Translating Images to Words: A Novel Approach for Object Recognition

Pınar Duygulu - Şahin

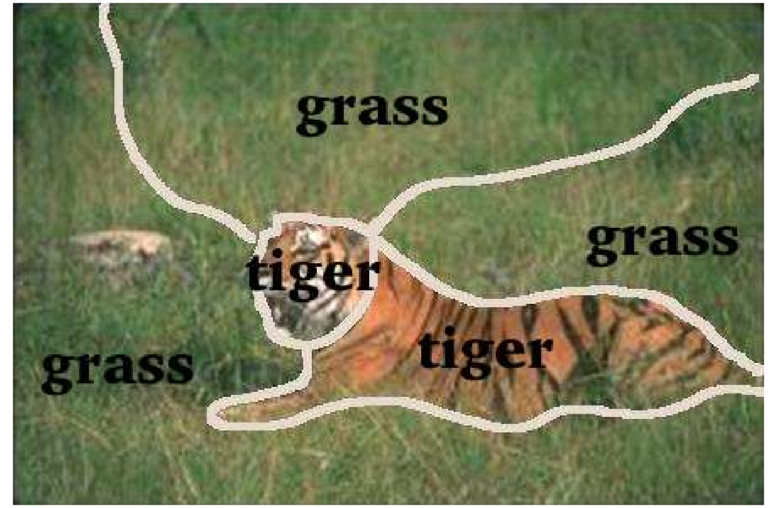
Dept. of Computer Engineering

Middle East Technical University

Linking Words to Images

Object recognition on a large scale is linking words with image regions

use joint probability of words and images in large data sets



tiger grass cat

Auto-annotation of images

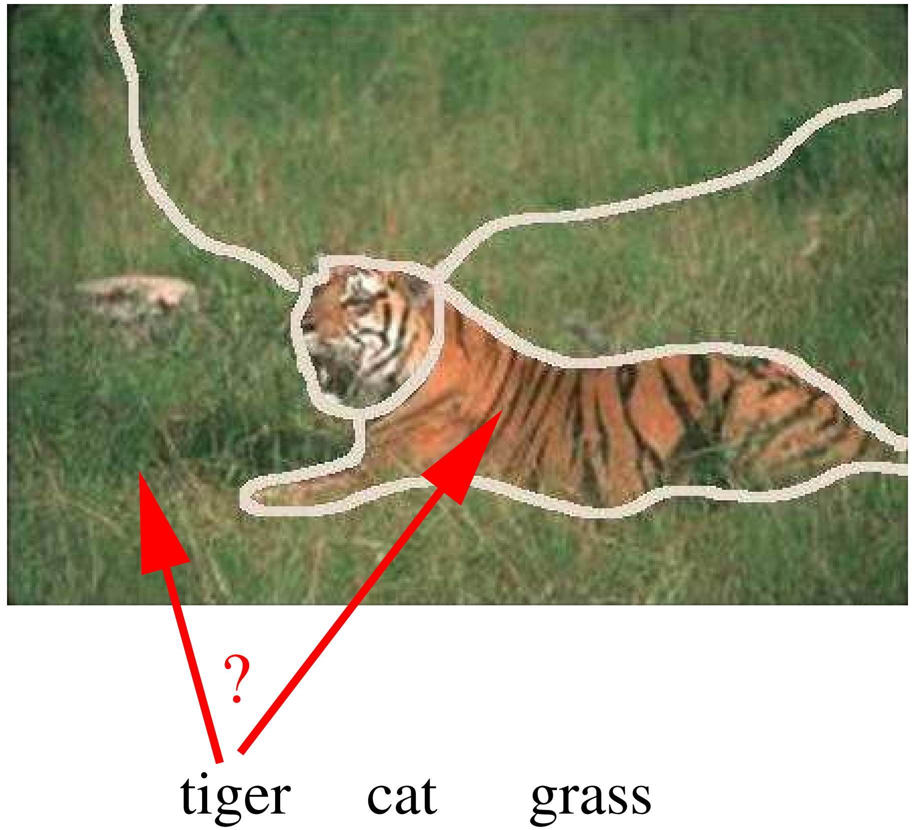
Predicting words for the images

tiger grass cat

Barnard and Forsyth (ICCV 2001), Barnard, Duygulu, Forsyth (CVPR2001)

Other related work : Maron98, Mori99

Annotation vs recognition

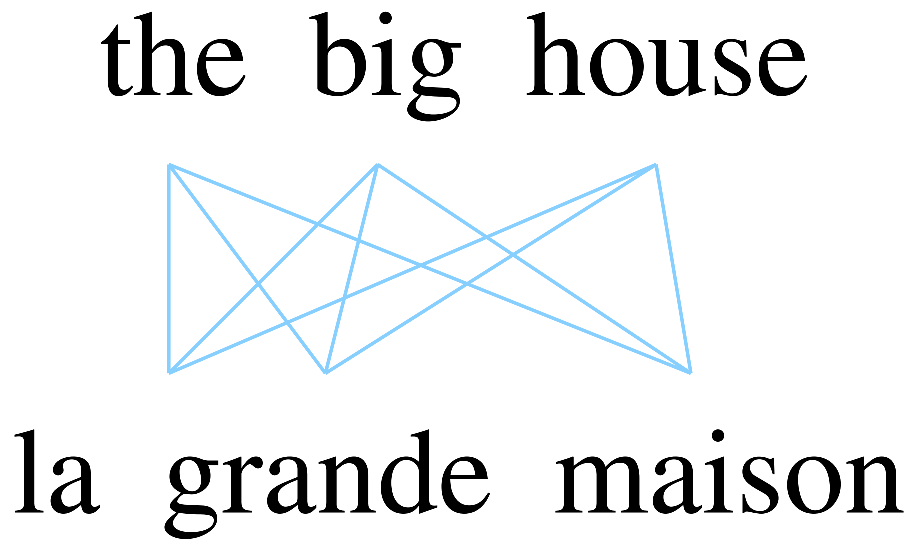


Cannot be resolved with a single example

Statistical machine translation

Data : aligned sentences

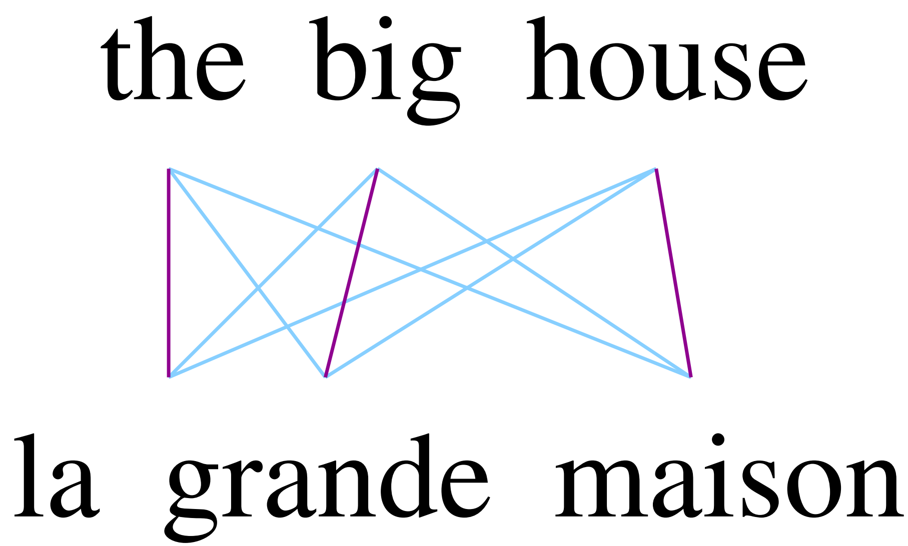
but word correspondences are unknown



Brown et.al.1993

Statistical machine translation

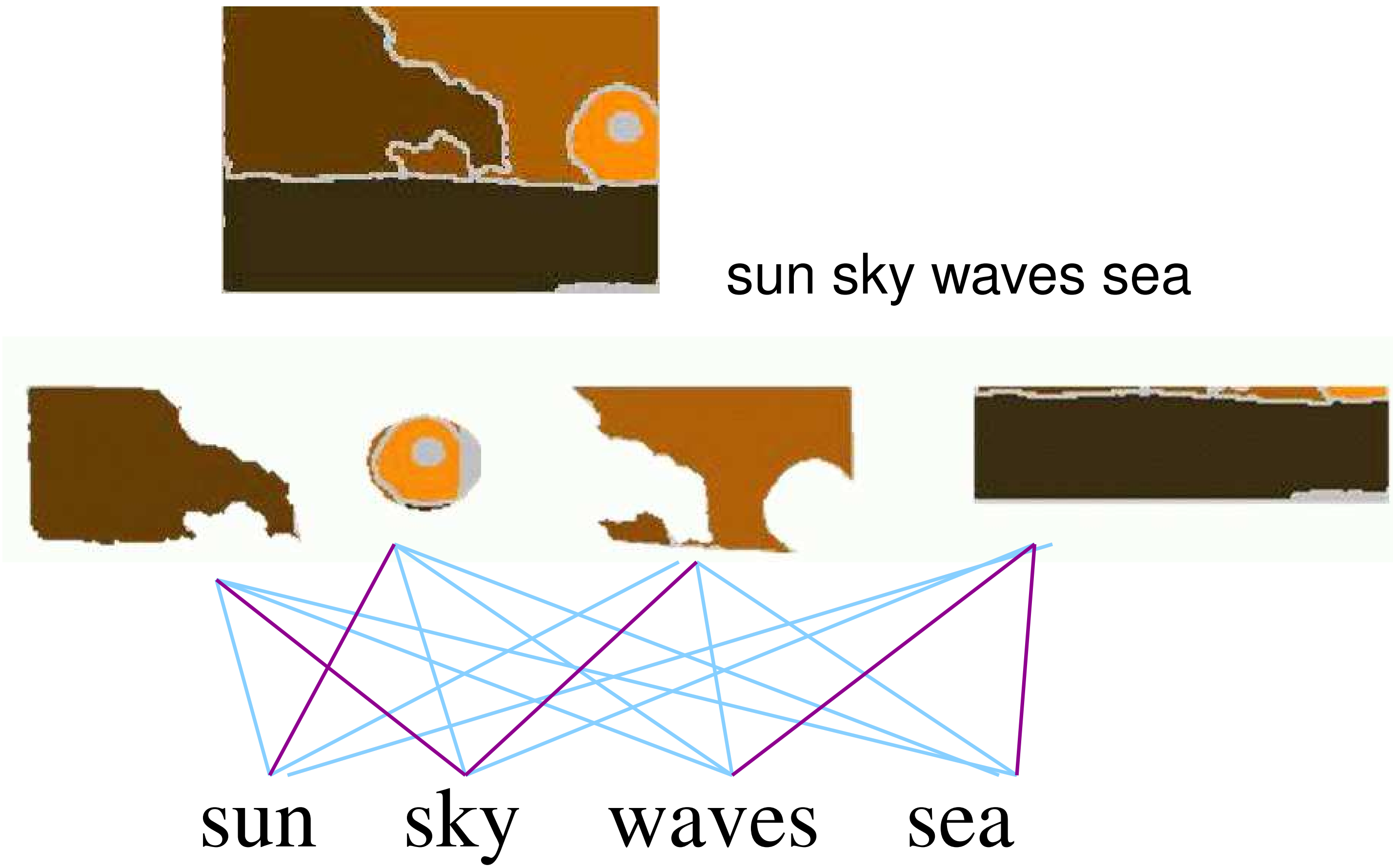
- Given the correspondences, we can estimate the translation $p(big \mid grande)$
- Given the probabilities, we can estimate the correspondences



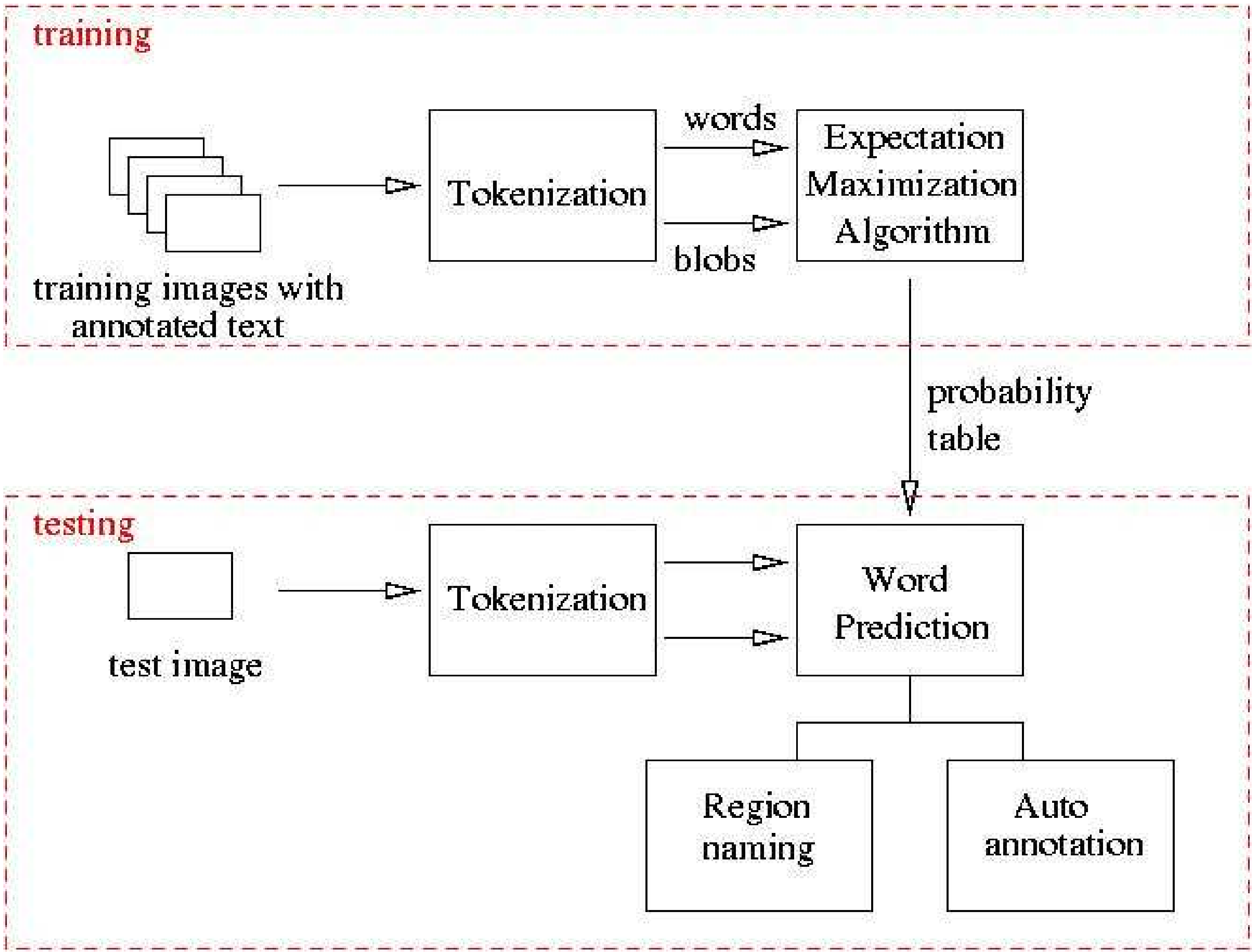
With enough data, it is possible to obtain the translation

$$p(big \mid grande) = 1$$

Multimedia translation



Overview of the system



Data

160 CDs from Corel data set (100 images in each)

10 experimental data sets
each:

