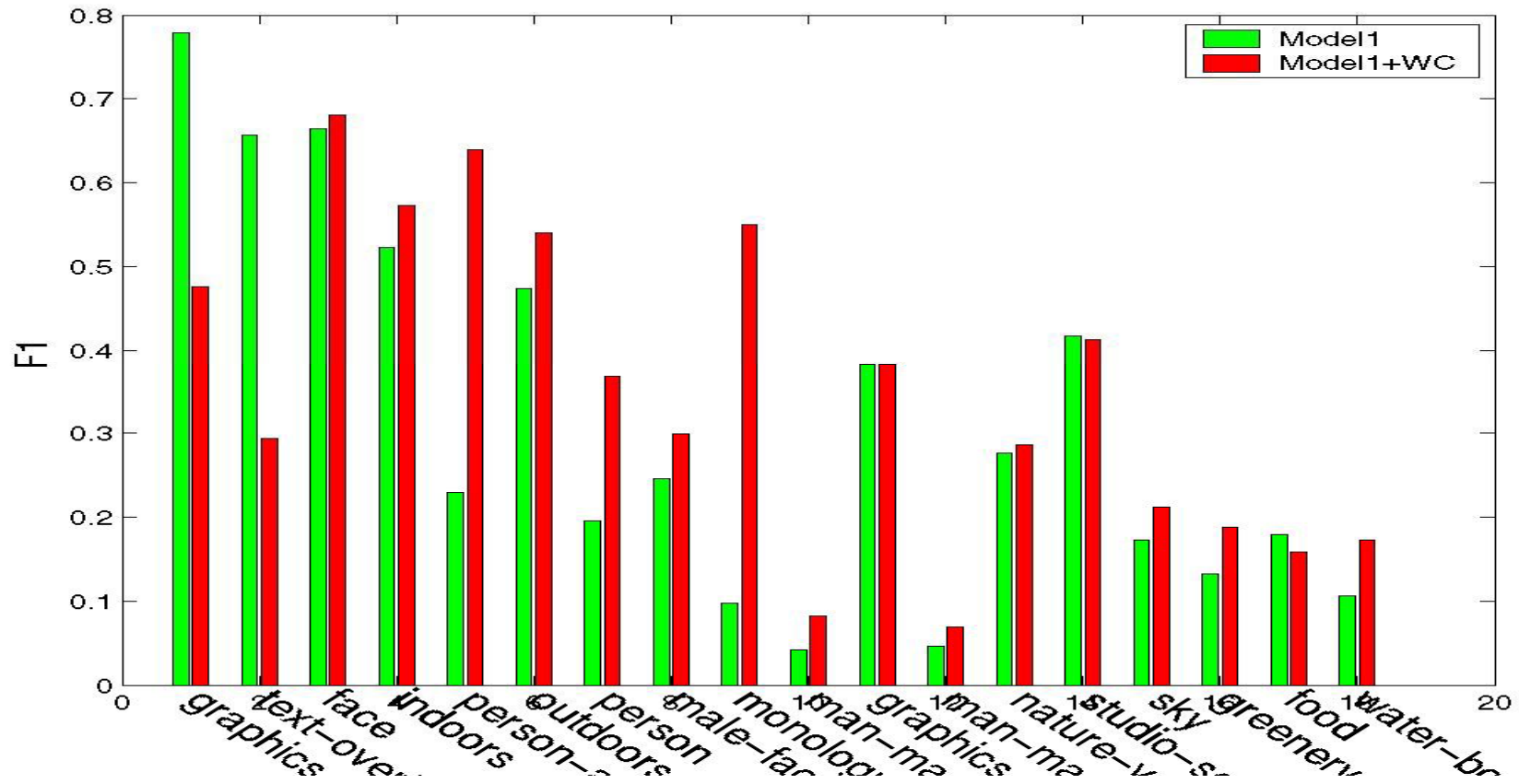
Precision(c) = (number of times c is correctly predicted) / (number of times c is predicted)

