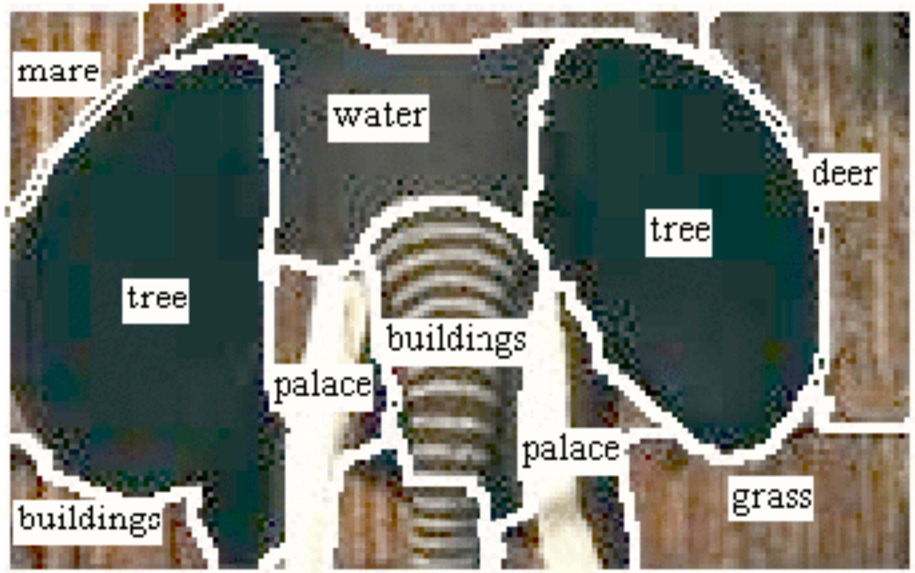


Matching Words and Pictures

Kobus Barnard, Pinar Duygulu,
David Forsyth, Nando de Freitas,
David M. Blei, Michael I. Jordan