- randomly selected 80 CDs
- 75% for training
- 25% for testing
- 150-200 words in the vocabulary

polar bears snow fight

sun tree plain sky

memorial flags grass

tiger cat water grass

Input representation

segmentation*

sun sky waves sea

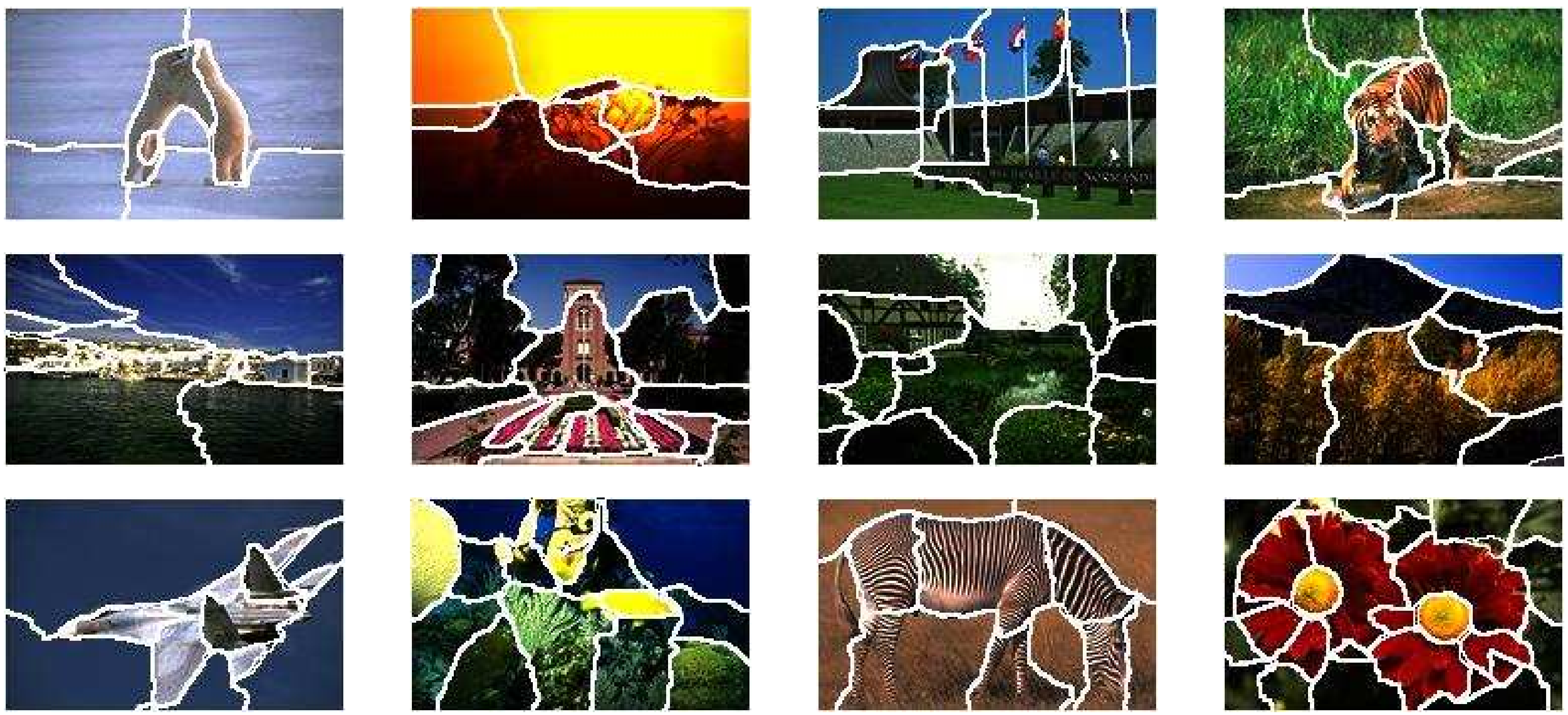
Each region is a large vector of features

- Region size
- Position
- Color
- Oriented energy (12 filters)
- Simple shape features

*Normalized-Cuts is used for segmentation.(thanks to Shi, Tal and Malik).

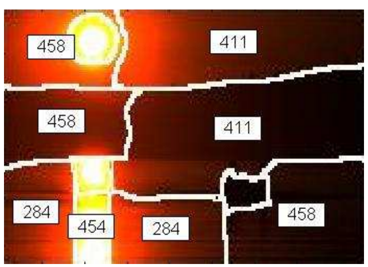
The process took a month.

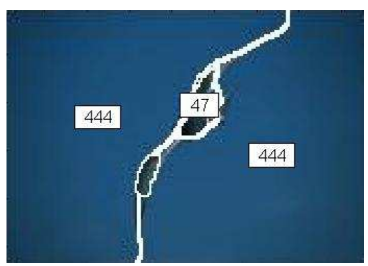
Sample segmentation results

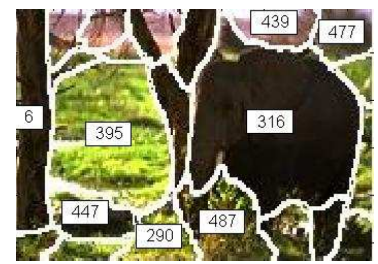


Tokenization

- Words in the vocabulary $\rightarrow$ word tokens
- Image regions
  - represented by 30 features (size, position, color, texture, shape)
  - feature space is clustered for grouping the region types
  - each region $\rightarrow$ closest region type $\rightarrow$ blob tokens