Results - Preliminary TRECVID

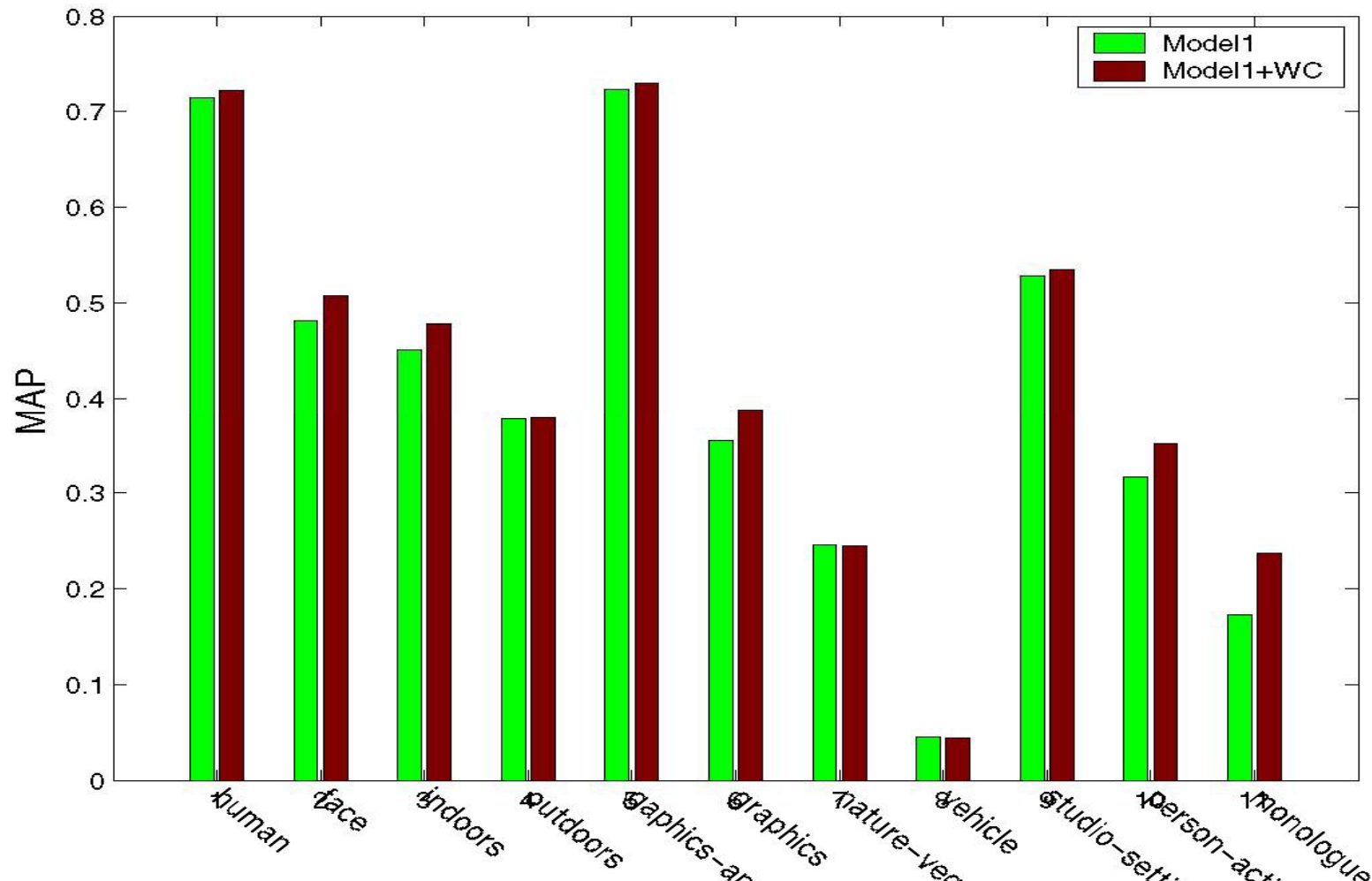
Model	Prediction performance	Recall	Precision	Num predicted	MAP
Model1	0.488	0.0540 (0.272)	0.0666 (0.336)	23	0.088 (0.309)
Model1 + WC	0.229	0.0533 (0.326)	0.0573 (0.350)	19	0.096 (0.303)