Multi-modal data: Images and Text

Applications

- Automated image annotation
- Image search via text query

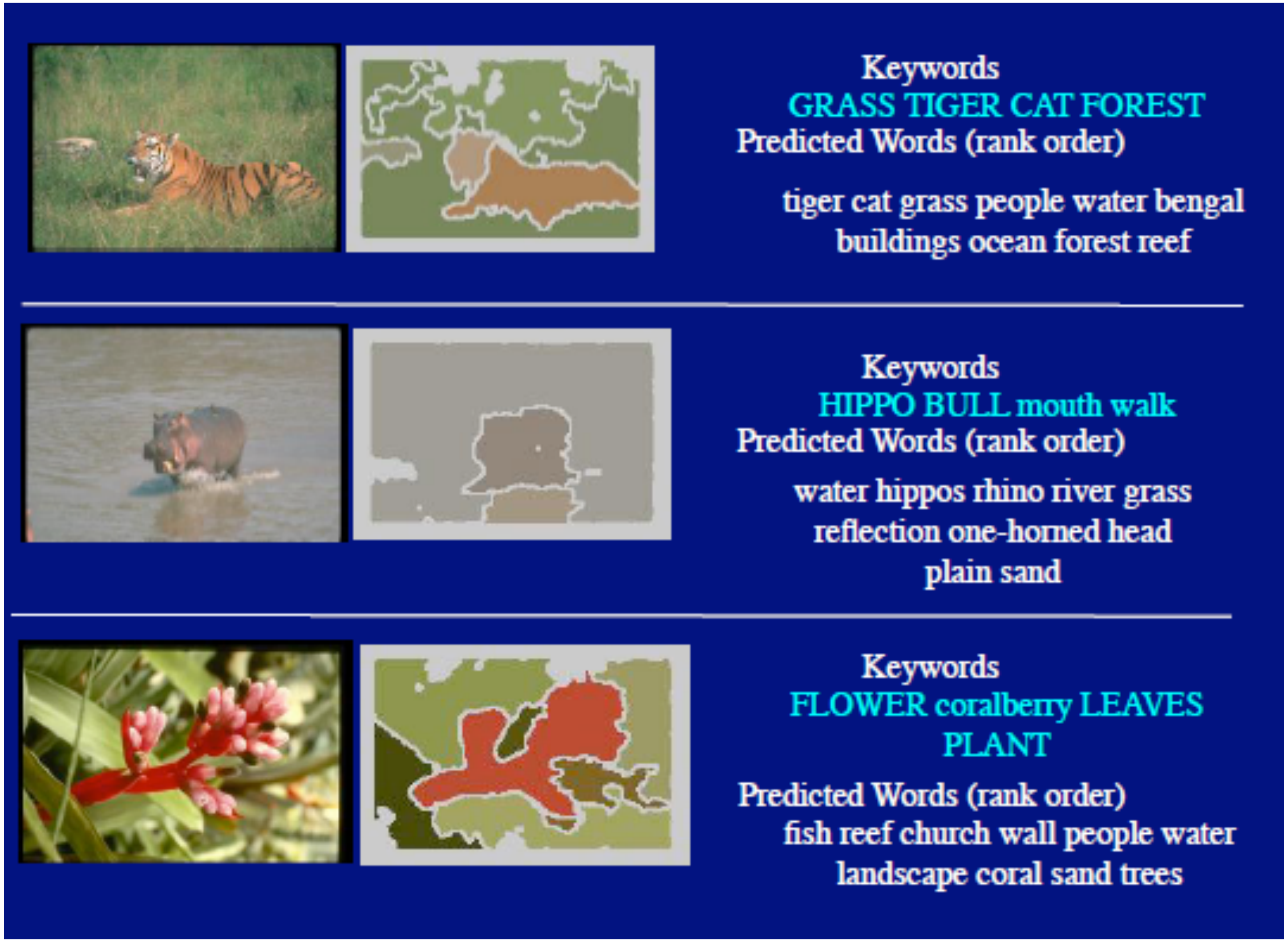
Tying Text to Images: Motivation

Auto-Annotation
  Generate textual descriptions for images

Auto-Illustration
  Select images from textual descriptions

Correspondence
  Tie semantic description directly to a subregion

Auto Annotation

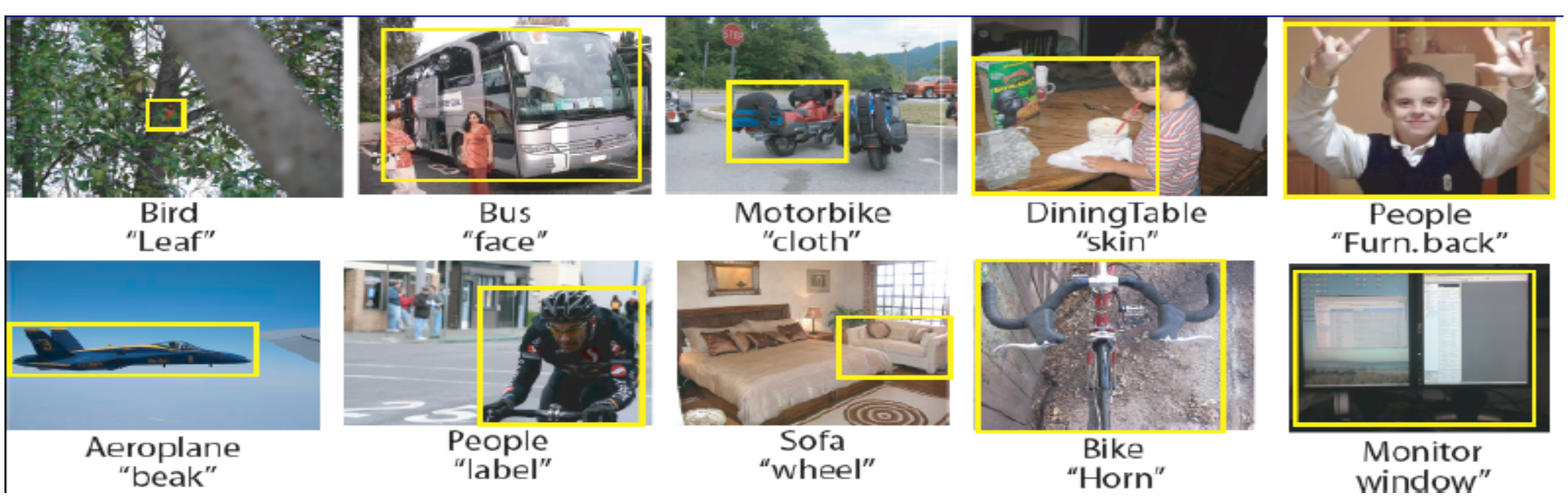


Auto Annotation: Describing Objects

Auto Annotation: Describing Objects

What's missing?

What's interesting?



Correspondence

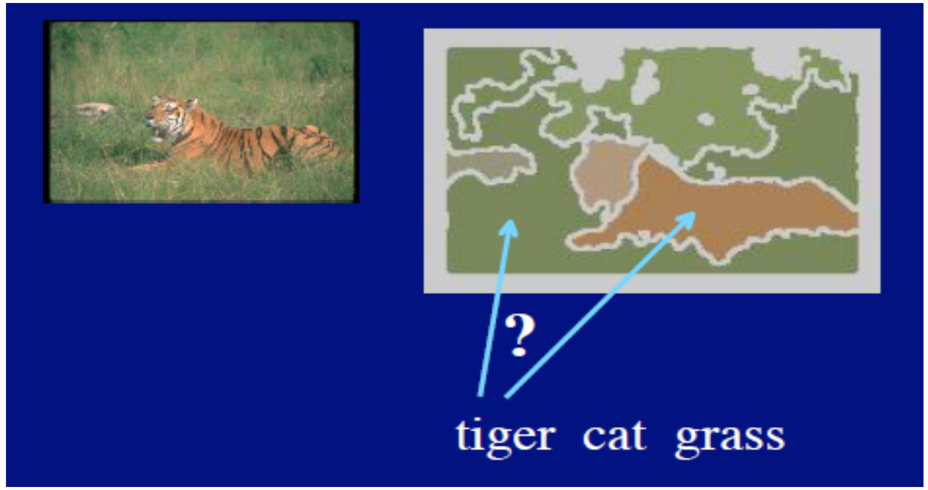
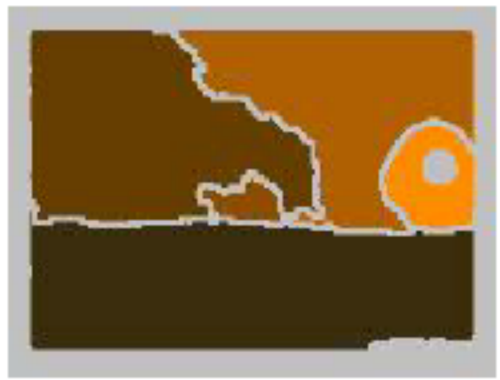