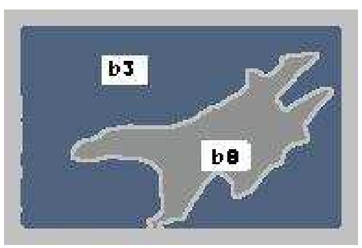
Tokenization

plane jet su-27 sky

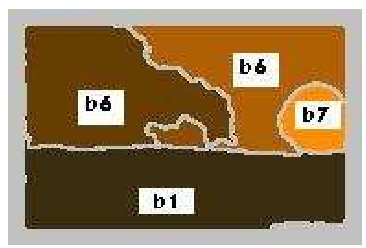
sun sea waves sky

grass tiger cat forest

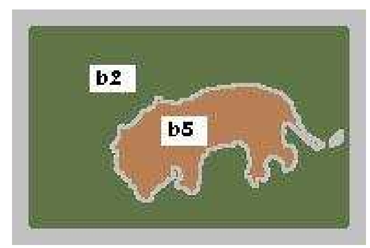
headland grass sky



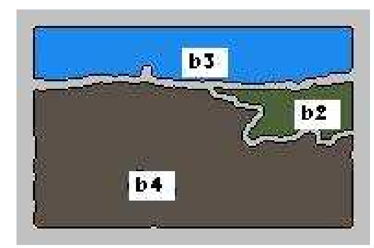
w3 w4 w5 w1



w6 w7 w8 w1

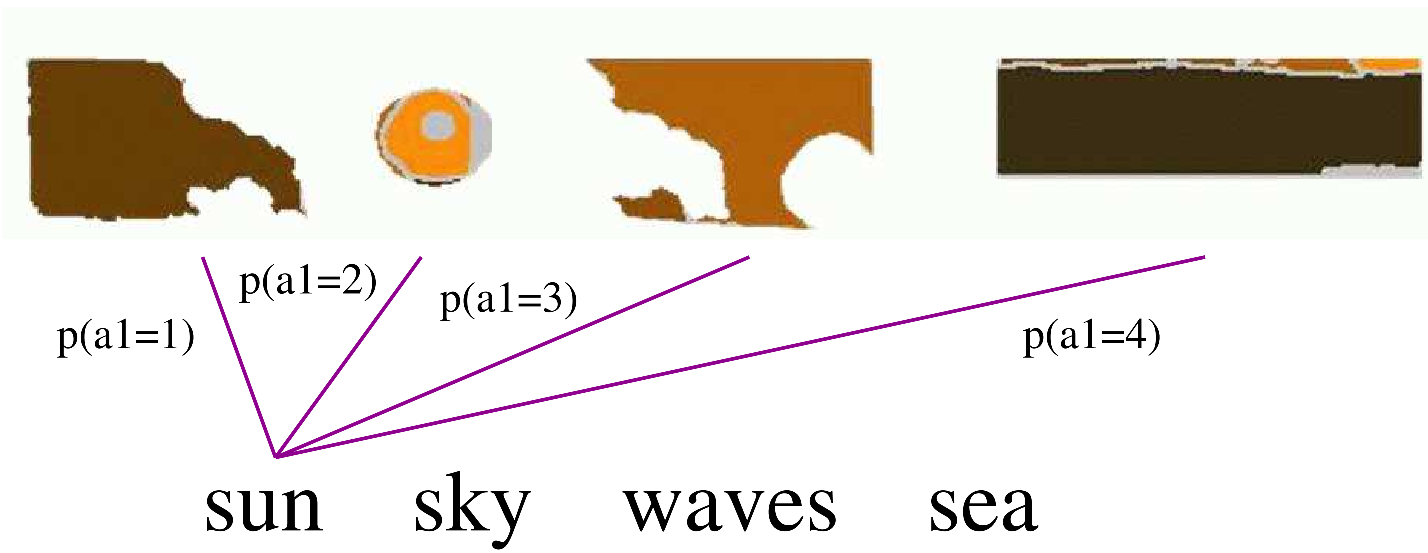


w2 w9 w10 w11

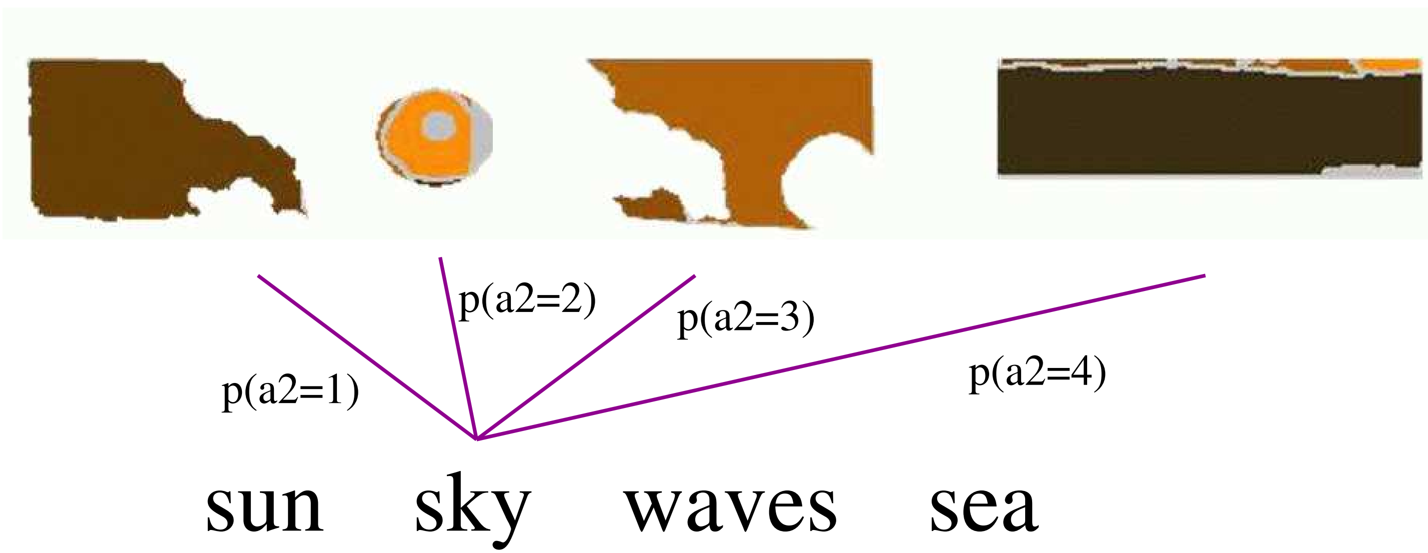


w12 w2 w1

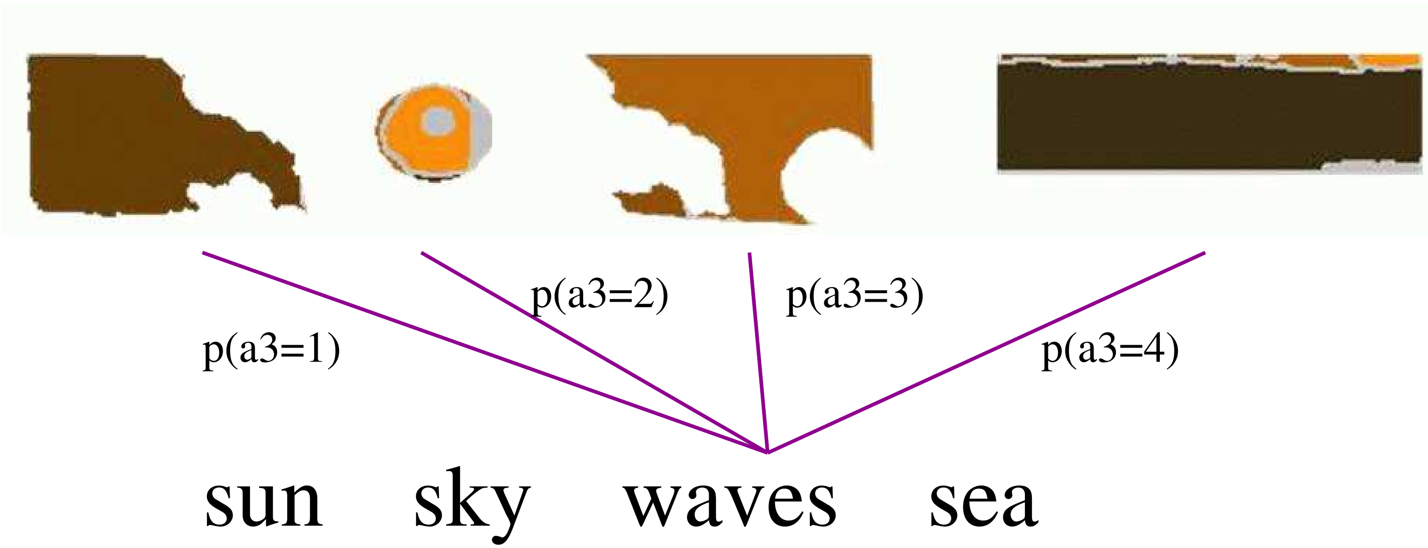
Assignments



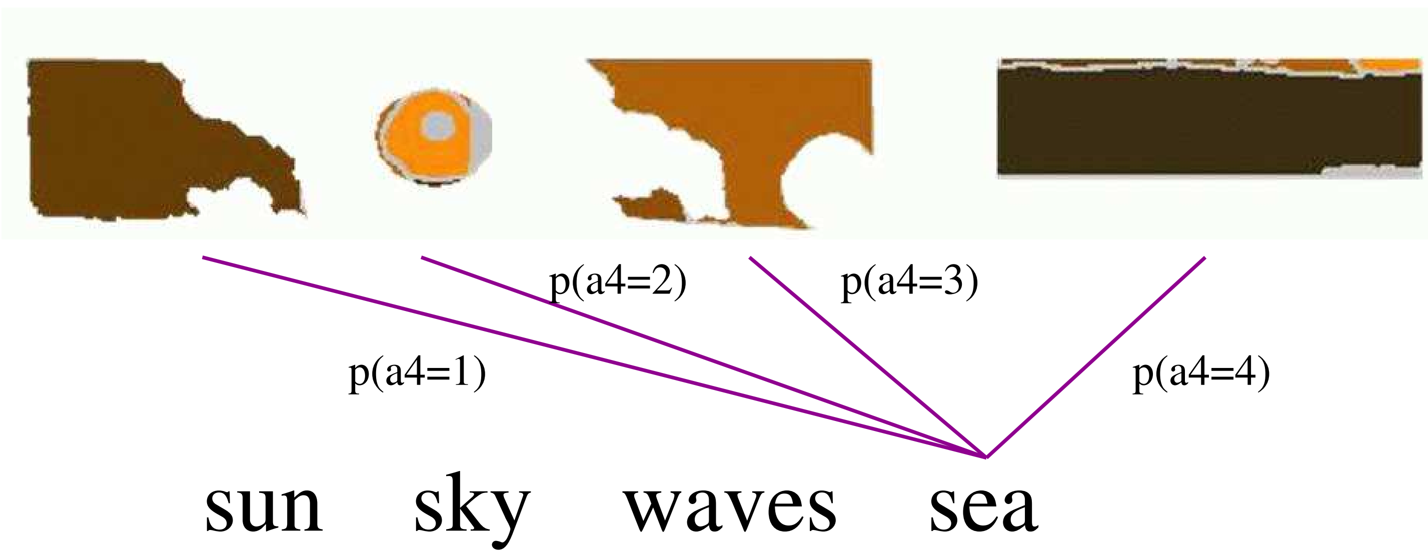
Assignments



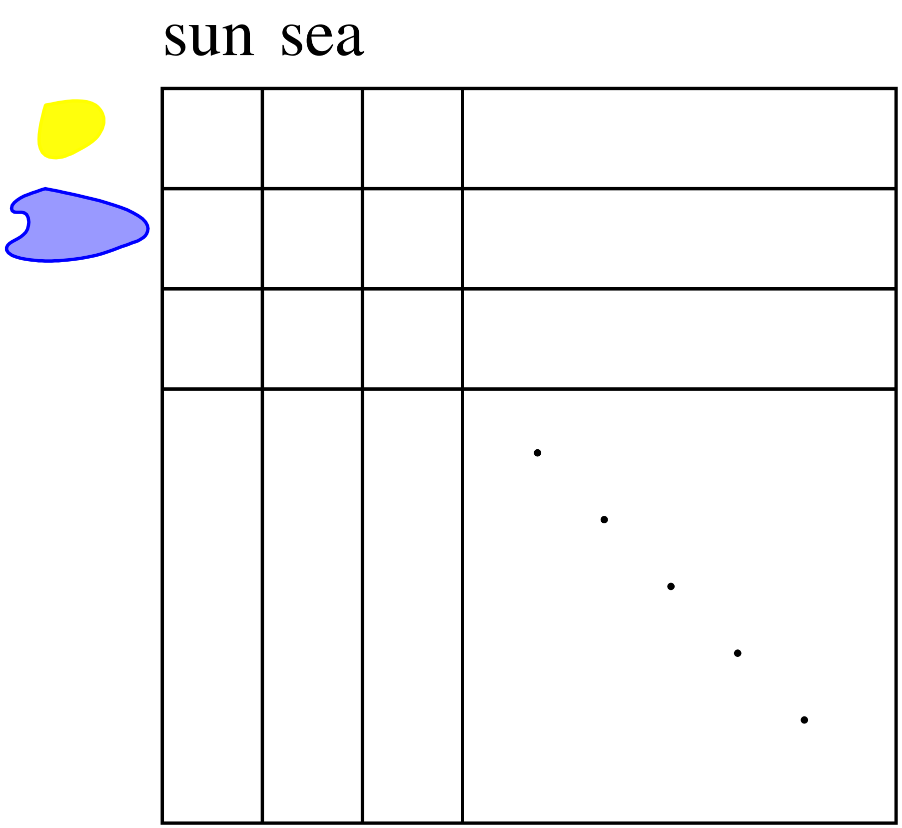
Assignments



Assignments



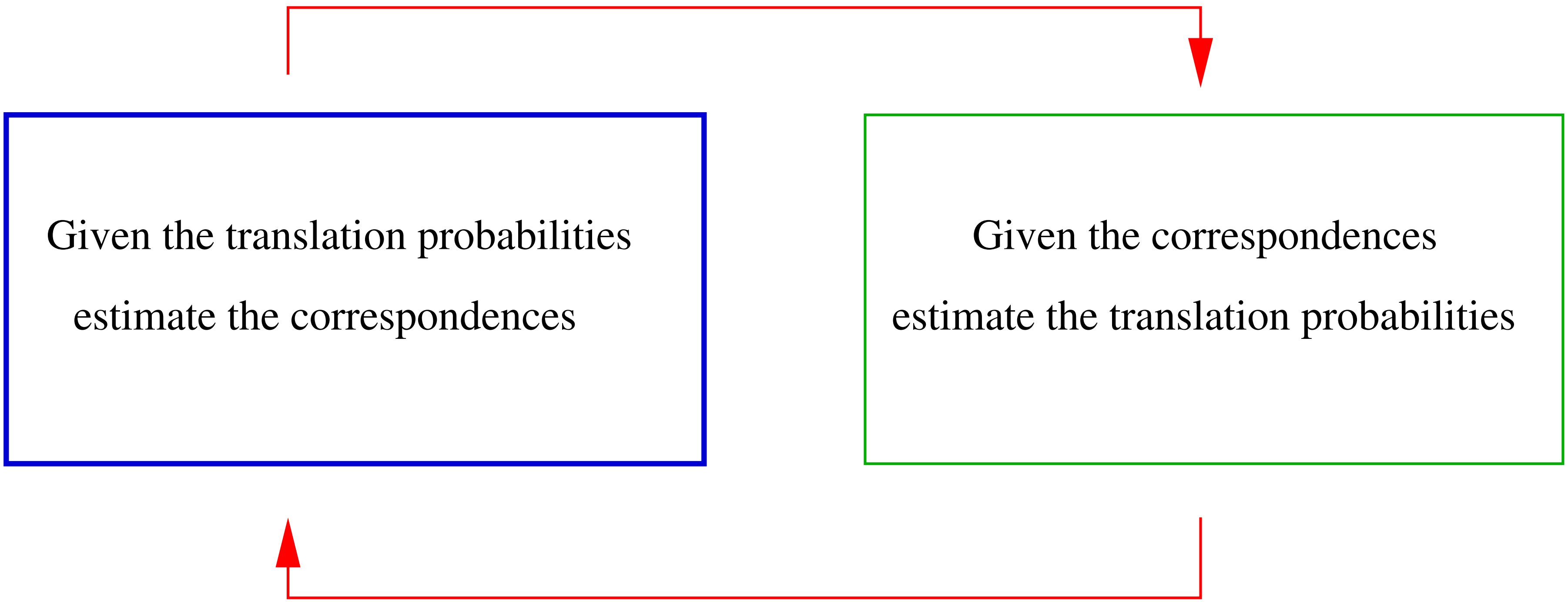
Initialization



Initialize translation table to blob-word co-occurrences (empirical distributions of words and blobs)

Rough estimate for the translation table

Expectation Maximization Algorithm



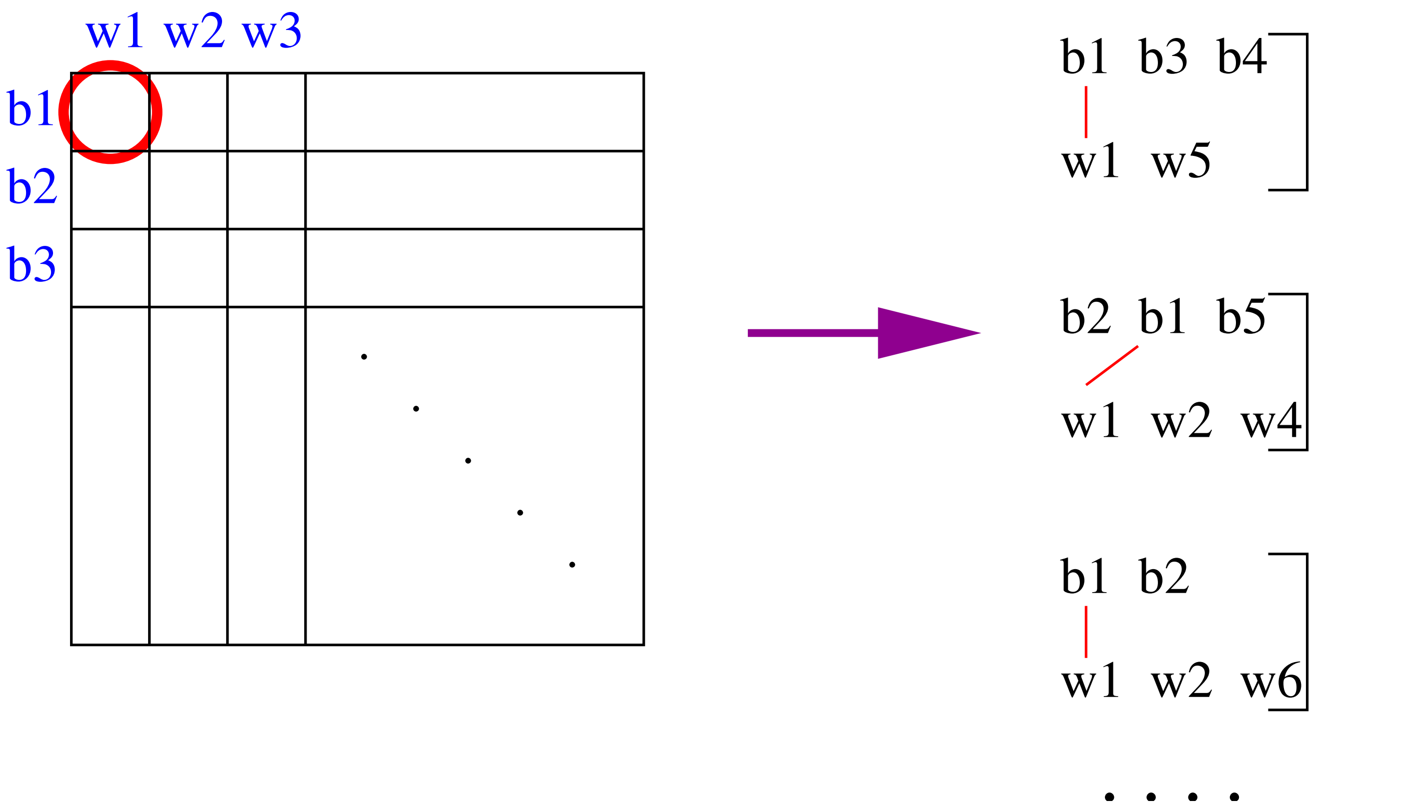
Dempster et.al

Expectation Maximization Algorithm

E step : (for one pair)

predict correspondences from translation probabilities

translation probabilities correspondences

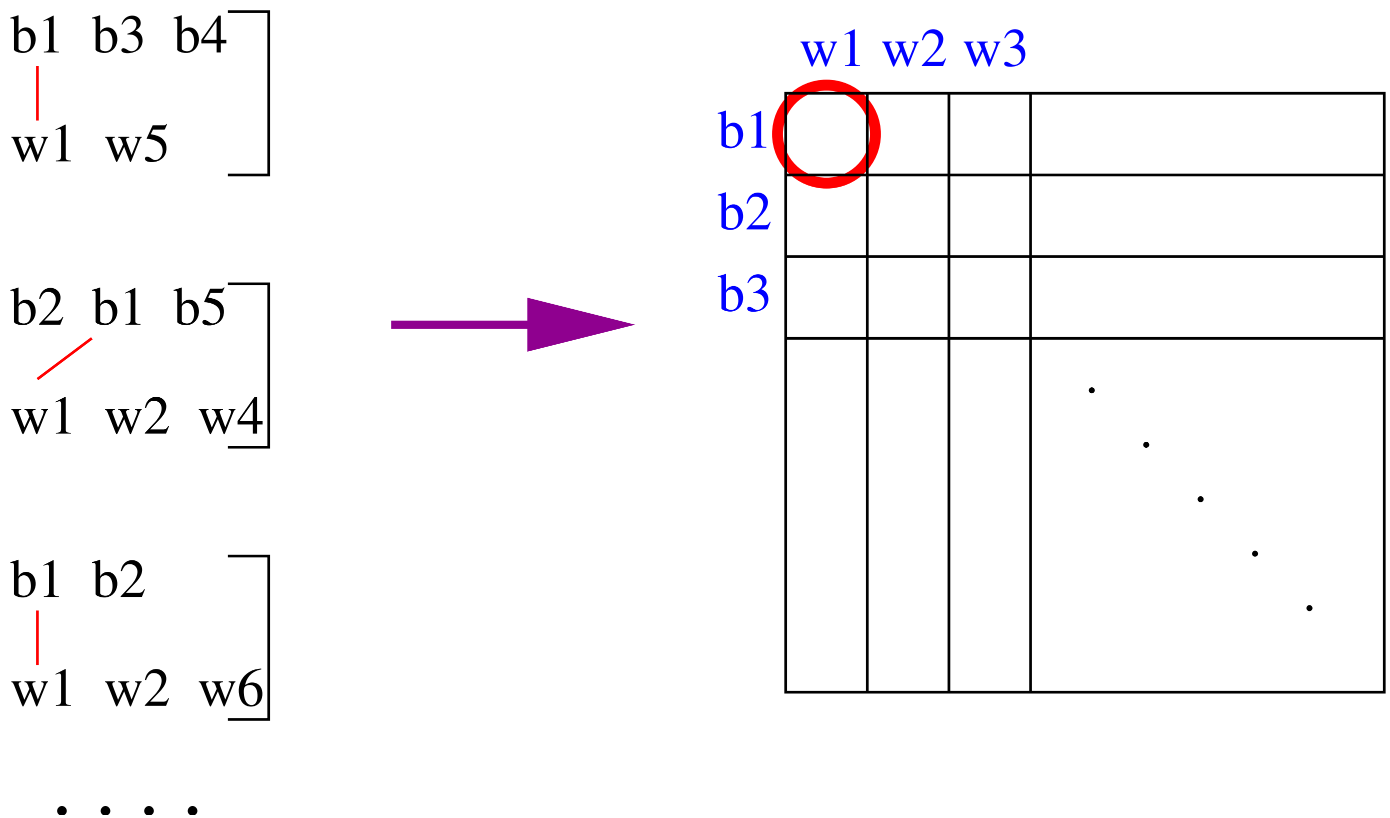


Expectation Maximization Algorithm

M step : (for one pair)