Comparison on annotation performance

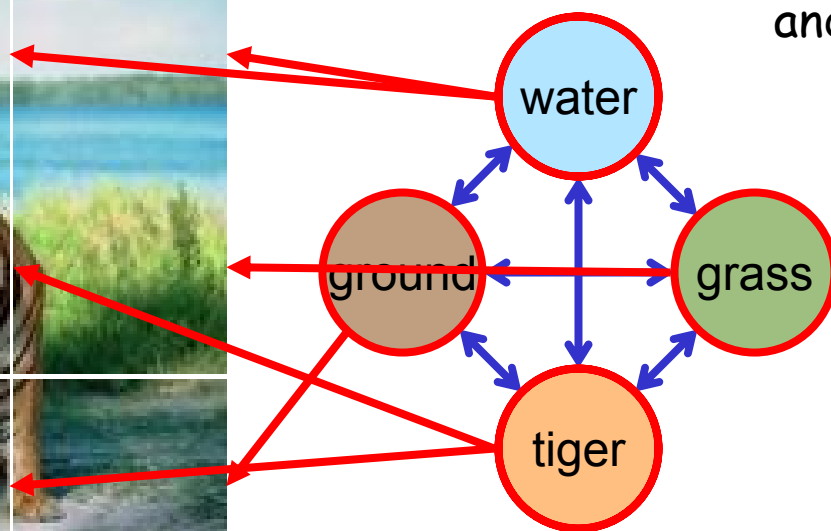
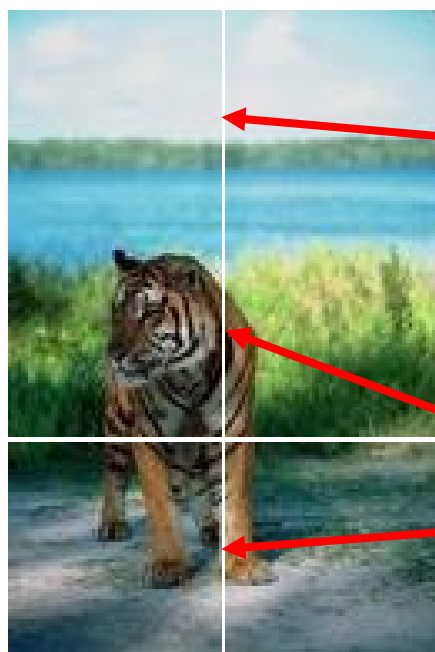
$$F1 = (\text{recall} + \text{precision}) / 2$$



Comparison on retrieval performance



HMMs for Image Annotation



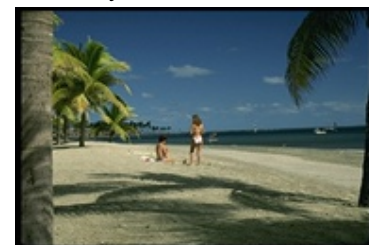
- Image blocks are “generated” by its caption-words
- Alignment between image blocks and captions is a **hidden variable**
 - Train via the EM algorithm
- **Training HMMs**: 3-5 known states, given by the caption.
 - 4500 training images
 - 374 word caption-vocab
 - 1 pdf/pmf per state

Test HMM has 374 states, with $p(w'|w)$ estimated by co-occurrence LM

Posterior probability from forward-backward pass used for $p(w|\text{Image})$

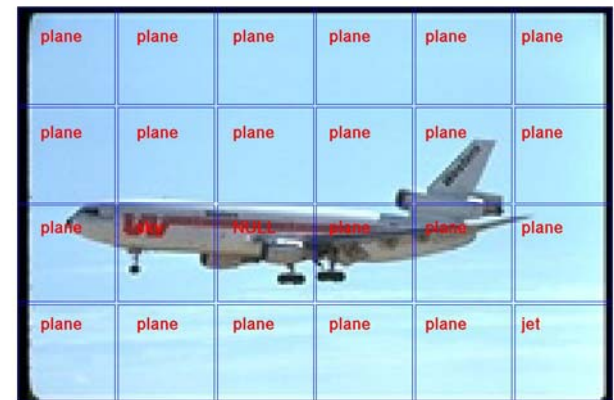
Challenges in HMM training

- There is **no** notion of **order** in the caption words.
- **No linear order** in image blocks (assume raster-scan)
 - Additional spatial dependence between block-labels is missed
 - Partially addressed via a more complex DBN (see next)
- Unsuitable (incomplete) human annotations
 - Annotators often mark only **interesting** objects, leaving large portions of the image unlabelled ...
 - in spite of having such objects (sky) in the caption vocabulary
- This may be alleviated by inserting an optional “unlabelled background” state during training



Accounting for Unlabelled Background (Gradual Training)

- Identify a set of "background" words (sky, grass, water,...)
- In the initial stages of HMM training
 - allow only "background" states to have their individual emission probability distributions
 - All other objects share a single "foreground" distribution
- Run several EM iterations
- Untie the foreground distribution and run more EM iterations
- Improved alignment of training images
- Retrieval performance on test images not so good ... work in progress



