

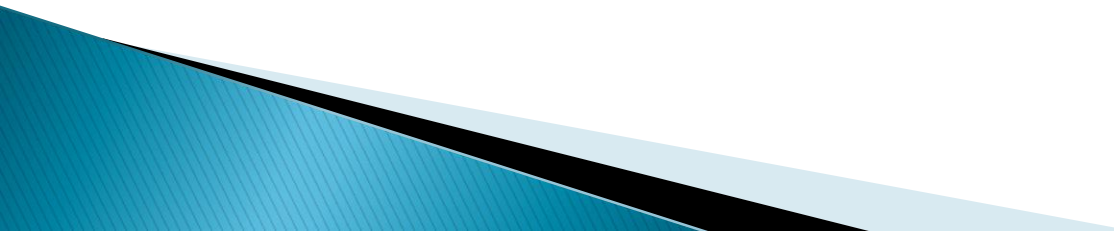
# Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary

P. Duygulu, K.Barnard, J.F.G. de Freitas, D.A. Forsyth

Presented by: Martino Buffolino



# Auto-Annotation

- ▶ Predicts words for a given image based on high posterior probability.
  - ▶ Doesn't tell us which image structure gave rise to which word.
  - ▶ This is not object recognition!
- 

# Recognition as Translation

- ▶ Similar to machine translation.
- ▶ Example:
  - C  $\rightarrow$  Assembly
  - (Image Regions; French)  $\rightarrow$  (Words; English)
- ▶ Lexicon's are learned from a dataset known as Aligned Bitext.
  - Any Problems?
    - Determining precise correspondences.

# Using EM to Learn a Lexicon

## ▶ Procedure

- Segment images into regions
- Learn to predict words using regions

## ▶ Problem?

- Machine translation maps discrete objects to other discrete objects
- Features associated with image regions do not occupy discrete space!

## ▶ Simple Solution

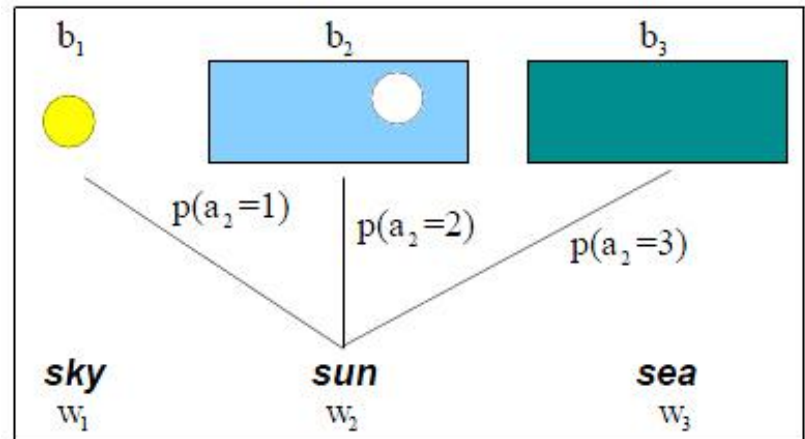
- K-means

# Using EM to Learn a Lexicon

- ▶ Difficulty in learning
  - Datasets do not provide explicit correspondence
  - Instead they provide probabilities
- ▶ How do we associate blob tokens with word tokens?
  - (Blob = label associated with a region)

- ▶ Use EM!

$b$  = blob  
 $w$  = word



# Models

$$p(w|b) = \prod_{n=1}^N \prod_{j=1}^{M_n} \sum_{i=1}^{L_n} p(a_{nj} = i) t(w = w_{nj} | b = b_{ni})$$

- ▶ Find the probability of word  $w$ , given blob  $b$ .
- ▶ This leads to the EM formulation.

$$\theta^{\text{ML}} = \arg \max_{\theta} p(w|b, \theta) = \arg \max_{\theta} \sum_a p(a, w|b, \theta).$$

- ▶ Find the maximum likelihood parameters

## Initialise

### E step

1. For each  $n = 1, \dots, N$ ,  $j = 1, \dots, M_n$  and  $i = 1, \dots, L_n$ , compute

$$\tilde{p}(a_{nj} = i | w_{nj}, b_{ni}, \theta^{(\text{old})}) = p(a_{nj} = i) t(w_{nj} | b_{ni}) \quad (5)$$