predict translation probabilities from correspondences

correspondences translation probabilities



EM formulation

Maximize

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

$$Q^{\text{ML}} = \sum_{n=1}^{N} \sum_{j=1}^{M_n} \sum_{i=1}^{L_n} p(a_{nj} = i \mid w_{nj}, b_{ni}, \theta^{(\text{old})})$$
$$\log \left[p(a_{nj} = i) t(w = w_{nj} \mid b = b_{(a_{nj}=i)}) \right].$$

with respect to the constraints :

$\sum_i p(a_{nj} = i) = 1$ and $\sum_{w^\star} t(w^\star \mid b^\star) = 1$.

EM formulation

E step:

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

$$\widetilde{p}(a_{nj} = i \mid w_{nj}, b_{ni}, \theta^{(\text{old})}) = p(a_{nj} = i) t(w_{nj} \mid b_{ni})$$

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

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

EM formulation

M step:

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

$$\widetilde{t}(w_{nj} = w^\star \mid b_{ni} = b^\star) = \sum_{n=1}^{N} \sum_{j=1}^{M_n} \sum_{i=1}^{L_n} p(a_{nj} = i \mid w_{nj}, b_{ni}, \theta^{(\text{old})}) \delta_{(w^\star, b^\star)}(w_{nj}, b_{ni})$$

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

2. Normalize $\widetilde{t}(w_{nj} = w^\star \mid b_{ni} = b^\star)$ to obtain $t(w_{nj} = w^\star \mid b_{ni} = b^\star)$.

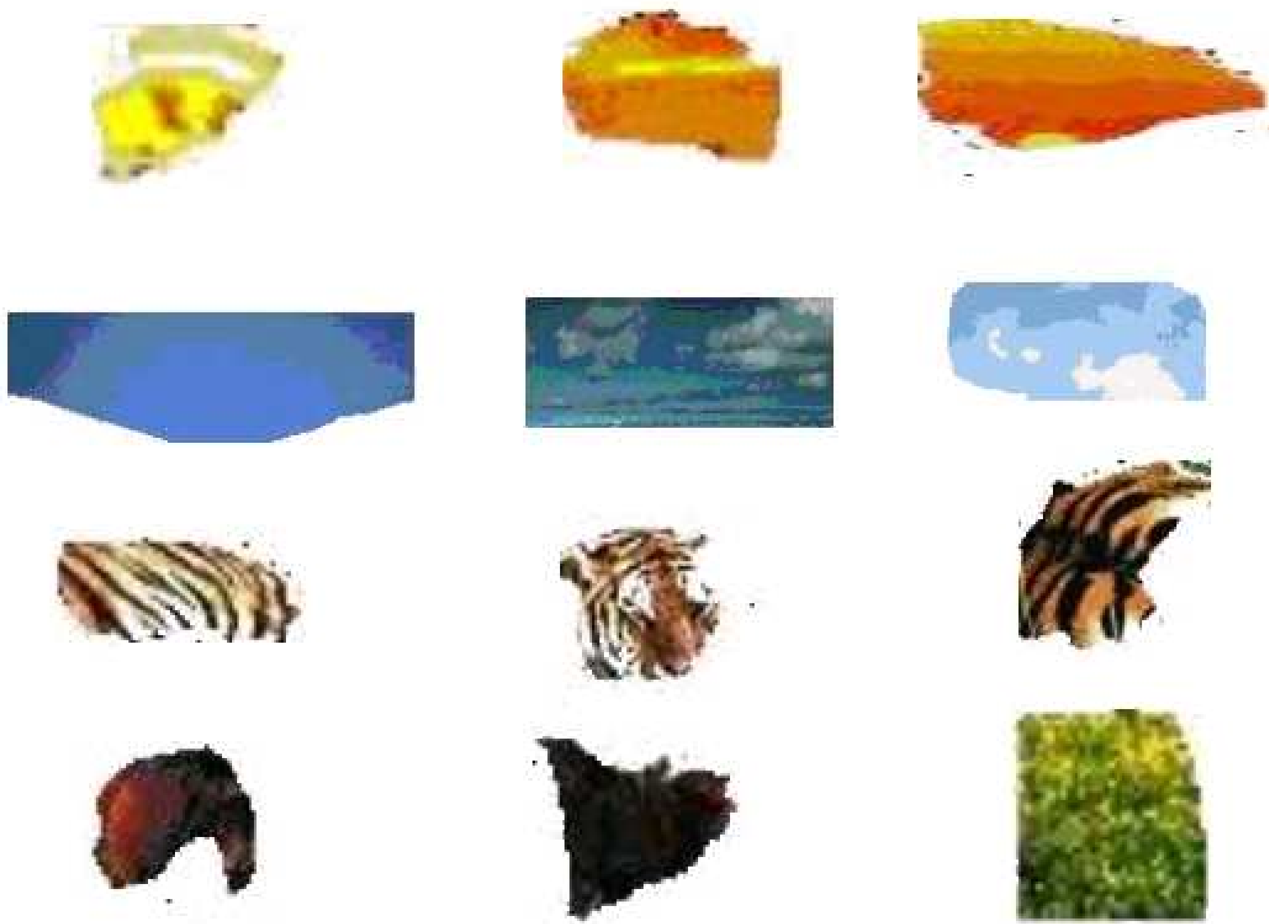
Dictionary

sun

sky

cat

horse



Word Prediction

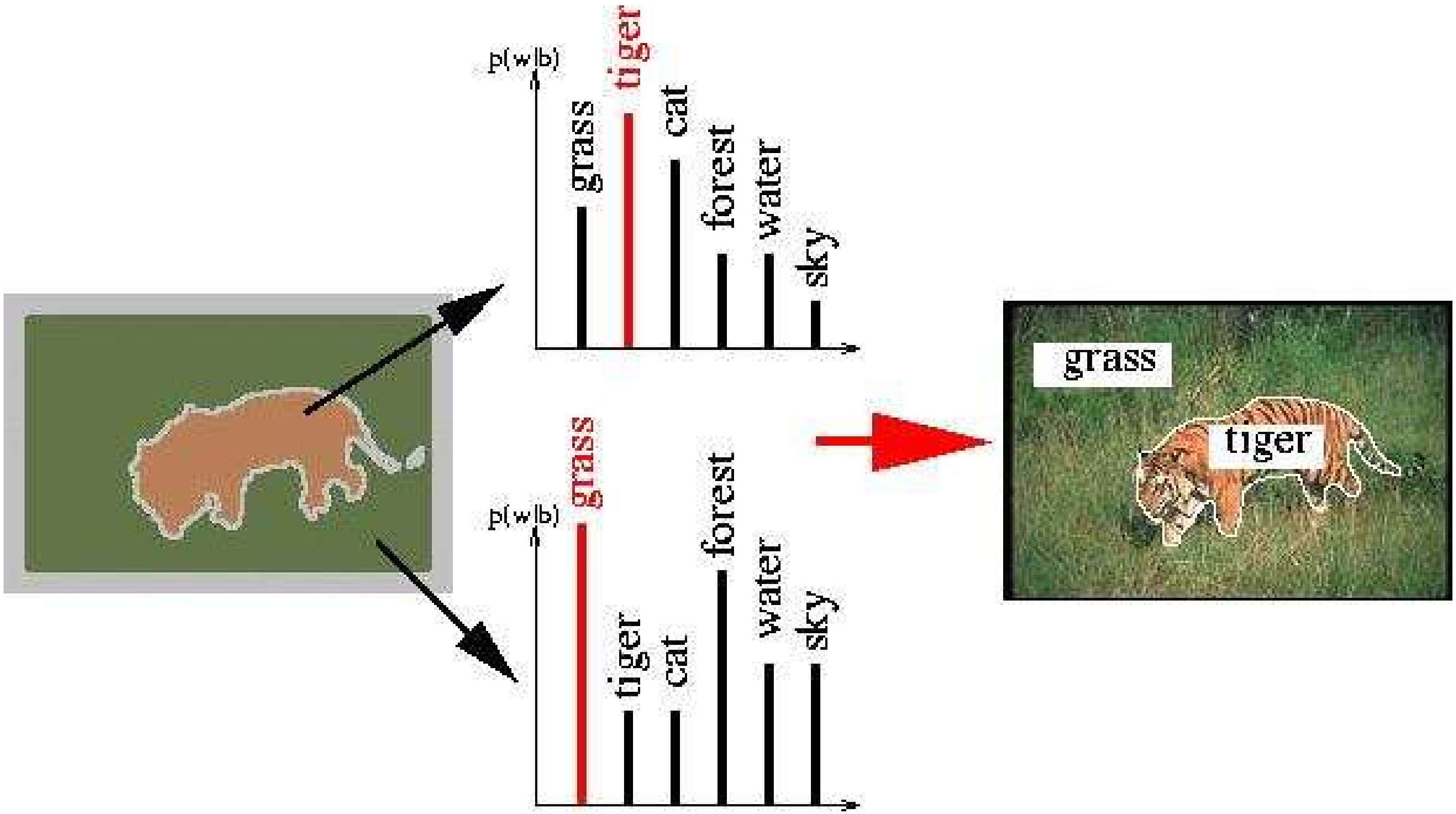
On a new test image

- segment the image
- extract the features from the regions
- then, for each region
  - find the corresponding blob token b using the nearest neighbor method
  - use the word posterior probabilities $p(w \mid b)$ to predict words