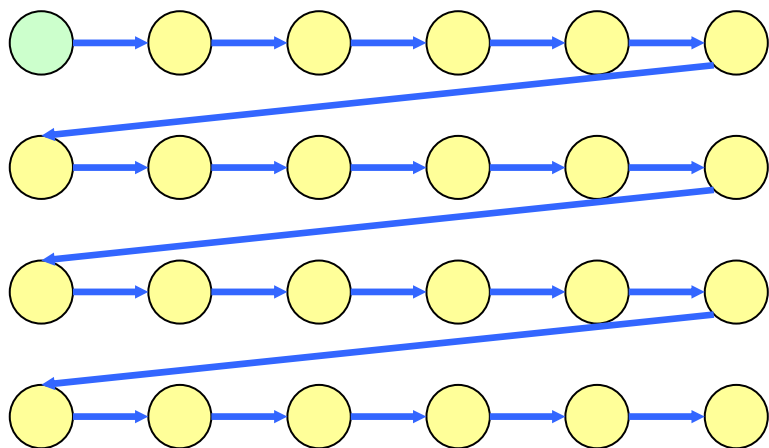
With Gradual Training

Results - HMM - Corel data

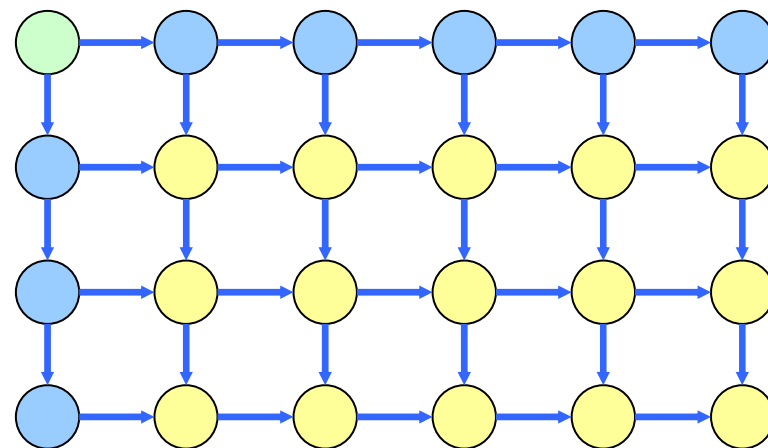
Model	MAP	Prediction performance	Recall	Precision
Discrete - w/o LM	0.1199	0.1023	0.1006 (0.3793)	0.0900 (0.3394)
Discrete - with LM	0.1501	0.3268	0.1646 (0.4774)	0.0991 (0.2875)
Continuous - w/o LM	0.1100	0.1017	0.1085 (0.4036)	0.0841 (0.3127)
Continuous - with LM	0.1231	0.2103	0.1320 (0.3787)	0.0954 (0.2738)

Modeling Spatial Dependency

- Hidden states have spatial dependence -- beyond the left-neighbors modeled by a raster-scan HMM
- Construct a graphical model, with as many hidden states as image-blocks, and explicitly model neighborhood dependencies



An HMM for a 24-block Image



A DBN for a 24-block Image

GMTK Implementation of DBN

- Each hidden variable depends on 2 neighbors
 - Leads to larger “transition probability” tables
 - Data sparseness in training
 - May be alleviated by tying various probabilities
 - Leads to a larger configuration (state) space
 - Running times go up during decoding
- Work in progress ...
 - Struggling with GMTK issues

Annotation using Cross-lingual IR


- Treat Image Annotation as a Cross-lingual IR problem (like Relevance Models)
 - Image comprising visterms (target language) and a query comprising concepts (source language)

$$P(c | d_V) = \lambda \left(\sum_{v \in d_V} p(v | B_V) p(c | v) \right) + (1 - \lambda) p(c | B_C)$$

Cross-Lingual IR using Lemur

What is needed?