2. Normalise  $\tilde{p}(a_{nj} = i | w_{nj}, b_{ni}, \theta^{(\text{old})})$  for each image  $n$  and word  $j$

$$p(a_{nj} = i | w_{nj}, b_{ni}, \theta^{(\text{old})}) = \frac{\tilde{p}(a_{nj} = i | w_{nj}, b_{ni}, \theta^{(\text{old})})}{\sum_{i=1}^{L_n} p(a_{nj} = i) t(w_{nj} | b_{ni})} \quad (6)$$

### M step

1. Compute the mixing probabilities for each  $j$  and image of the same size (e.g.  $L(n) = l$  and  $M(n) = m$ )

$$p(a_{nj} = i) = \frac{1}{N_{l,m}} \sum_{n:L(n)=l, M(n)=m}^N p(a_{nj} = i | w_{nj}, b_{ni}, \theta^{(\text{old})}) \quad (7)$$

where  $N_{l,m}$  is the number of images of the same length.

2. For each different pair  $(b^*, w^*)$  appearing together in at least one of the images, compute

$$\tilde{t}(w_{nj} = w^* | b_{ni} = b^*) = \sum_{n=1}^N \sum_{j=1}^{M_n} \sum_{i=1}^{L_n} p(a_{nj} = i | w_{nj}, b_{ni}, \theta^{(\text{old})}) \delta_{(w^*, b^*)}(w_{nj}, b_{ni}) \quad (8)$$

where  $\delta_{(w^*, b^*)}(w_{nj}, b_{ni})$  is 1 if  $b^*$  and  $w^*$  appear in image and 0 otherwise.

3. Normalise  $\tilde{t}(w_{nj} = w^* | b_{ni} = b^*)$  to obtain  $t(w_{nj} = w^* | b_{ni} = b^*)$ .

# Applying and Refining the Lexicon

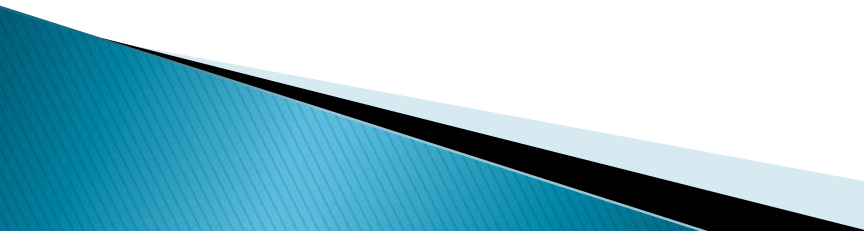
- ▶ Determine the blob that corresponds to each region by vector quantization (k-means)
- ▶ Annotate the region with the word that shows highest probability

# Applying and Refining the Lexicon

- ▶ All predictions are not expected to be good
  - Some blobs may not predict any word with high probability
- ▶ So we must prune!

$$p(\text{word}|\text{blob}) > \text{threshold}$$

# Clustering Indistinguishable Words

- ▶ Datasets won't contain a vocabulary that is perfect for annotating
  - ▶ Example:
    - eagle and a jet (appear the same in aerial view)
    - cat and tiger (visually indistinguishable)
    - mare or foals reliably predict horses
  - ▶ Instead cluster the similar words!
- 

# Evaluation

- ▶ Looking at the actual annotation
- ▶ Plot recall and precision



word	th = 0	th = 0.1	th = 0.2	th = 0.3	th = 0.4
	rec - prec	rec - prec	rec - prec	rec - prec	rec - prec
petals	0.50 - 1.00	0.50 - 1.00	0.50 - 1.00	0.50 - 1.00	0.50 - 1.00
sky	0.83 - 0.34	0.80 - 0.35	0.58 - 0.44		
flowers	0.67 - 0.21	0.67 - 0.21	0.44 - 0.24		
horses	0.58 - 0.27	0.58 - 0.27	0.50 - 0.26		
foals	0.56 - 0.29	0.56 - 0.29	0.56 - 0.29		
mare	0.78 - 0.23	0.78 - 0.23			
tree	0.77 - 0.20	0.74 - 0.20			
people	0.74 - 0.22	0.74 - 0.22			
water	0.74 - 0.24	0.74 - 0.24			
sun	0.70 - 0.28	0.70 - 0.28			
bear	0.59 - 0.20	0.55 - 0.20			
stone	0.48 - 0.18	0.48 - 0.18			
buildings	0.48 - 0.17	0.48 - 0.17			
snow	0.48 - 0.17	0.48 - 0.19			