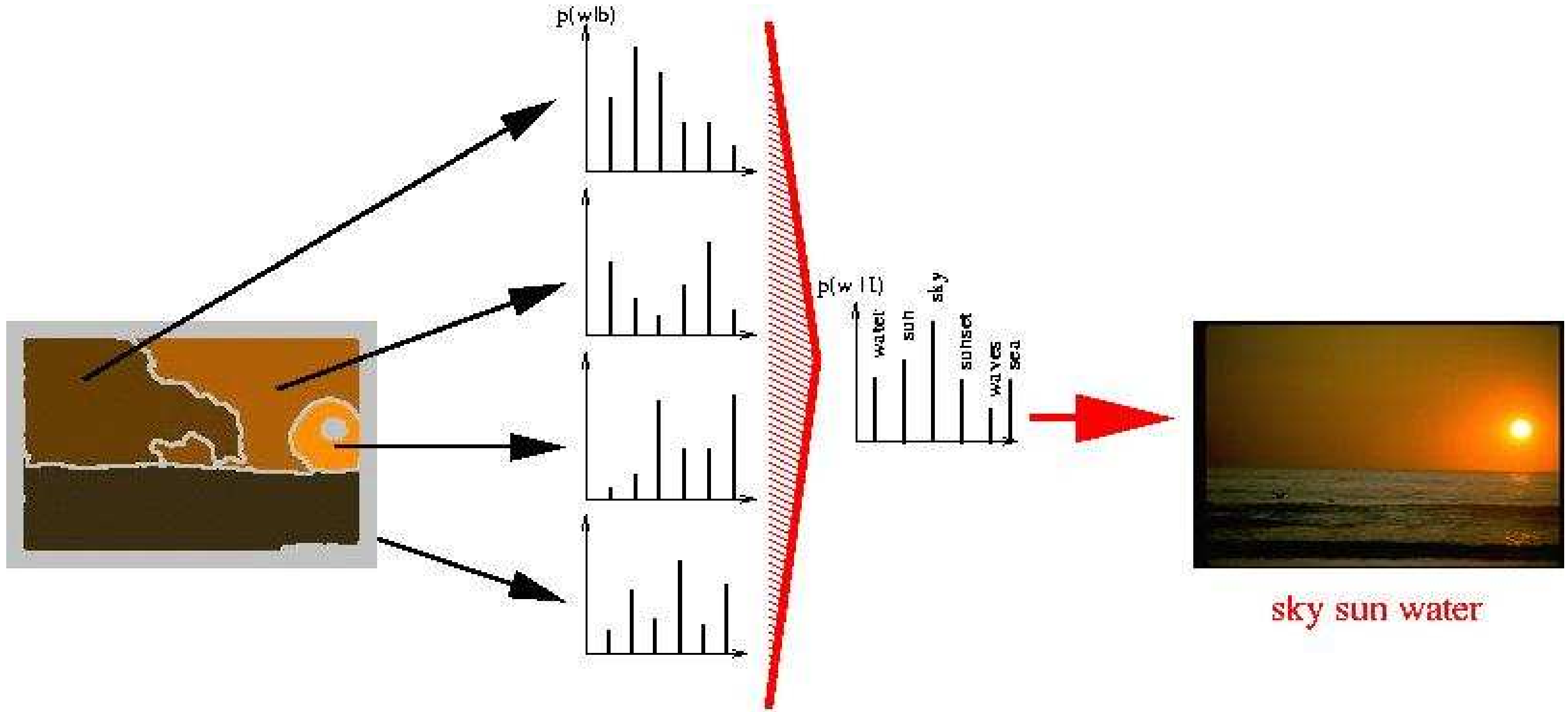
use predicted words

- for region naming
- for auto-annotation

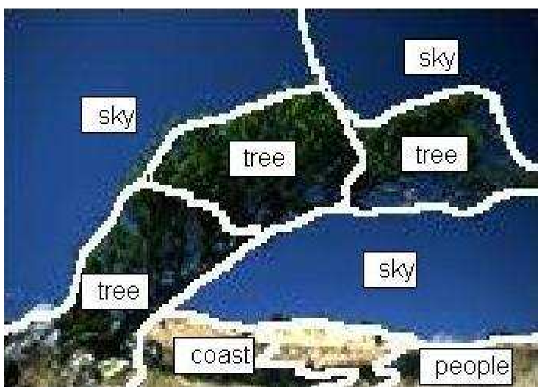
Region Naming

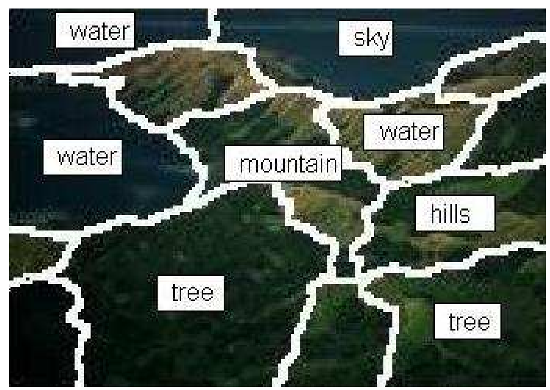


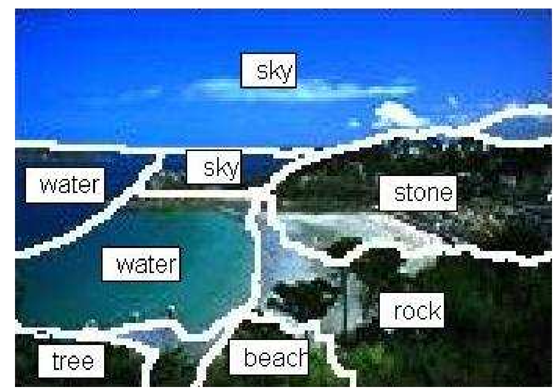
Auto-annotation



hills sky tree mountain tree water beach sky tree water



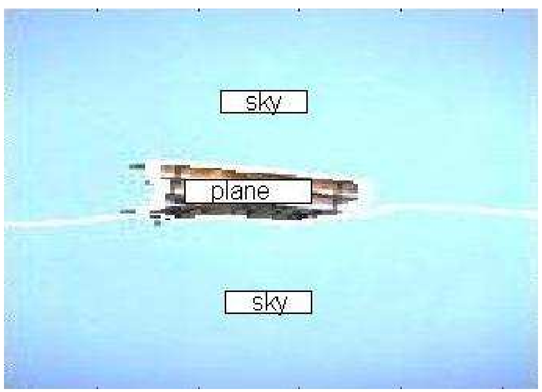


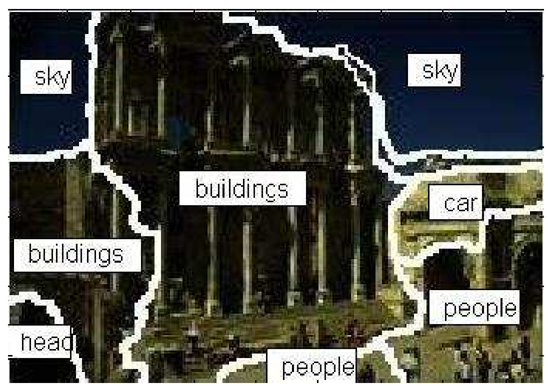


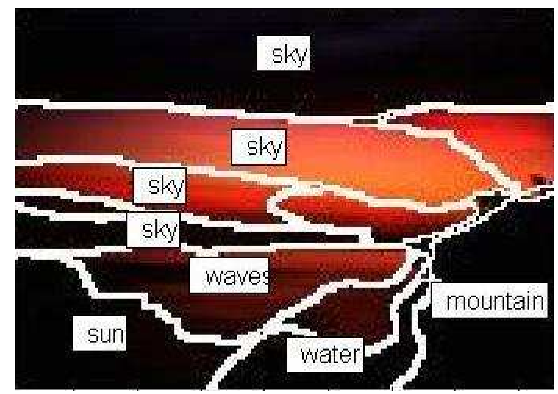
plane sky

people ruins stone

sunset tree water







Measuring the performance

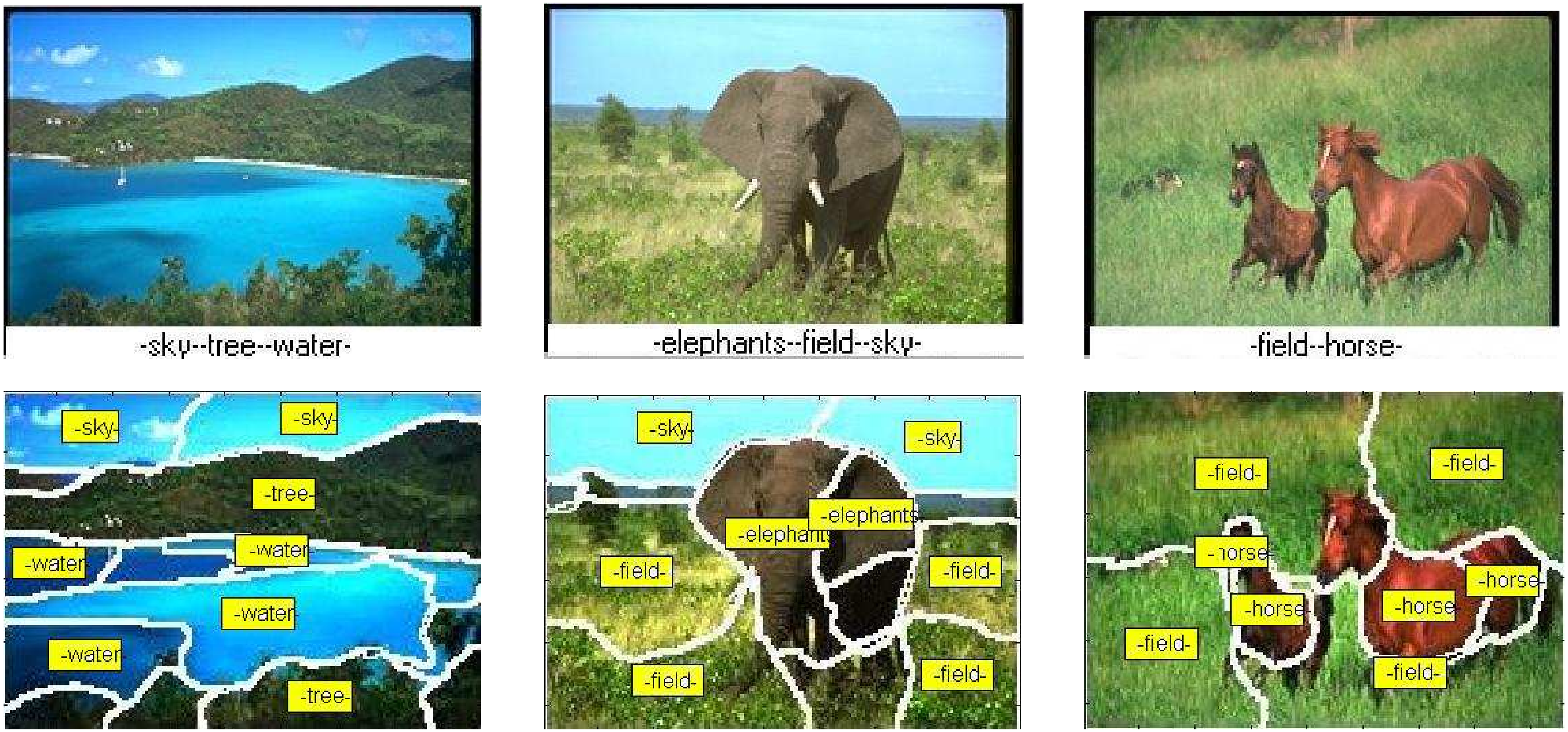
- Visually inspecting the images
- Using a hand-labeled data for scoring the correspondences
- Using annotation performance as a proxy

Visually inspecting the images

- Do we predict the right words ?
- are they on the right place?

Visual inspection answers both of the questions, but it is not possible to do for a large number of images

Using hand-labeled data



450 images are labeled manually, to evaluate correspondence performance

subjective and error prone

hard to do on a large number of images

Correspondence scores

word	num predicted	num labeled	num correct
water	459	229	92
sky	352	382	119
people	292	41	13
buildings	120	130	21
tree	430	230	65
grass	110	239	21
clouds	75	26	5
flowers	49	96	8
sea	4	3	2
windows	4	3	1

Measuring Annotation Performance

Actual keywords → grass tiger cat forest

Predicted words → cat horse grass water

Measuring Annotation Performance

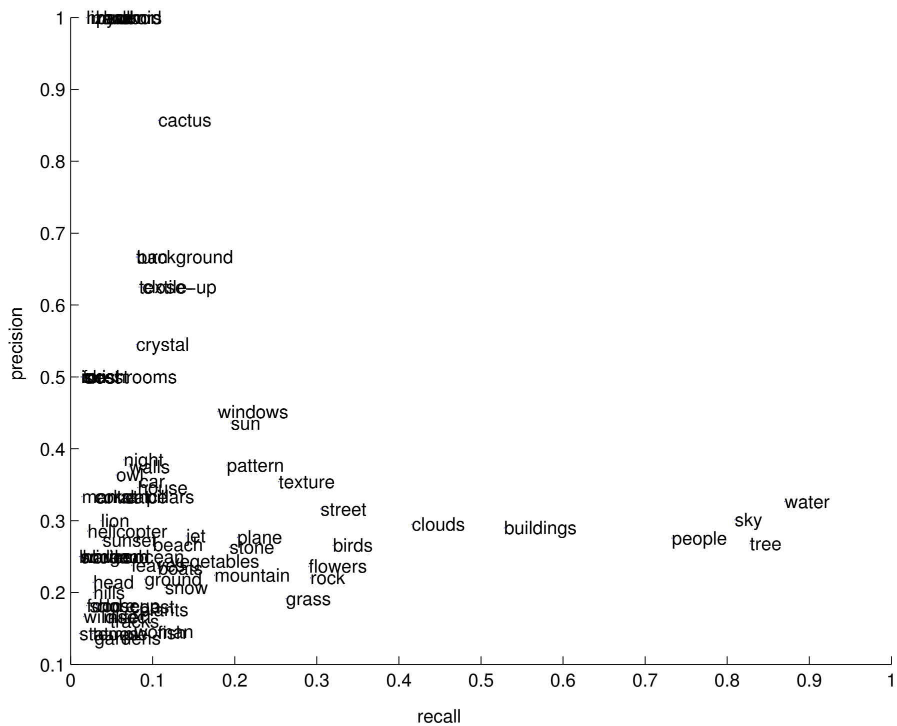
Actual keywords → grass tiger cat forest

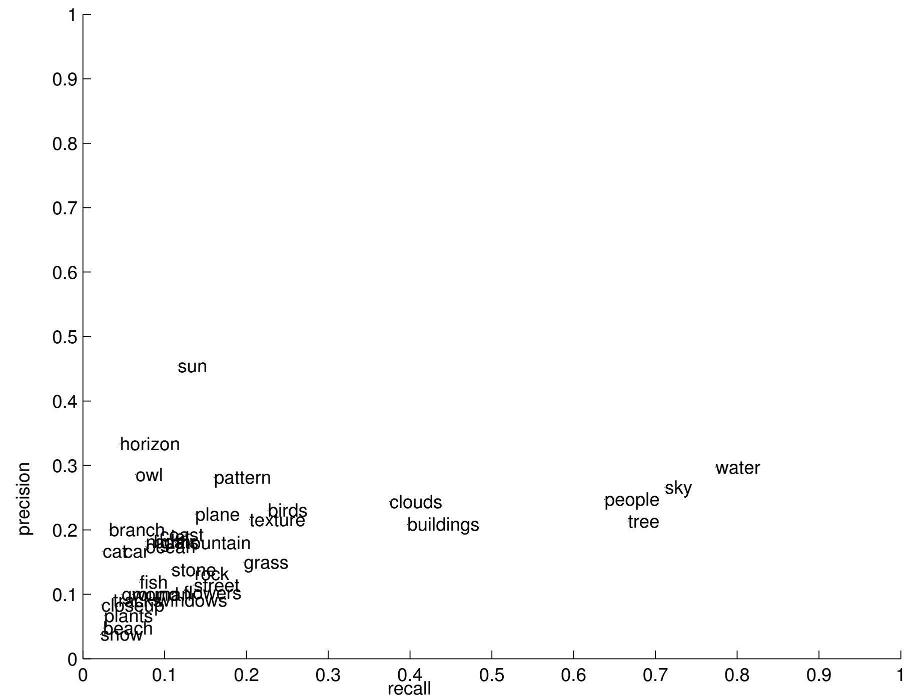
Predicted words → cat horse grass water

Prediction rates using annotation as a proxy

word	num pred.	num occur.	true pos.	false pos.	false neg.
water	1022	393	304	718	89
tree	946	303	202	744	101
sky	834	312	222	612	90
people	785	304	194	591	110
buildings	240	126	50	190	76
grass	167	127	25	142	102
clouds	160	104	39	121	65
boats	33	69	6	27	63
plane	49	80	11	38	69
sun	11	43	5	6	38
owl	7	31	2	5	29

Recall versus precision





Recall: number of correct predictions / number of actual occurrence

Precision : number of correct predictions / number of total predictions

76 words in training set and 36 words in standard test set have nonzero values (total number of words is 153)

Measuring Annotation Performance

- Kullback-Leibler divergence between the predicted and target distributions
- Word prediction measure
- Normalized classification score

Kullback-Leibler divergence

$$E_{KL} = \sum_w p(w) log \frac{p(w)}{p(w \mid B)}$$

- $p(w)$: target distribution
- $p(w \mid B)$: predicted distribution
- B : set of blobs in the image

Kullback-Leibler divergence

set	training	standard test	novel test
001	3.5602	5.2089	5.6769
002	3.4932	4.9387	4.3696
003	3.5322	4.9982	5.4598
004	3.6355	5.3491	5.7723
005	3.5123	5.0050	5.5352
006	3.5206	5.1052	5.9007
007	3.7002	5.2544	4.3680
008	3.5643	5.1617	5.5048
009	3.6573	5.2011	4.4484
010	3.4594	4.9578	5.4725

Word prediction measure

$$E_{PR} = r/n$$

- n : number of actual words in the image
- r : number of words predicted correctly
- the number of predicted words (r+w) is set to the number of actual keywords

Word prediction measure

set	training	standard test	novel test
001	0.2708	0.2171	0.2236
002	0.2799	0.2262	0.2173
003	0.2763	0.2288	0.2095
004	0.2592	0.1925	0.2172
005	0.2853	0.2370	0.2059
006	0.2776	0.2198	0.2163
007	0.2632	0.2036	0.2217
008	0.2799	0.2363	0.2102
009	0.2659	0.2223	0.2114
010	0.2815	0.2297	0.1991

Normalized classification score

$$E_{NS} = r/n - w/(N - n)$$

- N : vocabulary size
- n : number of actual words in the image
- r : number of words predicted correctly
- the number of predicted words (r+w) is set to the number of actual keywords

Normalized classification score

set	training	standard test	novel test
001	0.2560	0.2012	0.2102
002	0.2657	0.2111	0.2053
003	0.2616	0.2129	0.1968
004	0.2449	0.1771	0.2048
005	0.2713	0.2222	0.1933
006	0.2636	0.2046	0.2037
007	0.2501	0.1895	0.2097
008	0.2664	0.2220	0.1978
009	0.2527	0.2082	0.1990
010	0.2659	0.2131	0.1854

Evaluating the results

Compare the results of the proposed method with

- Empirical word densities
- Co-occurrences of words and blobs

Comparing with the empirical word densities

- Predict the most common words for all the images in the set.
- Then use the prediction rates as a baseline for evaluating the performance of the proposed method.