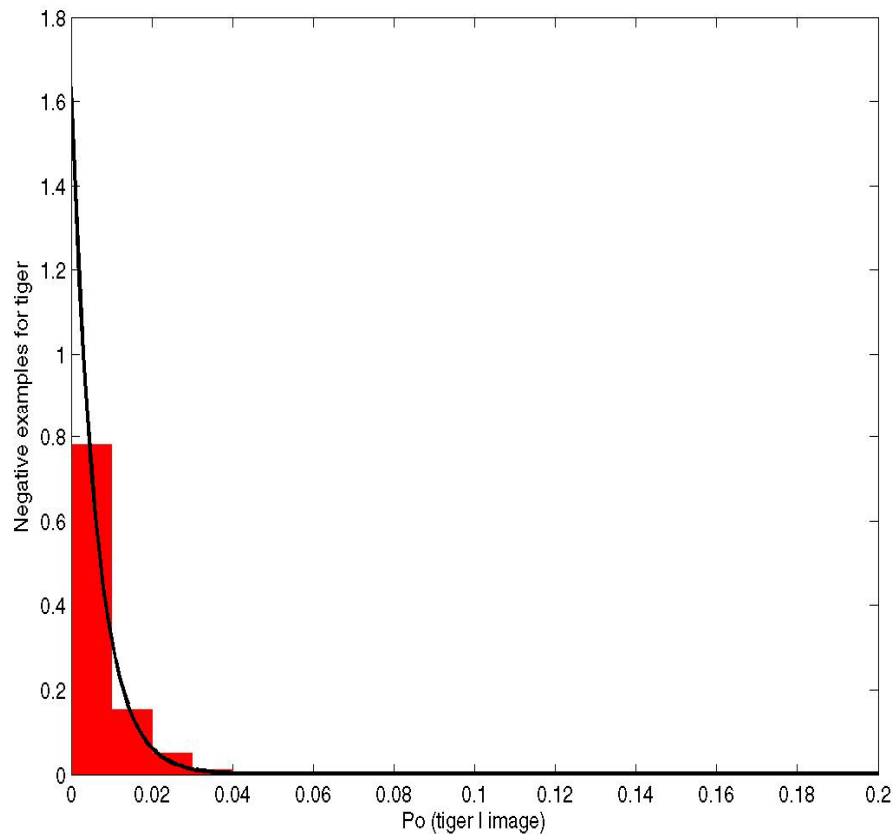
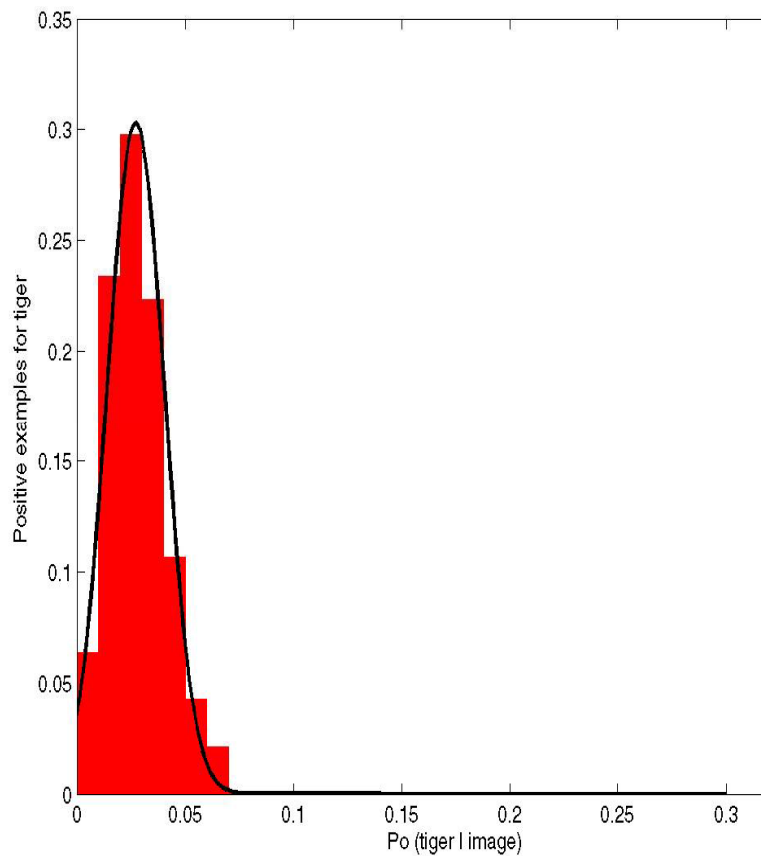
- Source Documents (concepts)
 - 26608 Documents
 - Vocabulary size 75
- Target Documents (visterms)
 - 26608 Documents
 - Vocabulary size 2981
- Dictionary:
 - Giza++
 - $p(\text{concept}|\text{visual})$
- Queries: all concepts
 - 75 query concepts
- Work in progress

$$p(\text{tiger} | \text{img}) = 0.7$$


Score distribution modeling

- Each model gives a "score" for a concept
- Obtain a histogram of scores when the concept is present (absent) in the training set
- Re-rank annotation based on this ratio

Score distribution modeling



Outline

- TRECVID description
- Our retrieval models
- Models for Image Annotation-- $p(q_w | d_v)$
- Models for Text Annotation -- $p(q_v | d_w)$
- Next Steps