# Effect of Retraining

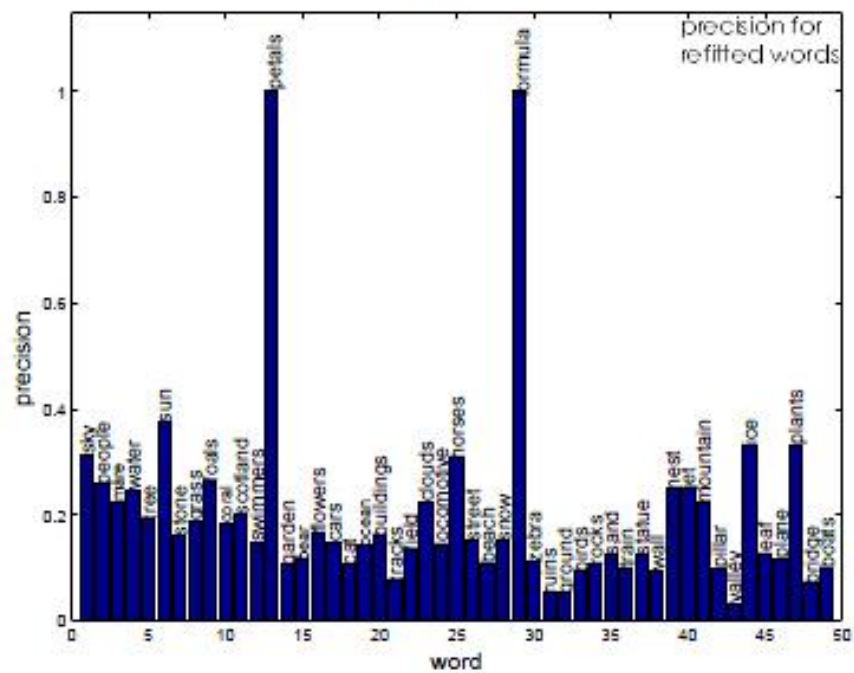
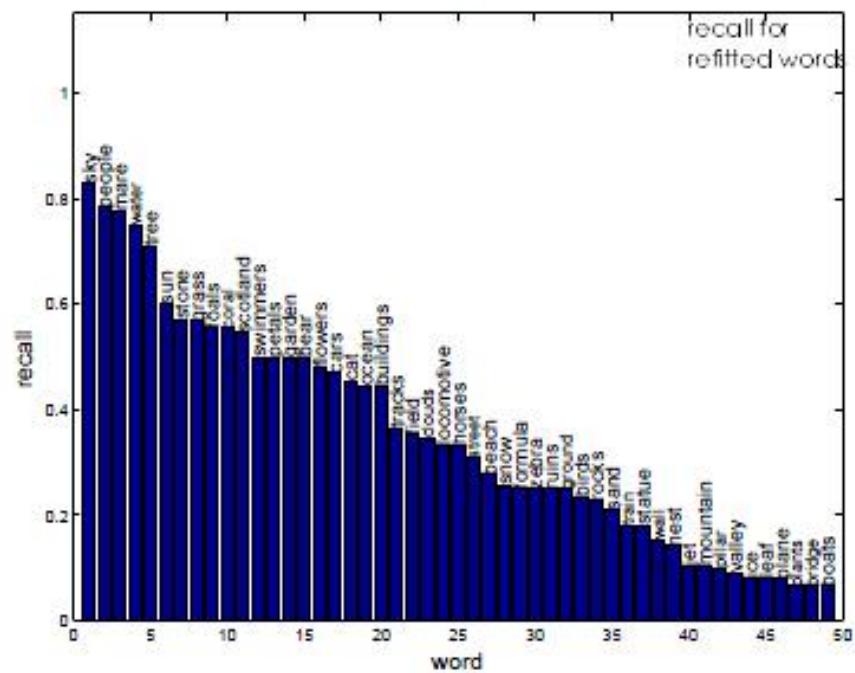
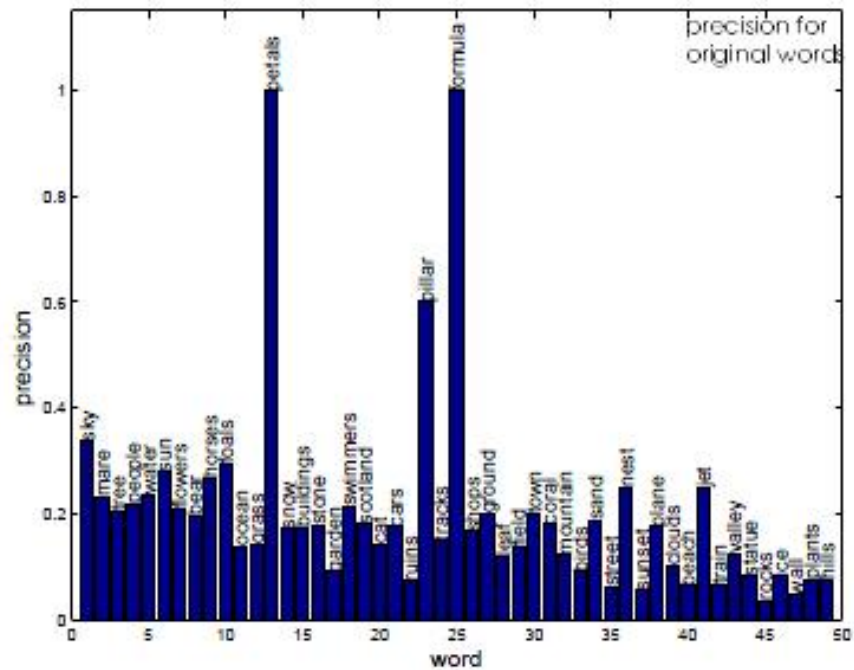