	KL	NS	PR
training	4.8458 - 3.5635	0.1732 - 0.2598	0.1894 - 0.2740
standard test	4.8416 - 5.1180	0.1754 - 0.2062	0.1914 - 0.2211

Comparing with the empirical word densities

Recall and precision when empirical word densities are used :

	'water'	'sky'	'tree'	'people'
training	1.000 - 0.217	0.993 - 0.190	0.893 - 0.208	0.366 - 0.168
std. test	1.000 - 0.225	0.994 - 0.187	0.894 - 0.205	0.349 - 0.176

Recall and precision when the proposed method is used :

	'water'	'sky'	'tree'	'people'
training	0.870 - 0.326	0.809 - 0.301	0.827 - 0.268	0.733 - 0.276
std. test	0.774 - 0.297	0.712 - 0.266	0.667 - 0.214	0.638 - 0.247

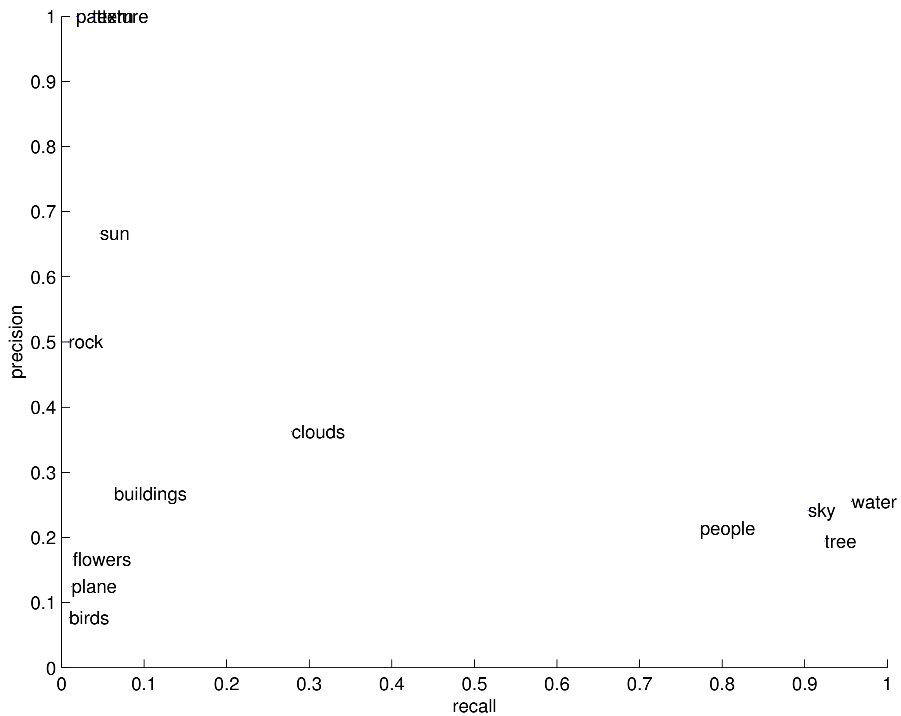
Comparing with co-occurrences

Use the co-occurrence of words and blobs in the data, as the translation probability table

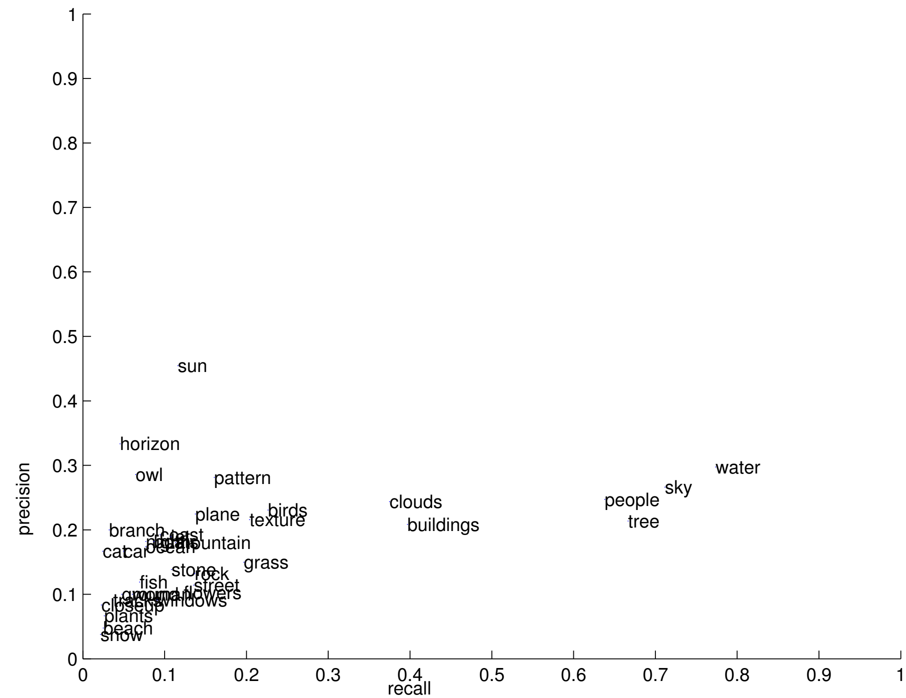
	KL	NS	PR
training	4.0427 - 3.5635	0.2200 - 0.2598	0.2350 - 0.2740
standard test	4.5428 - 5.1180	0.2048 - 0.2062	0.2199 - 0.2211

Comparing with co-occurrences

Recall versus precision values on the standard test set:



using co-occurrences



using proposed method

Improving the system

- Refusing to predict
- Retraining on refined vocabulary
- Merging indistinguishable words

Refusing to predict

Null and fertility problems

simple solution to null - refusing to predict

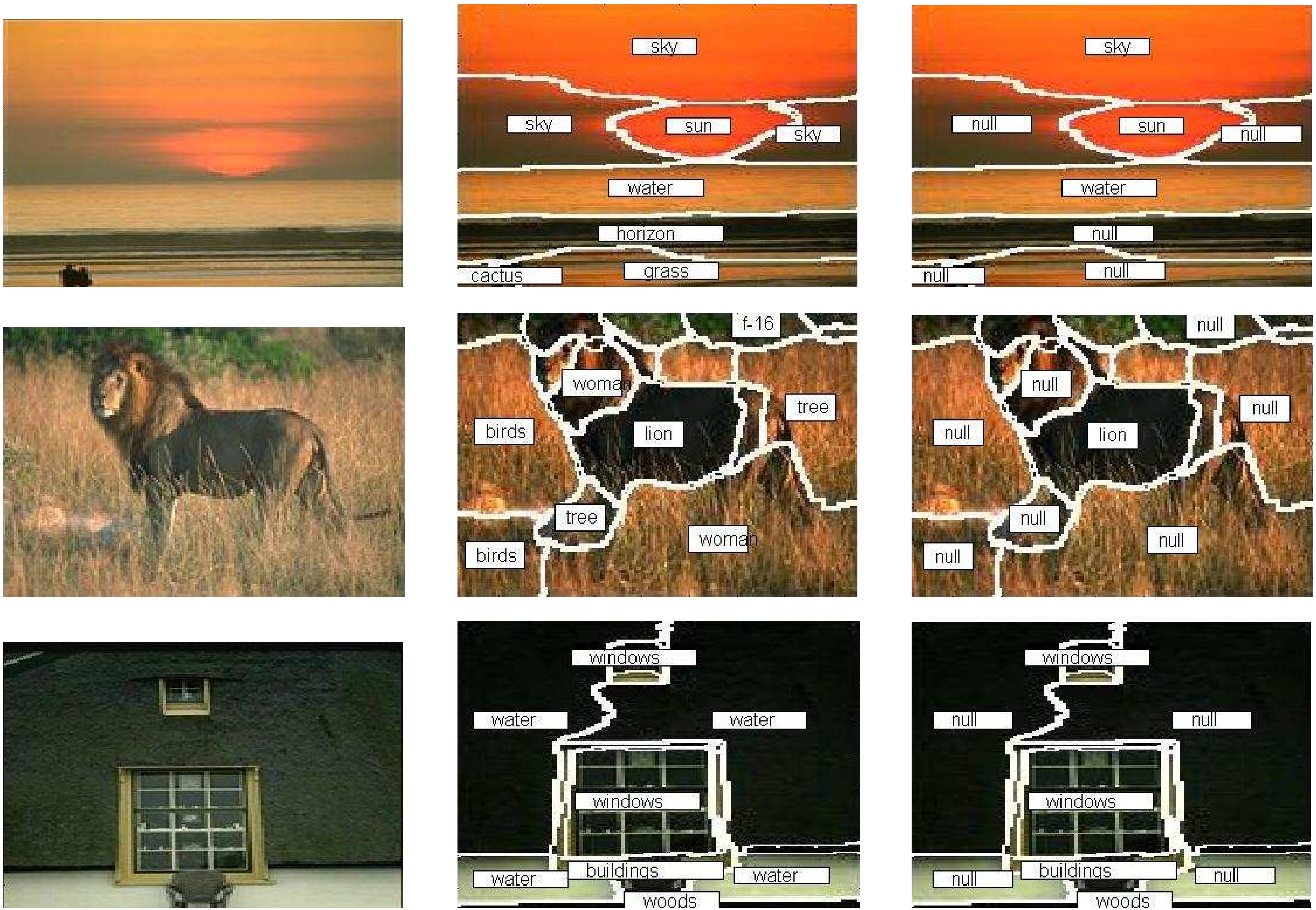
if $prob(word \mid blob) > threshold$ then

predict the word

else

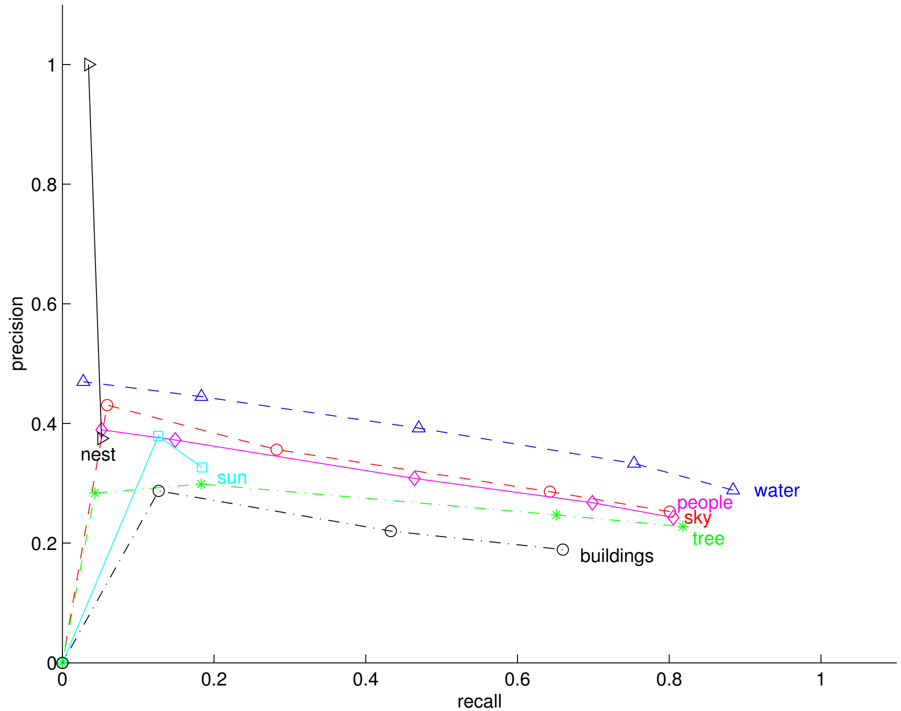
assign NULL

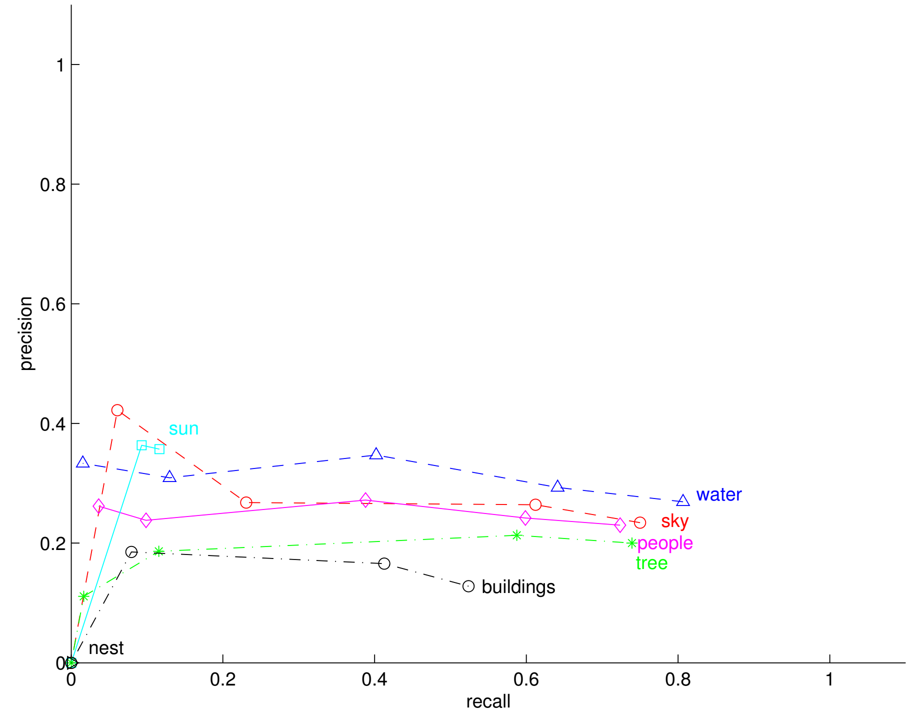
NULL prediction



Effect of NULL threshold

Recall versus precision as a function of NULL threshold :