Component $p(q_v | d_w)$: Text Annotation

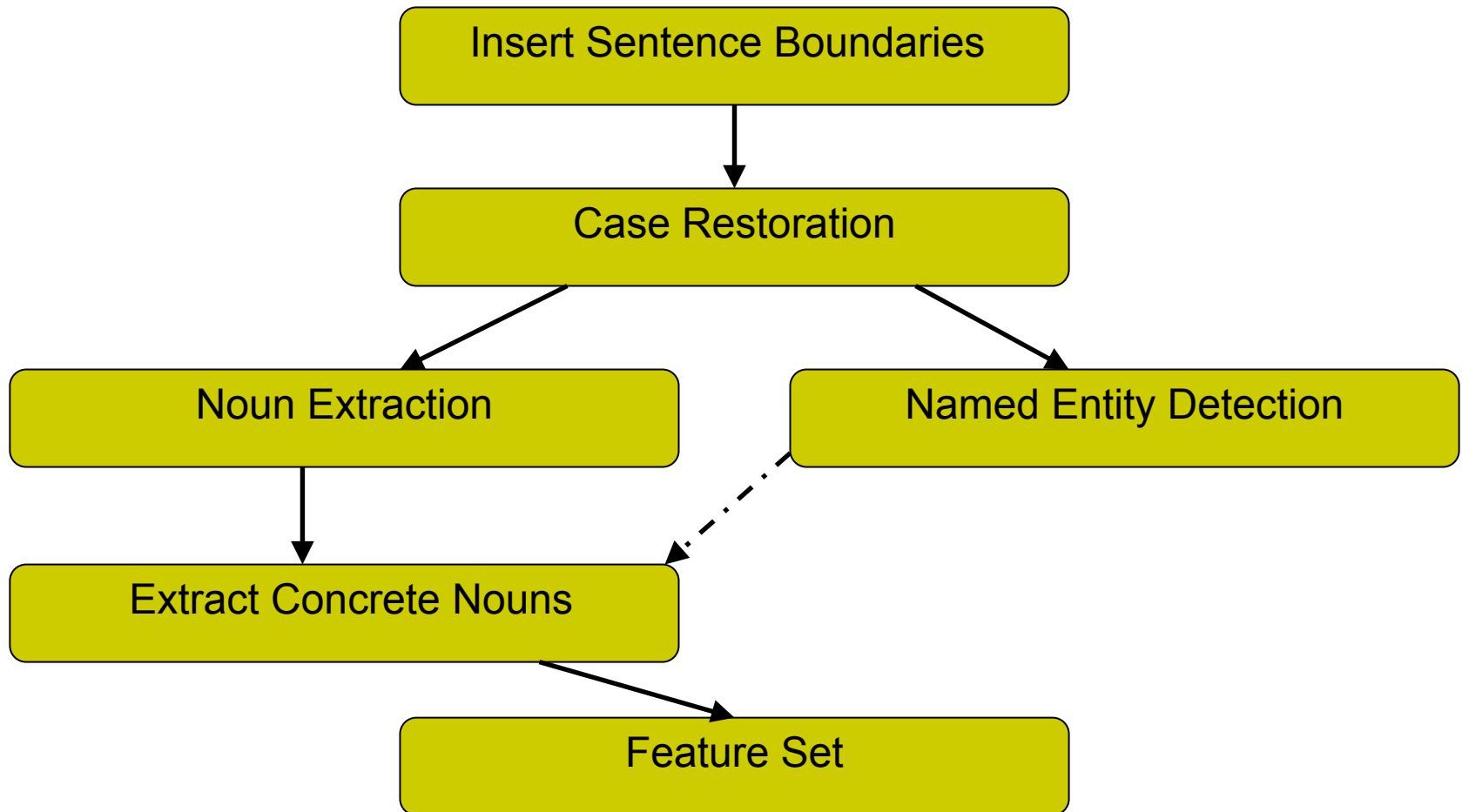
$$p(q_v | d_w) = \sum_c p(q_v | c) p(c | d_w)$$

- Given visual part of the query, process it using one of the previous models to get $p(q_v | c)$

Text Annotation Models

- Preprocess ASR text to extract features
 - Named Entities, Concrete Nouns
- Feature Selection
 - Information Gain
- Models tried
 - LM-based, Naive Bayes, SVM, MaxEnt

Building Features



Language Model Based Annotation

- Train two language models:
 - $P_c(f_1 \dots f_m)$
 - $P_{nc}(f_1 \dots f_m)$
- Pick the one that minimizes perplexity on test set
- Issues:
 - Picking features from ASR