Image Representation

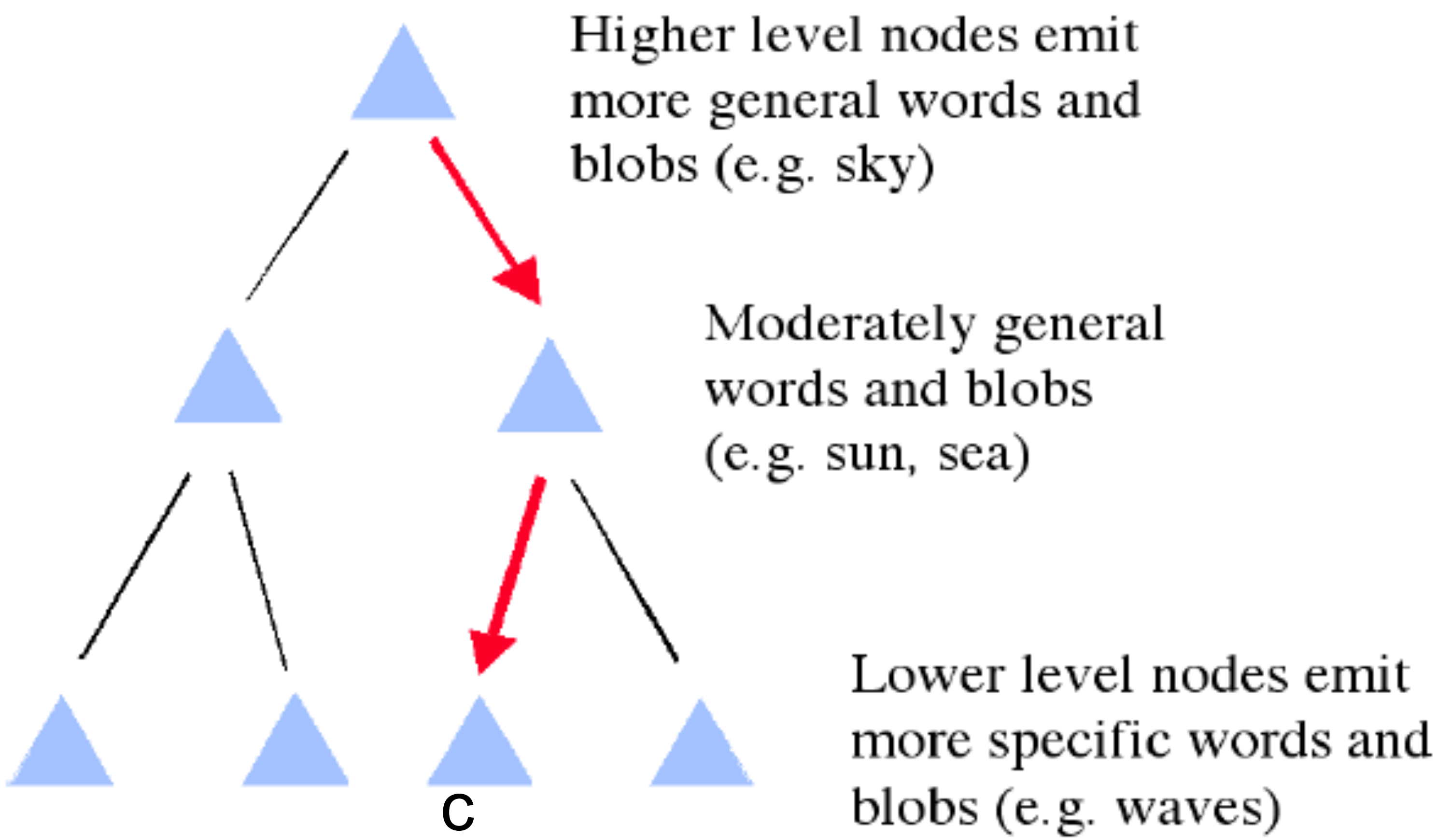
Segmented using normalized cuts (Shi, Malik)



Per region features:

- Size, Position, Color (mean, std. dev.)
- Texture Filter Responses (mean, var),
- Shape
  - area/perimeter^2
  - area/conv. hull area

Model 1:
Multi-Modal Hierarchical Aspect Model



Linear (degenerate) tree - 1 child per parent
Binary tree - 2 children per parent

Model 1: Parameter Description

D - a given document composed of:
  W = {w} - words (multinomial model)
  B = {b} - image regions (gaussian model)

c - cluster index (leaf of tree)
l - level of tree
(c,l) uniquely determines a node in the tree

N_w - Maximum number of words in any document
$N_{w,d}$ - Number of words in D

N_b - Maximum number of regions in any document
$N_{b,d}$ - Number of regions in D