Retraining on a refined vocabulary

To refine the vocabulary

- choose a threshold,
- allow only the words which have higher prediction probabilities

	num words	KL	NS	PR
original	153	3.5602	0.2560	0.2708
> 0.0	86	3.3132	0.2820	0.2936
> 0.1	80	3.2878	0.2853	0.2966
> 0.2	65	3.1968	0.2950	0.3054
> 0.3	41	2.9685	0.3235	0.3320

Retraining on a refined vocabulary

Prediction probabilities :

word	org	> 0.0	> 0.1	> 0.2	> 0.3
grass	0.241	0.303	0.306	0.341	0.400
water	0.545	0.653	0.657	0.701	0.836
sun	0.321	0.396	0.400	0.443	0.479
sky	0.408	0.492	0.500	0.522	0.578
plane	0.300	0.358	0.362	0.392	0.350
texture	0.222	0.302	0.302	0.314	0.392
nest	0.590	0.619	0.621	0.633	0.679
fish	0.270	0.318	0.320	0.406	0.476
church	0.155	0.180	0.191	0.000	0.000

Retraining on a refined vocabulary

Recall and precision values :

word	org	> 0.0	> 0.1	> 0.2	> 0.3
grass	0.407-0.134	0.442-0.138	0.451-0.138	0.484-0.139	0.475-0.145
water	0.884-0.289	0.875-0.294	0.875-0.295	0.885-0.294	0.902-0.304
sun	0.184-0.327	0.184-0.327	0.184-0.327	0.184-0.333	0.184-0.348
sky	0.801-0.253	0.793-0.259	0.788-0.261	0.786-0.265	0.797-0.270
plane	0.191-0.142	0.191-0.144	0.216-0.141	0.266-0.134	0.203-0.134
texture	0.363-0.128	0.363-0.130	0.363-0.130	0.363-0.131	0.403-0.133
nest	0.052-0.375	0.052-0.375	0.052-0.375	0.052-0.429	0.052-0.500
fish	0.374-0.098	0.441-0.102	0.441-0.102	0.380-0.102	0.346-0.114
church	0.075-0.080	0.075-0.080	0.075-0.080	0.000-0.000	0.000-0.000

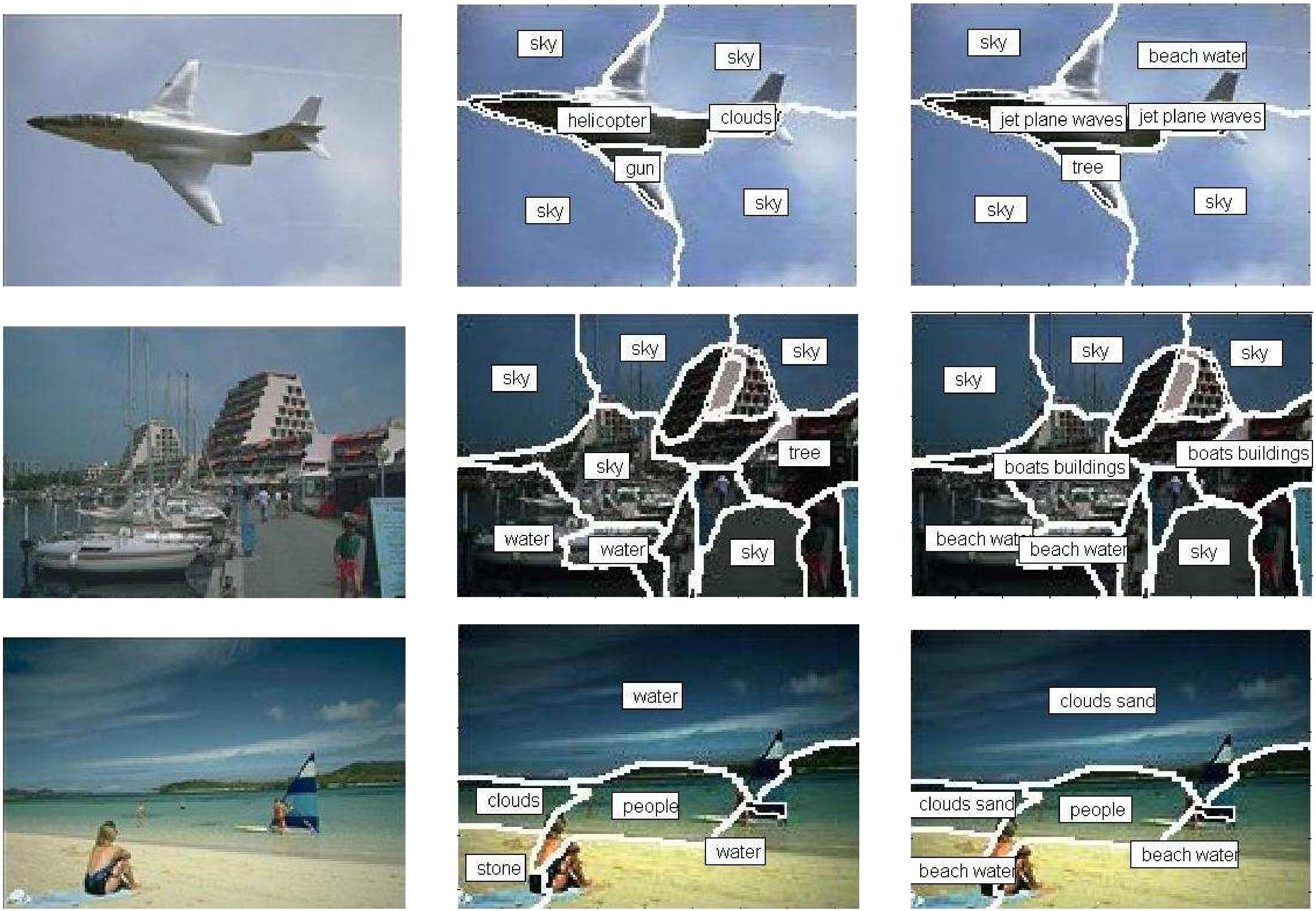
Merging indistinguishable words

Some words cannot be set apart

- either they are synonyms
  (e.g. locomotive and train)
- or they are indistinguishable using the current feature set
  (e.g. eagle and jet)

construct a similarity matrix based on the posterior probabilities $p(b \mid w)$
then, use a graph cut algorithm for clustering

Merging indistinguishable words



Merging indistinguishable words

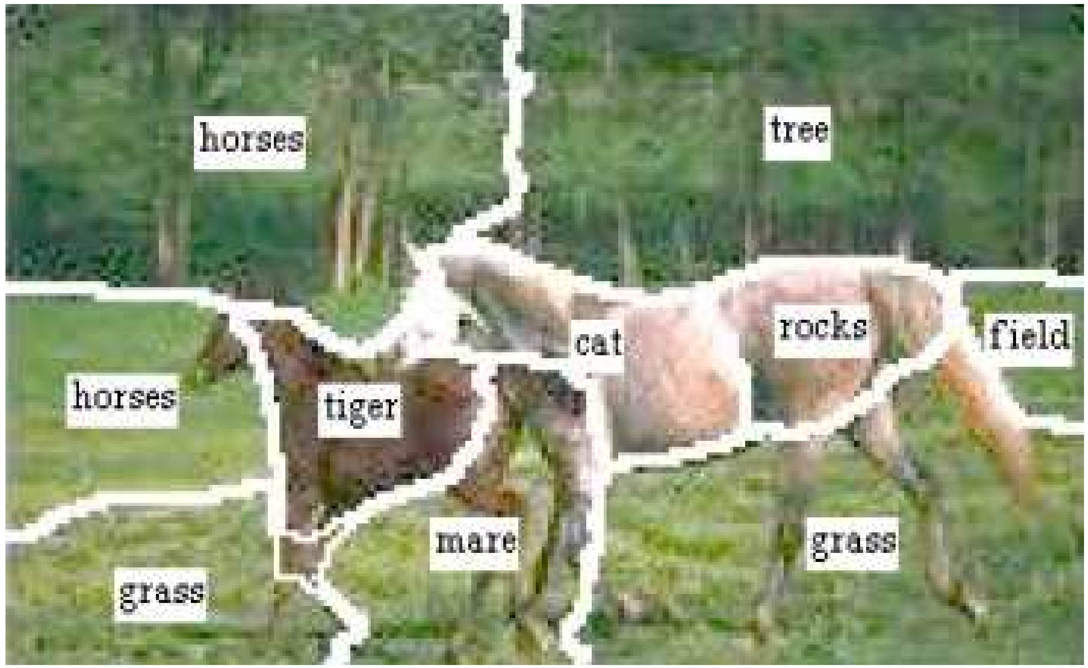
	original	merged
NS - standard test	0.2012	0.2242
PR - standard test	0.2171	0.2395
NS - training	0.2708	0.2490
PR - training	0.2560	0.2616

Merging indistinguishable words

Recall and precision values:

water beach	0.870 - 0.326 0.025 - 0.047	beach - water	0.988 - 0.333
coral ocean	0.000 - 0.000 0.077 - 0.173	coral - ocean	0.086 - 0.120
jet plane waves	0.107 - 0.189 0.137 - 0.224 0.000 - 0.000	jet plane waves	0.303 - 0.199
plants leaves	0.026 - 0.067 0.000 - 0.000	leaves plants	0.125 - 0.125
boats buildings	0.087 - 0.181 0.397 - 0.208	boats buildings	0.482 - 0.255

Integrating supervised data



a small amount of supervised data can be helpful

- for breaking symmetries
- for a better clustering

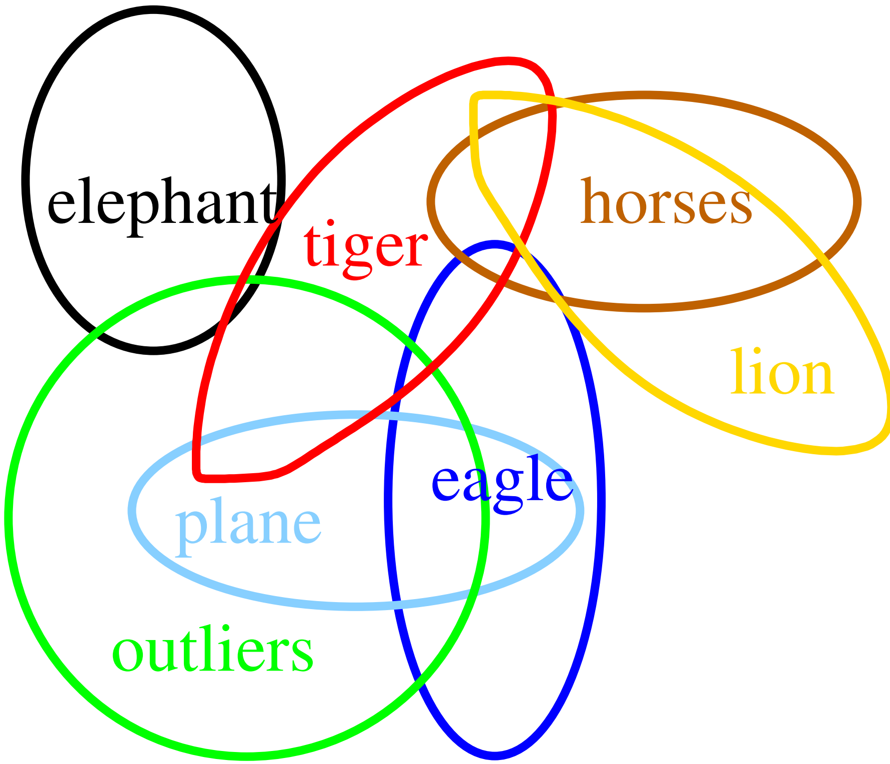
Integrating supervised data

A set of regions are labeled manually

6 CDs, 10 images from each

- eagles
- elephants
- tigers
- horses
- planes
- lions

Integrating supervised data

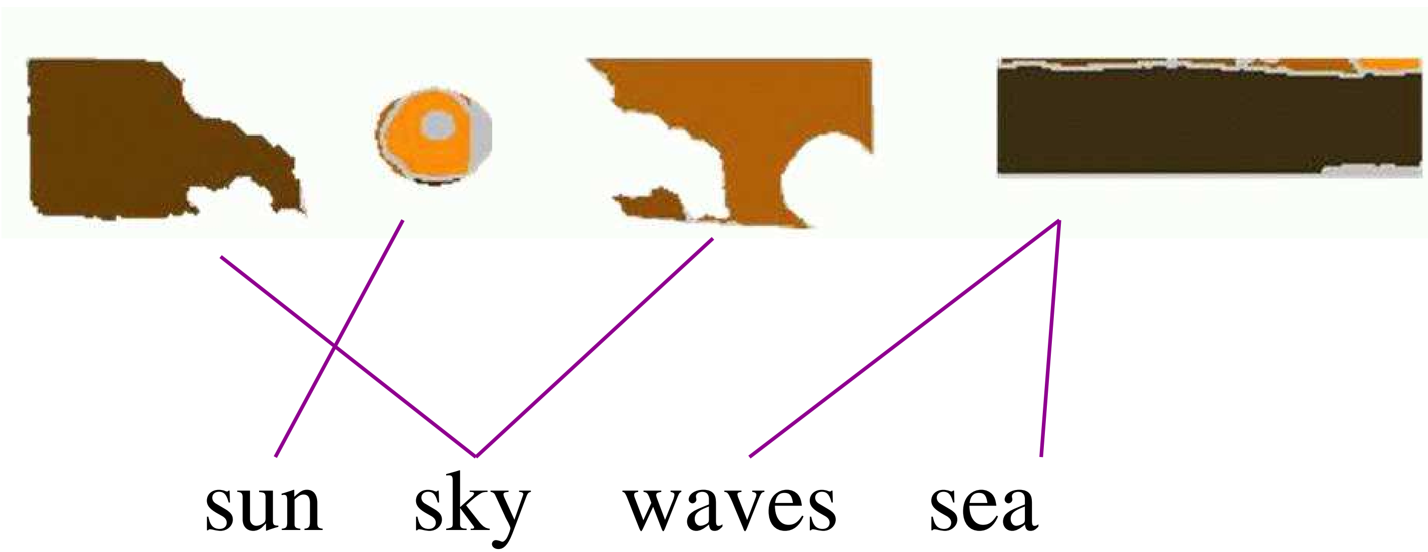


21 label words + outlier = 22 labeled classes

apply linear discriminant analysis

Integrating supervised data

Fixing correspondences:



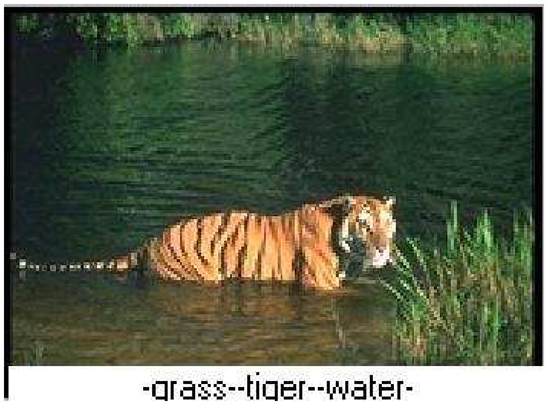
set the alignments between the labeled regions and the corresponding words to 1, and the others to 0

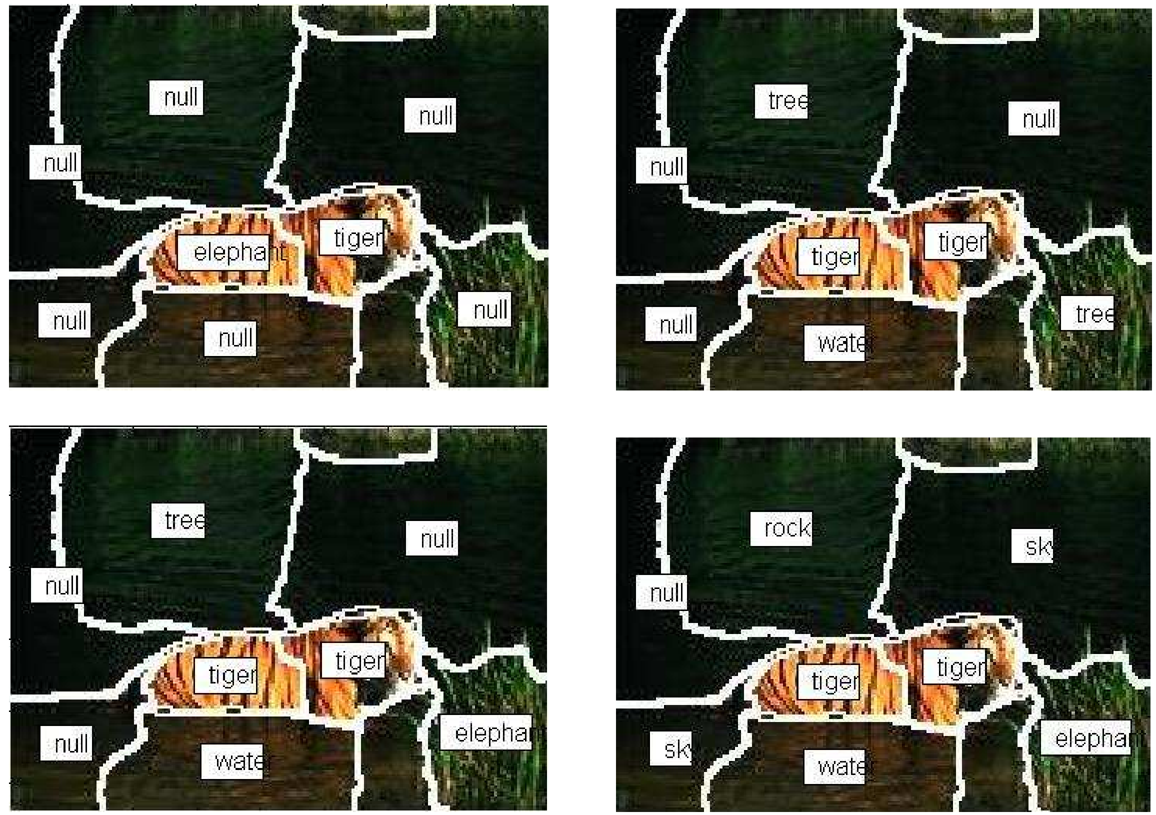
Integrating supervised data

4 methods can be applied :

method	clustering	training
method 1	k-means	unsupervised data + EM
method 2	labeled data	unsupervised data + EM
method 3	labeled data	nearest neigbor classifier
method 4	labeled data	supervised data + EM

Integrating supervised data





Integrating supervised data

label	method 1	method 2	method 4
tiger	elephant horses field	tiger null water	tiger null water
plane	sky plane forest	plane sky null	plane null sky
runway	null sky eagle	runway plane eagle	runway plane eagle
field	plane null sky	null horses field	field null elephant
horses	tiger null forest	null tiger tree	horses null tiger
sky	forest sky tiger	sky eagle null	sky null eagle
elephant	sky null grass	tree elephant null	elephant null tree
grass	horses null plane	grass horses null	grass horses field
tree	plane sky runway	elephant horses null	tree field horses
water	tiger plane water	water null sky	water null sky
lion	tiger null plane	grass lion tiger	lion grass tiger

Integrating supervised data

False positive and false negative rates :

word	supervised	nearest neighbor
eagle	0.0000-1.0000	0.8487-0.6714
forest	0.0000-1.0000	0.9524-0.9048
grass	0.7736-0.6364	0.7807-0.3788
horses	0.8231-0.6286	0.8496-0.7571
lion	0.7520-0.5714	0.7582-0.6857
rocks	0.0000-1.0000	0.9884-0.9091
runway	0.7647-0.4286	0.7647-0.4286
sky	0.6630-0.7207	0.6813-0.4775
tree	0.8667-0.5135	0.9355-0.8378
water	0.8033-0.6620	0.8047-0.5352

Conclusions

We proposed a new approach to object recognition

- motivated by the available annotated image collections,
- inspired from machine translation.

The proposed method

- can learn correspondences between image regions and words,
- is unsupervised - using the available large data sets efficiently,
- can be used for object recognition at a broad scale;
  - region naming: predict words corresponding to particular regions,
  - auto-annotation: predict words associated aith whole images.

Conclusions

The system is applied on the Corel data set and its performance is evaluated. The words predicted by the system is measured using:

- a set of hand-labelled images,
- annotation as a proxy.

The sytem performance is compared against

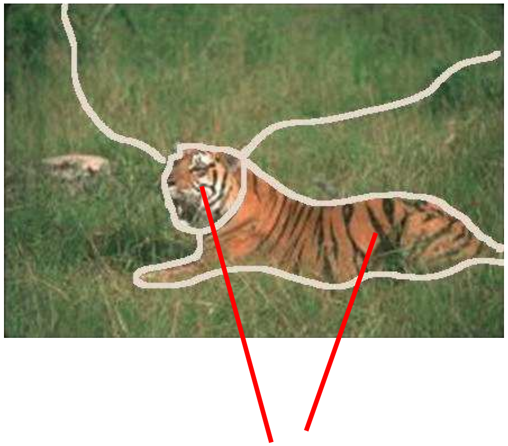
- empirical word densities,
- co-occurrences of blobs and regions.

Discussion and future directions

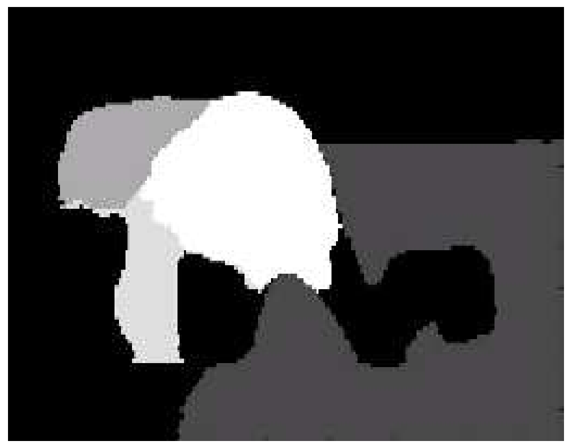
The proposed method has

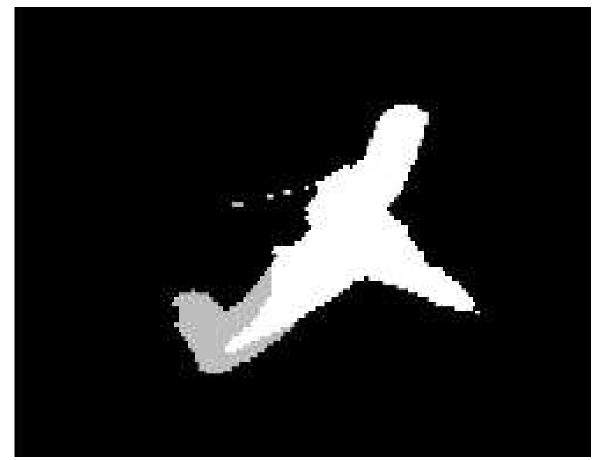
- problems due to annotations;
  - NULL and fertility,
  - compound words,
- problem due to the image;
  - segmentation,
  - feature extraction,
  - clustering.

Future Directions - Propose merging



propose merging





Other available data sets

Corel Image Data	40,000 images
Fine Arts Museum of San Francisco	83,000 images
Cal-flora	20,000 images
News photos with captions	1,500 images per day
Hulton-Getty collection	40,000,000 images
TV news archives	several terabytes
Google Image Crawl	$>$ 330,000,000 images

Thanks for listening!

Sample images with annotations

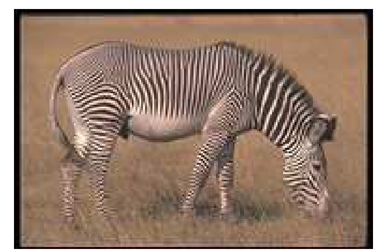
water harbor sky clouds

garden building flowers trees

garden flowers house trees

plane jet su-27 sky

diver fish ocean



zebra grass herd planes

flo

Problems in Object Recognition

what is an object?

how to model?

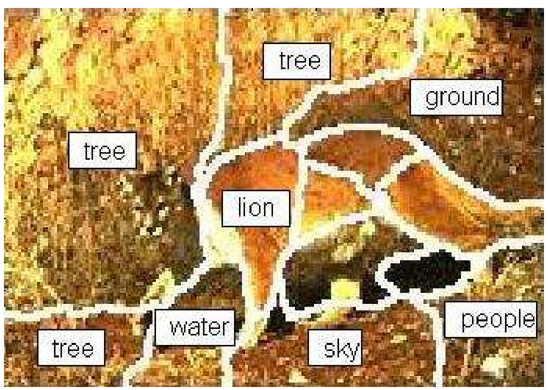
scalability

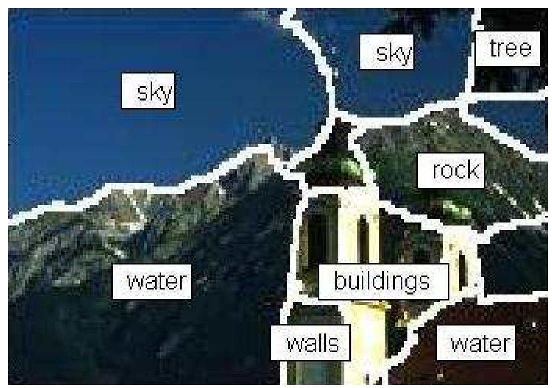
Tokenization

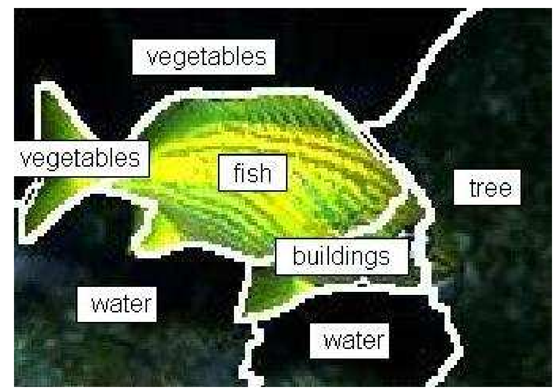
cat cougar hills rock

church mountain tree

fish reefs water



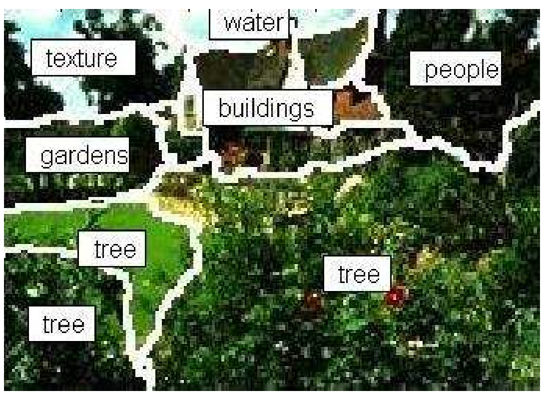


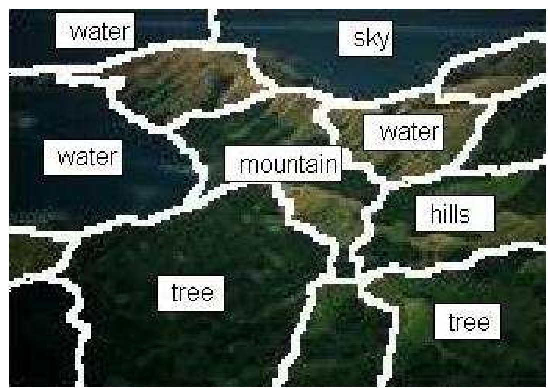


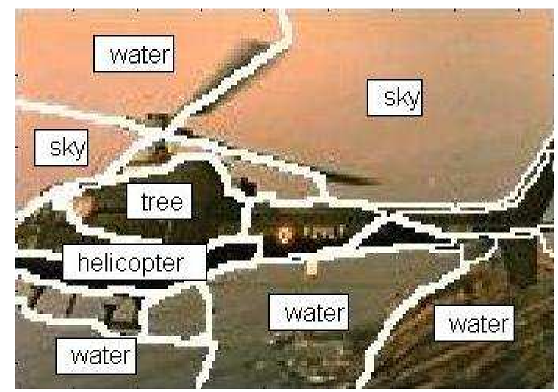
gardens house tree

mountain tree water

coast helicopter water







Merging indistinguishable words