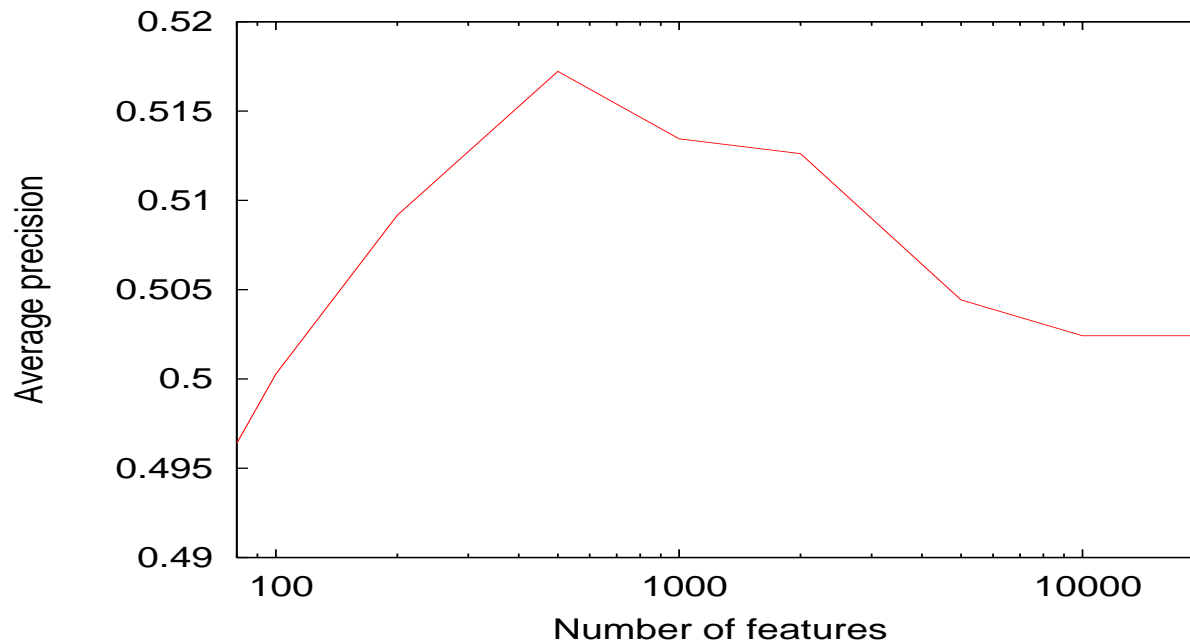
Important features for "face"

Feature f	Rel. Position	Count	$P(f,C)/P(f)/P(C)$
Empty	Previous shot	2855	0.70
Empty	This shot	1251	0.56
Empty	Next shot	845	0.56
NE_Person-male	This shot	1773	1.30
NE_JobTitle	This Shot	869	1.36
...

Total number of shots: 28054

- No annotation: good indicator for absence of "face"
- No single strong positive feature