Model 1: Variants

Model I0

$$p(D|d) = \sum_c p(c) \prod_{w \in W} \left[\sum_l p(w|l,c) p(l|d) \right]^{\frac{N_w}{N_{w,d}}} \prod_{b \in B} \left[\sum_l p(b|l,c) p(l|d) \right]^{\frac{N_b}{N_{b,d}}}$$

Model I1

$p(l|d)$ becomes $p(l|c,d)$

Model I2

$$p(D) = \sum_c p(c) \prod_{w \in W} \left[\sum_l p(w|l,c) p(l|c) \right]^{\frac{N_w}{N_{w,d}}} \prod_{b \in B} \left[\sum_l p(b|l,c) p(l|c) \right]^{\frac{N_b}{N_{b,d}}}$$

Parameter Learning

Hidden Variables

Document's cluster index (c)

Specificity of word (l - depth in tree)

EM

Given cluster, depth assignments, can easily estimate probabilities

Given probability distributions, can easily estimate assignments

$$p(D) = \sum_{c} p(c) \prod_{w \in W} \left[\sum_{l} p(w|l,c) p(l|c) \right]^{\frac{N_w}{N_{w,d}}} \prod_{b \in B} \left[\sum_{l} p(b|l,c) p(l|c) \right]^{\frac{N_b}{N_{b,d}}}$$

Update rules similar to mixture model EM (extends Hofmann Puzicha '98)

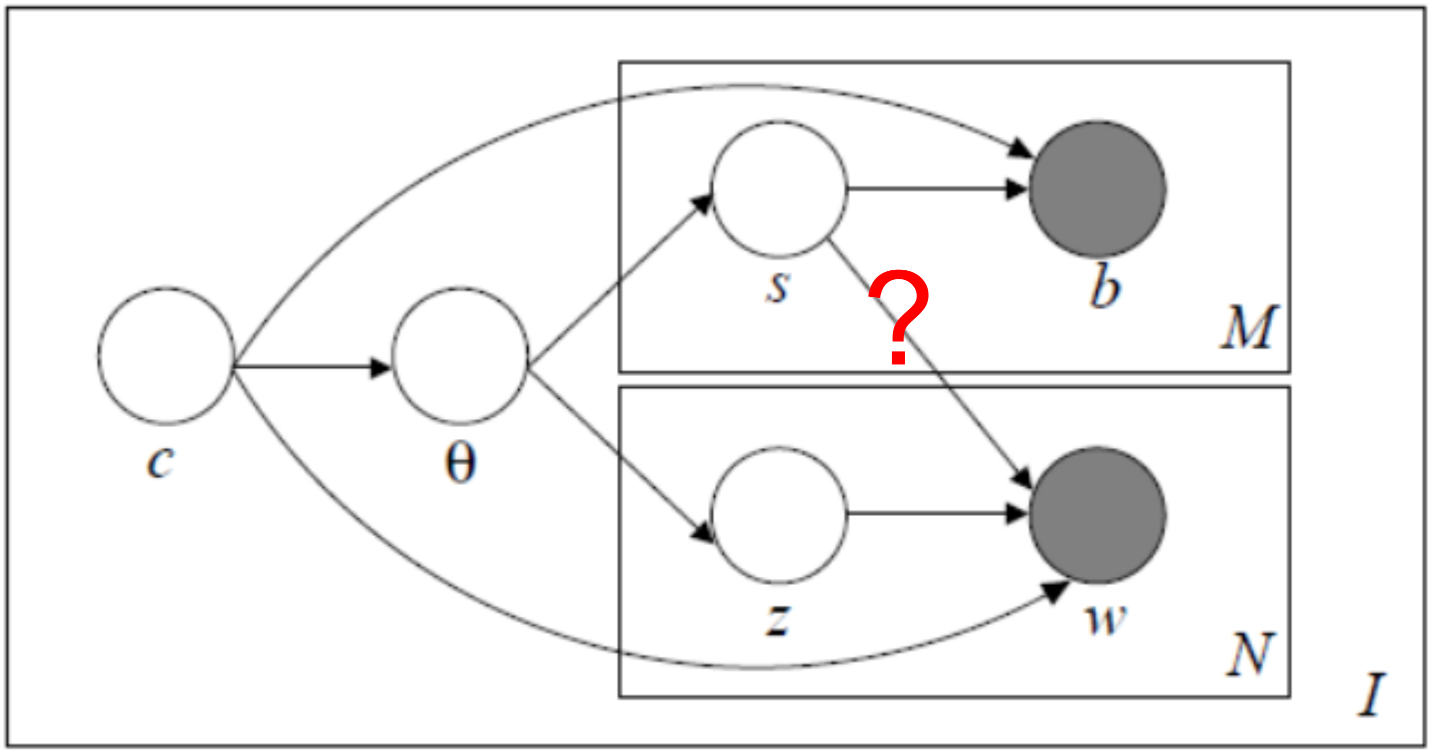
Image Based Word Prediction

Predict word give regions from an image

$$p(w|B) \propto \sum_c p(c)p(w|c)p(B|c)$$

$$p(w|B) = \sum_c p(c)\left[\sum_l p(w|l,c)p(l|c)\right]\prod_{b\in B}\left[\sum_l p(b|l,c)p(l|c)\right]^{\frac{N_b}{N_{b,d}}}$$