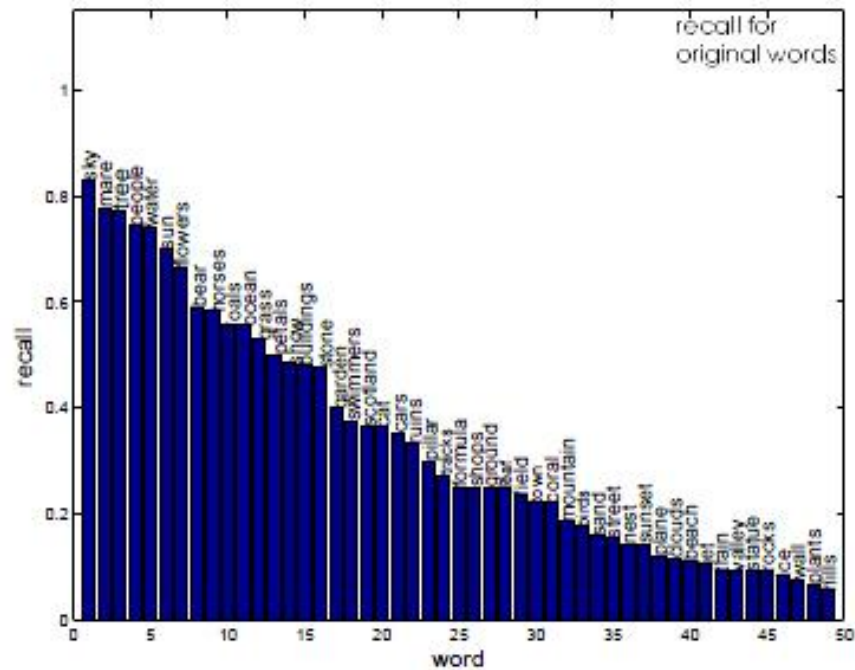
- ▶ Reduce your vocabulary to only words that can be predicted
- ▶ To do this, run EM again!