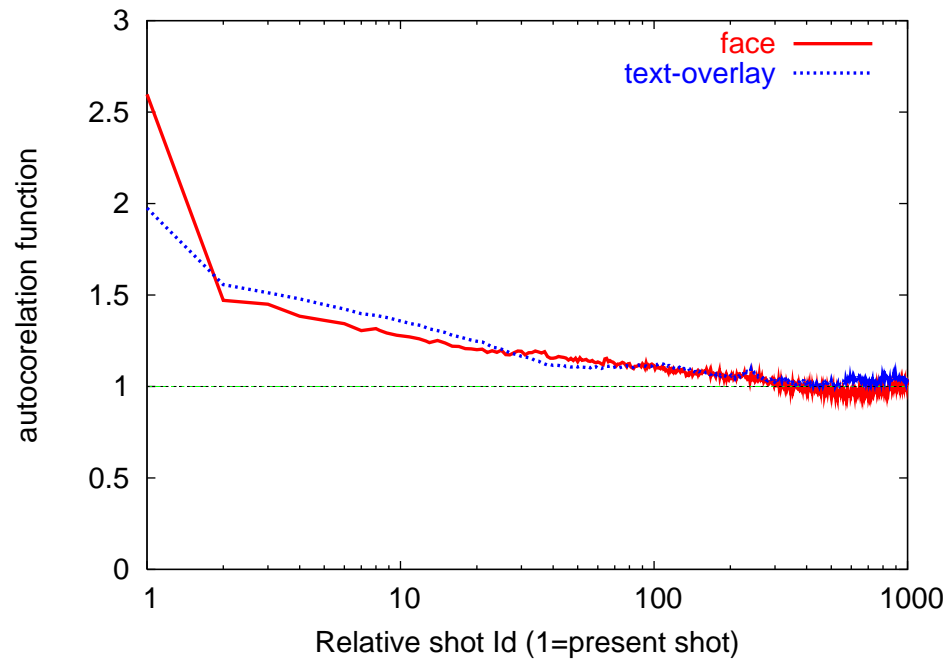
Optimal Number of Features



Concept:
Face

- A few thousand features are sufficient

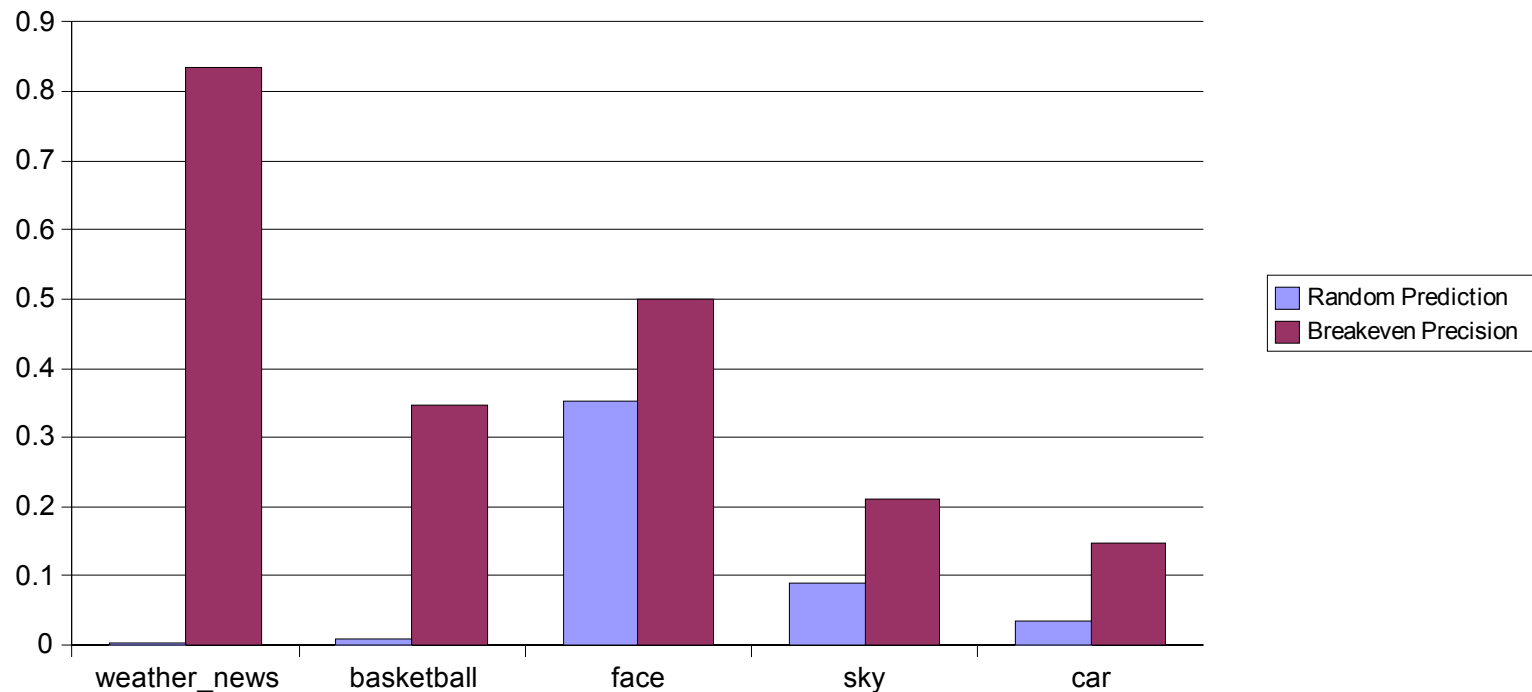
Temporal correlation of concepts



- Weak temporal correlations
(Exception: concept "monologue")

Naive Bayes classifier results for select concepts

Binary Naive Bayes Classification



Outline

- TRECVID description
- Our retrieval models
- Models for Image Annotation-- $p(q_w | d_v)$
- Models for Auto-Illustration-- $p(q_v | d_w)$
- **Next Steps**

Next Steps

- Integration Experiments
 - Evaluate linear and log-linear models on TRECVID03 queries
- Improve component models & iterate ...