Mixture of Multi-Modal Latent Dirichlet Allocation



Mixture of Multi-Modal Latent Dirichlet Allocation

1. Choose one of J mixture components $c \sim \text{Multinomial}(\eta)$.
2. Conditioned on the mixture component, choose a mixture over J factors, $\theta \sim \text{Dir}(\alpha_c)$.
3. For each of the N words:
   (a) Choose one of K factors $z_n \sim \text{Multinomial}(\theta)$.
   (b) Choose one of V words w_n from $p(w_n|z_n, c, \beta)$, the conditional probability of w_n given the mixture component and latent factor.
4. For each of the M blobs:
   (a) Choose a factor $s_m \sim \text{Multinomial}(\theta)$.
   (b) Choose a blob b_m from $p(b_m|s_m, c, \mu, \Sigma)$, a multivariate Gaussian distribution with diagonal covariance, conditioned on the factor s_m and the mixture component c.

Parameters

- A J-dimensional multinomial parameter η.
- A $J \times K$ matrix α where α_c is is the J-dimensional Dirichlet parameter conditioned on mixture component.
- A $J \times K \times V$ matrix β where β_{cz} is the distribution over words conditioned on the mixture component and hidden factor.
- A $J \times K \times D$ matrix μ and a $J \times K \times D$ matrix Σ where μ_{cs} and Σ_{cs} are parameters to the D-dimensional multivariate Gaussian distribution over blobs, conditioned on the mixture component and hidden factor.

EM algorithm with a variational E step

Correspondence

Rather than predict words for the whole image, attempt to associate particular words with particular
image regions

Method 0: Direct Translation

Build translation model between words and regions
- Assume one-one correspondence

Alignment: missing data problem

Convert Each region to a "word"
- Vector quantize via k-means

Method 1: Correspondence from a Hierarchical Clustering Model

If a word and an image region always co-occur, their correspondence can be captured by the clustering model

Region only:

$$p(w|b) \propto \sum_c p(c) \sum_l p(l) p(w|l,c) p(b|l,c)$$

Region-cluster:

replace p(c) with p(c|B)

Method 2: Integrating Correspondence and Hierarchical Clustering

D-0 model (D for dependent)

Words are generated implicitly conditioned on regions

$$p(D|d) = \sum_{c} p(c) \prod_{w \in W} \left[\sum_{l} p(w|l,c) p(l|B,c,d) \right]^{\frac{N_w}{N_{w,d}}} \prod_{b \in B} \left[\sum_{l} p(b|l,c) p(l|d) \right]^{\frac{N_b}{N_{b,d}}}$$

Wher
e

$$p(l|B,c,d) \propto \sum_{b \in B} p(l|b,c,d)$$

Method 3: Paired Word and Region Emission at Nodes
C-0 model

$D = \{(w,b)_i\}$

$$p(D|d) = \sum_c p(c) \prod_{(w,b)\in D} \left[\sum_l p((w,b)|l,c)p(l|d) \right]$$

Need to estimate correspondence as part of the training process.

Find the correspondence:

$$p(w \Leftrightarrow b) \approx \sum_c p(c) \sum_l p((w,b)|l,c)p(l|d)$$

Evaluation methods

Measuring annotation performance

- Comparing the words predicted by various models with words actually present for test data.

- Some words are frequent. The increment of performance over the empirical density is a sensible indicator

Measurement

- KL divergence between the predictive distribution and the target distribution

$$E_{KL}^{(model)} = \sum_{w \in vocabulary} p(w) \log \frac{p(w)}{q(w|B)}$$

- Interested in knowing improvement over empirical distribution

$$E_{KL} = \frac{1}{N} \sum_{data} \left(E_{KL}^{(empirical)} - E_{KL}^{(model)} \right)$$