word	th = 0	th = 0.1	th = 0.2	th = 0.3	th = 0.4
	rec - prec	rec - prec	rec - prec	rec - prec	rec - prec
petals	0.50 - 1.00	0.50 - 1.00	0.50 - 1.00	0.50 - 1.00	0.50 - 1.00
sky	0.83 - 0.34	0.80 - 0.35	0.58 - 0.44		
flowers	0.67 - 0.21	0.67 - 0.21	0.44 - 0.24		
horses	0.58 - 0.27	0.58 - 0.27	0.50 - 0.26		
foals	0.56 - 0.29	0.56 - 0.29	0.56 - 0.29		
mare	0.78 - 0.23	0.78 - 0.23			
tree	0.77 - 0.20	0.74 - 0.20			
people	0.74 - 0.22	0.74 - 0.22			
water	0.74 - 0.24	0.74 - 0.24			
sun	0.70 - 0.28	0.70 - 0.28			
bear	0.59 - 0.20	0.55 - 0.20			
stone	0.48 - 0.18	0.48 - 0.18			
buildings	0.48 - 0.17	0.48 - 0.17			
snow	0.48 - 0.17	0.48 - 0.19			

after retraining

word	th = 0	th = 0.1	th = 0.2	th = 0.3	th = 0.4
	rec - prec	rec - prec	rec - prec	rec - prec	
petals	0.50 - 1.00	0.50 - 1.00	0.50 - 1.00	0.50 - 1.00	
sky	0.83 - 0.31	0.83 - 0.31	0.75 - 0.37	0.58 - 0.47	
people	0.78 - 0.26	0.78 - 0.26	0.68 - 0.27	0.51 - 0.31	
water	0.75 - 0.25	0.75 - 0.25	0.72 - 0.26	0.44 - 0.27	
mare	0.78 - 0.23	0.78 - 0.23	0.67 - 0.21		
tree	0.71 - 0.19	0.71 - 0.19	0.66 - 0.20		
sun	0.60 - 0.38	0.60 - 0.38	0.60 - 0.43		
grass	0.57 - 0.19	0.57 - 0.19	0.49 - 0.22		
stone	0.57 - 0.16	0.57 - 0.16	0.52 - 0.23		
foals	0.56 - 0.26	0.56 - 0.26	0.56 - 0.26		
coral	0.56 - 0.19	0.56 - 0.19	0.56 - 0.19		
scotland	0.55 - 0.20	0.55 - 0.20	0.45 - 0.19		
flowers	0.48 - 0.17	0.48 - 0.17	0.48 - 0.18		
buildings	0.44 - 0.16	0.44 - 0.16			

before



<b>1st clusters</b>	<b>r</b>	<b>p</b>	<b>2nd clusters</b>	<b>r</b>	<b>p</b>	<b>3rd clusters</b>	<b>r</b>	<b>p</b>
horses mare	0.83	0.18	kit horses mare foals	0.77	0.16	kit horses mare foals	0.77	0.27
leaf flowers	0.69	0.22	leaf flowers plants vegetables	0.63	0.25	leaf flowers plants vegetables	0.60	0.19
plane	0.12	0.14	jet plane arctic	0.46	0.18	jet plane arctic prop flight penguin dunes	0.43	0.17
pool athlete	0.33	0.31	pool athlete vines	0.17	0.50	pool athlete vines swimmers	0.75	0.27
sun ceiling	0.60	0.30	sun ceiling	0.70	0.30	sun ceiling cave store	0.62	0.35
sky beach	0.83	0.30	sky beach cathedral	0.82	0.31	sky,beach cathedral clouds mural arch waterfalls	0.87	0.36
water	0.77	0.26	water	0.72	0.25	water waves	0.70	0.26
tree	0.73	0.20	tree	0.76	0.20	tree	0.58	0.20
people	0.68	0.24	people	0.62	0.26	people	0.54	0.25



- ▶ Checking annotations is subjective!
- ▶ Each test image must be viewed by hand to tell whether an annotation of a region is correct.
- ▶ Words such as grass, tree, and sky are almost always correctly predicted!