Measuring Correspondence Performance

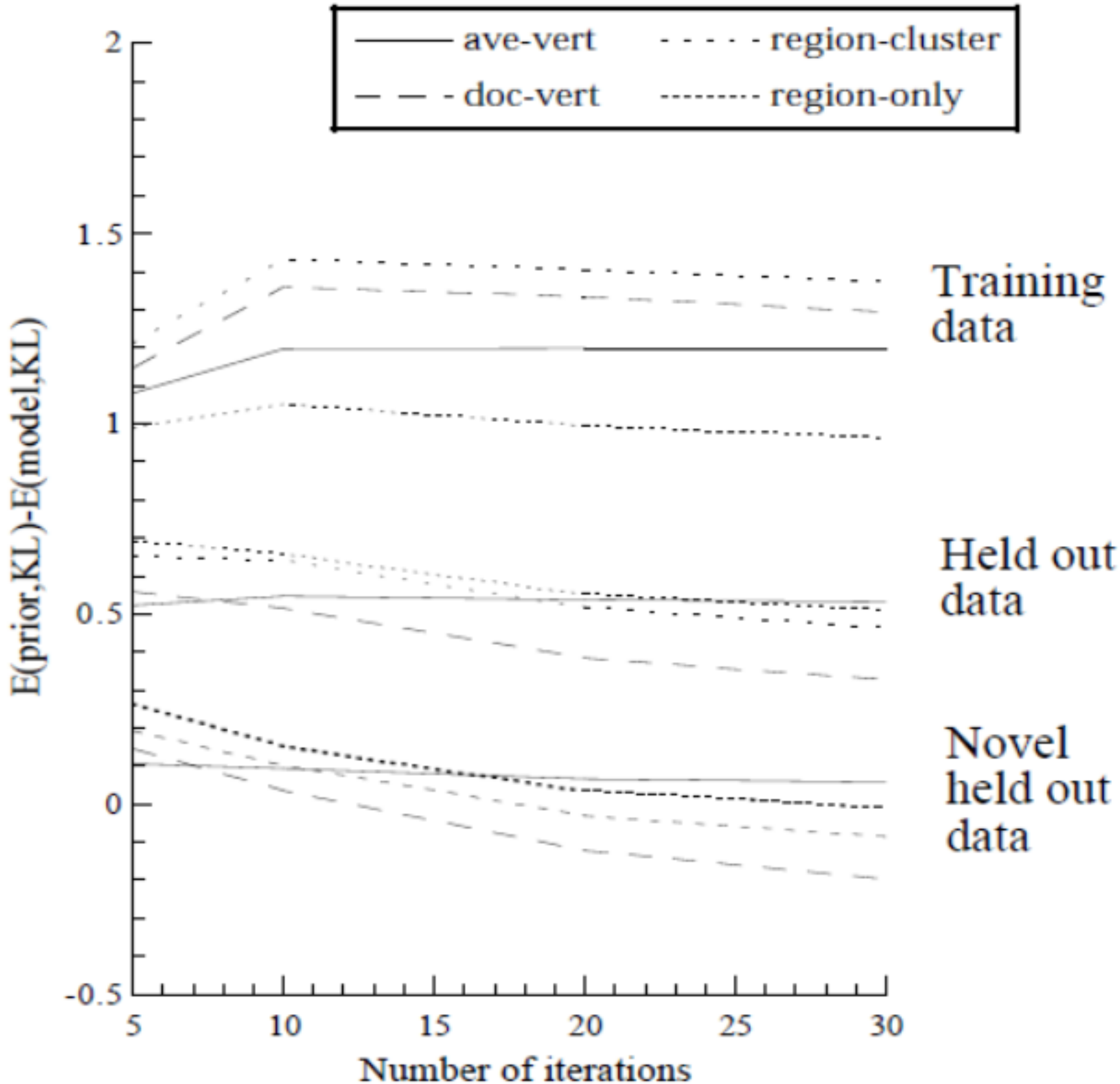
- Using annotation as a proxy
- Manual correspondence scoring

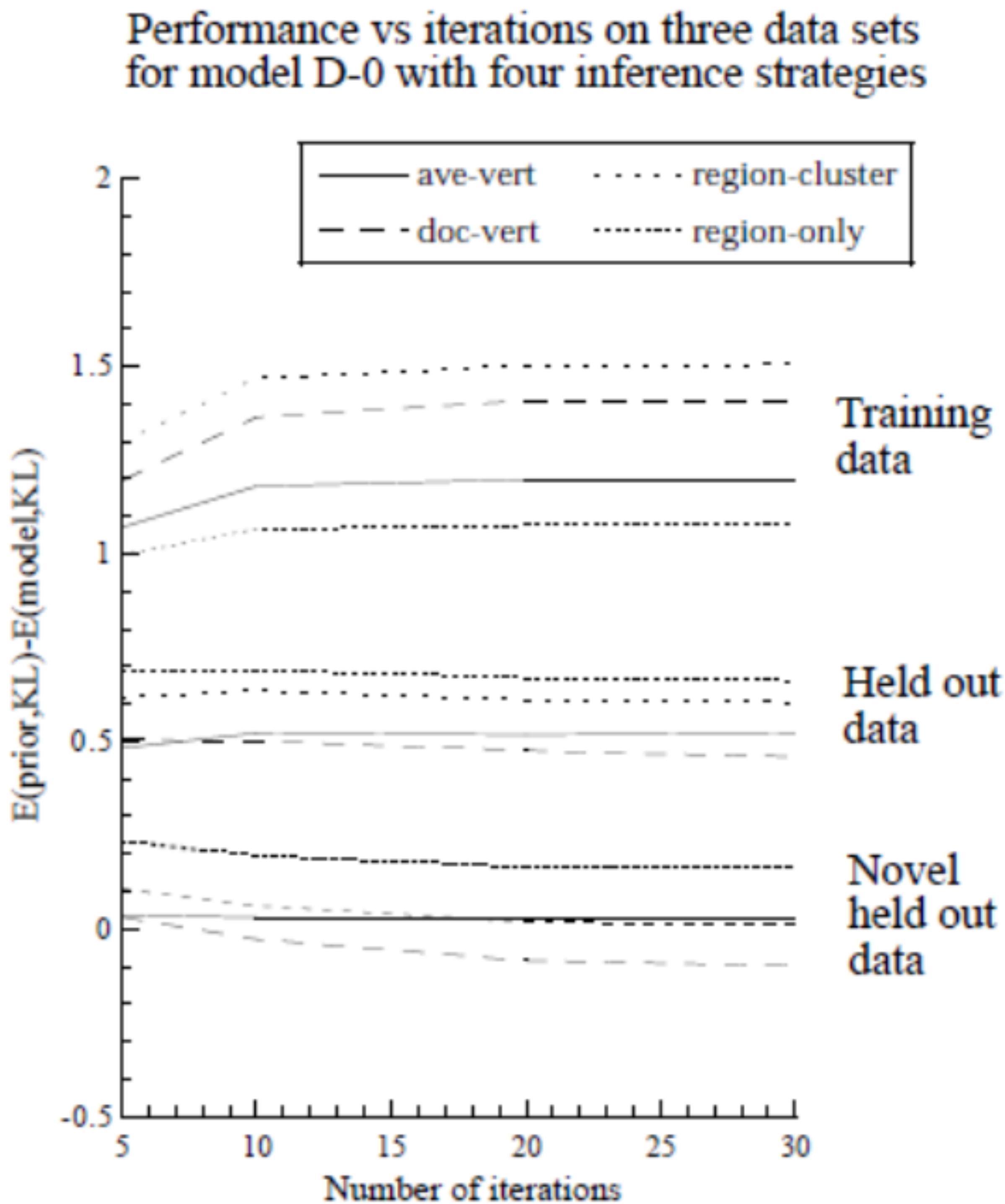
Experiments

Experimental setting

- Corel image data set
- 600 training images; 200 test images
- 155 words
- Images are segmented using N-Cuts
- Image features: size, position, color, oriented energy (12 filters), and a few simple shape features.

Performance vs iterations on three data sets
for model I-0 with four inference strategies



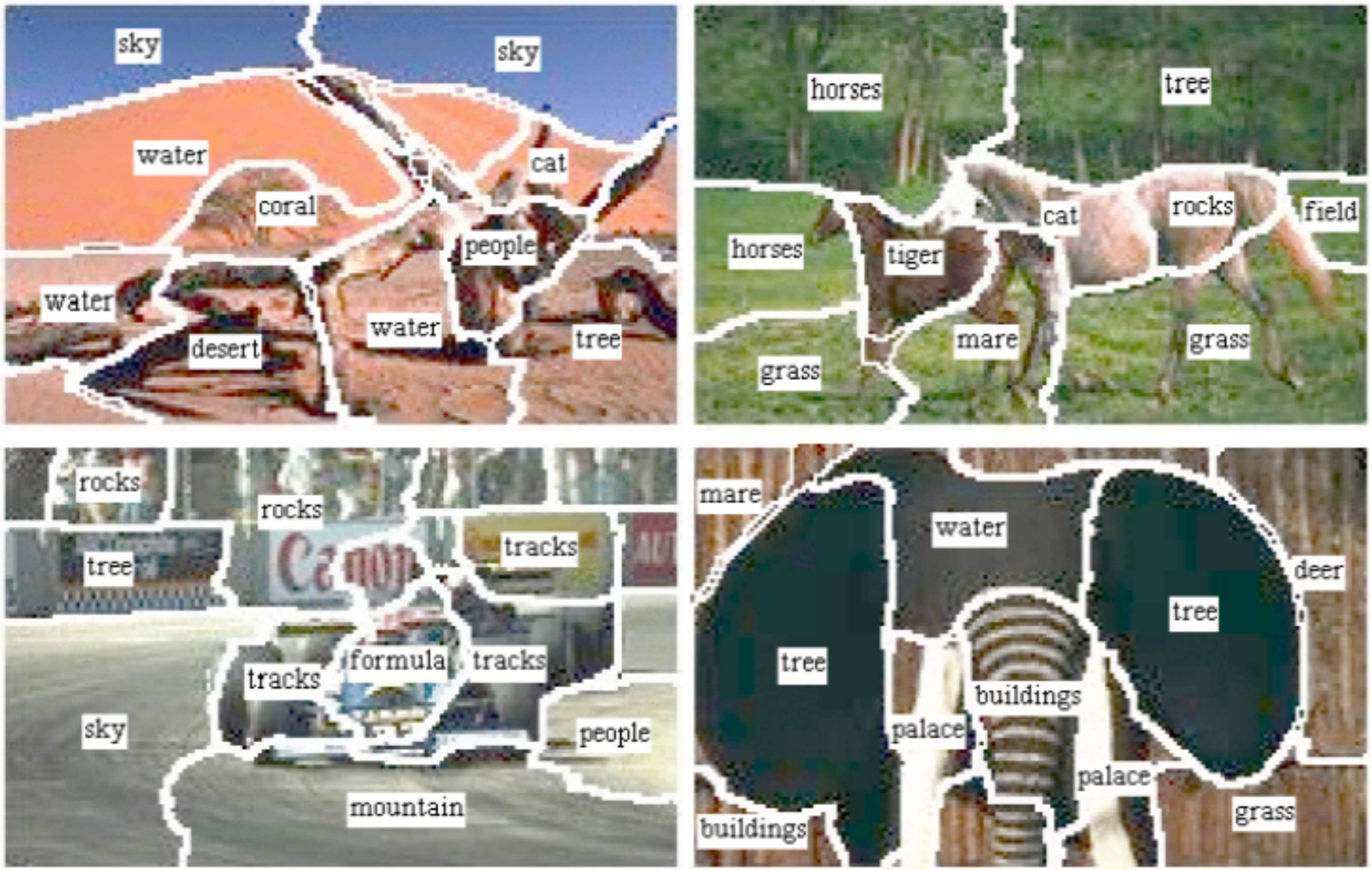

Performance vs iterations on three data sets
for model D-0 with four inference strategies
ave-vert
doc-vert
region-cluster
region-only
E(prior,KL)-E(model,KL)
Number of iterations
2
1.5
1
0.5
0
-0.5
5
10
15
20
25
30
Training data
Held out data
Novel held out data

Method	Training data	Held out data	Novel data
linear-I-0-doc-vert	1.235 (0.02)	0.688 (0.02)	0.258 (0.01)
binary-I-0-ave-vert	1.210 (0.03)	0.563 (0.02)	0.060 (0.01)
binary-I-0-doc-vert	1.385 (0.02)	0.587 (0.02)	0.061 (0.02)
binary-I-0-region-cluster	1.429 (0.03)	0.651 (0.02)	0.094 (0.02)
binary-I-0-region-only	1.061 (0.02)	0.684 (0.02)	0.160 (0.02)
binary-I-2-ave-vert	1.367 (0.03)	0.608 (0.02)	0.084 (0.01)
binary-I-2-doc-vert	1.320 (0.03)	0.627 (0.02)	0.129 (0.01)
binary-I-2-region-cluster	1.342 (0.03)	0.694 (0.02)	0.156 (0.01)
binary-I-2-region-only	1.016 (0.02)	0.709 (0.02)	0.211 (0.01)
linear-D-0-doc-vert	1.376 (0.02)	0.714 (0.02)	0.268 (0.01)
binary-D-0-ave-vert	1.169 (0.03)	0.550 (0.02)	0.057 (0.01)
binary-D-0-doc-vert	1.417 (0.03)	0.601 (0.02)	0.074 (0.01)
binary-D-0-region-cluster	1.466 (0.03)	0.669 (0.02)	0.105 (0.02)
binary-D-0-region-only	1.086 (0.02)	0.700 (0.02)	0.175 (0.02)
binary-D-2-ave-vert	1.310 (0.005)	0.627 (0.003)	0.089 (0.005)
binary-D-2-doc-vert	1.589 (0.005)	0.674 (0.003)	0.102 (0.005)
binary-D-2-region-cluster	1.613 (0.005)	0.739 (0.003)	0.132 (0.005)
binary-D-2-region-only	1.155 (0.005)	0.747 (0.003)	0.180 (0.005)
linear-C-0-region-only	0.980 (0.02)	0.472 (0.02)	0.106 (0.01)
binary-C-0-ave-vert	1.020 (0.02)	0.516 (0.02)	0.071 (0.01)
binary-C-0-doc-vert	1.205 (0.02)	0.541 (0.02)	0.042 (0.01)
binary-C-0-region-cluster	1.254 (0.02)	0.601 (0.02)	0.104 (0.01)
binary-C-0-region-only	1.015 (0.02)	0.643 (0.02)	0.179 (0.01)
discrete-translation	1.347 (0.02)	0.433 (0.002)	-0.072 (0.01)
MoM-LDA	0.452 (0.01)	0.401 (0.01)	0.171 (0.01)

Take home messages:

1. Explicitly (or implicitly) modeling correspondence helps to do annotation
2. The LDA model doesn't work so well
3. All the models work better than directly using empirical distribution of words


grass
cat
horses
buildings
tiger
grass
mare
mare
ground
flowers
ground
antlers
shops
town
crab
tiger
town
ground
garden
flowers
plants
petals
flowers
forest
petals
tree
paintings
plants
aterfalls
sky
row
buildings
building
buildings
house
people
tree
tree
plane
plane
sky
tree
sky
people
people
garden
swimmers
people
people
people
tiger
flowers
street


sky
sky
water
cat
coral
people
water
water
desert
tree
horses
tree
cat
rocks
field
horses
tiger
mare
grass
grass
rocks
rocks
tracks
tree
formula
tracks
tracks
sky
people
mountain
mare
water
deer
tree
tree
buildings
palace
palace
grass
buildings

Correspondence evaluation

Method	PR measure
linear-I-0-region-only	0.099 (0.02)
binary-I-0-region-cluster	0.101 (0.01)
binary-I-0-region-only	0.103 (0.01)
binary-I-2-region-cluster	0.101 (0.01)
binary-I-2-region-only	0.093 (0.01)
linear-D-0-region-only	0.132 (0.01)
binary-D-0-region-cluster	0.096 (0.01)
binary-D-0-region-only	0.104 (0.01)
binary-D-2-region-cluster	0.103 (0.01)
binary-D-2-region-only	0.092 (0.01)
linear-C-0-region-only	0.101 (0.01)
discrete-translation	0.066 (0.01)

Correspondence model doesn't do much better on this task

Table 4: Correspondence performance as measured over 10 sets of 50 manually annotated images from the held out set using the PR measure. All values are relative to the performance using the empirical distribution (about 0.094). For this task, the PR is arguably the most indicative measure as it corresponds to forcing each region to only emit a small number of words (the number of alternative labels). The NS measure is not appropriate because the refuse to predict level was calibrated under different conditions. Note that for comparison with the annotation results, linear-I-0-region-only and linear-I-0-doc-vert give the same results, as do linear-D-0-doc-vert and linear-D-0-region-only.

Conclusion

- A variety of methods for predicting words from pictures
- Data sets contain free text annotations?
- The effect of supervision?

Thanks