Text and images are separately ambiguous
Text and images are separately ambiguous
Text and images are separately ambiguous
Text and images are separately ambiguous

However:
They tend not to be when jointly displayed

Figure: Ubuntu 7.10 with Compiz Fusion
Outline

- Applications
- Data Preprocessing
- Annotation Models
- Correspondence Models
- Model Integration
- Evaluation
- Experiments
- Conclusion
Applications
Applications - Real World

- Browsing and searching
- Identify faces, naked people, pedestrians or cars
Applications - Real World

- Browsing and searching
- Identify faces, naked people, pedestrians or cars

However, there is a large disparity between user needs and what technology supplies
Applications - User Requests

A request to a stock photo library

Pretty girl doing something active, sporty in a summery setting, beach - not wearing lycra, exercise clothes - more relaxed in tee-shirt. Feature is about deodorant so girl should look active - not sweaty but happy, healthy, carefree - nothing too posed or set up - nice and natural looking.
Applications - User Requests

A request to a stock photo library

Pretty girl doing something active, sporty in a summery setting, beach - not wearing lycra, exercise clothes - more relaxed in tee-shirt. Feature is about deodorant so girl should look active - not sweaty but happy, healthy, carefree - nothing too posed or set up - nice and natural looking.

So, what do users request from an image matching application?
Applications - User Requests

- Images both by object kinds and identities
- Images by what they depict and by what they are about
- Queries based on image histograms, texture, overall appearance, etc.
  are vanishingly uncommon
- Text associated with images is extremely useful in practice
Images both by object kinds and identities
Images by what they depict and by what they are about
Queries based on image histograms, texture, overall appearance, etc. are vanishingly uncommon
Text associated with images is extremely useful in practice

Given the user requests, how do we fulfil these requests?
Boolean Queries: Does the text link to the image? YES or NO

Probability Models: Does the text link to the image? 83.74% possibility
Boolean Queries: Does the text link to the image? YES or NO

Probability Models: Does the text link to the image? 83.74% possibility

A Probability Model would be more beneficial: one doesn’t need to know exactly the right search terms to get useful results.
One possible way of using probability models is to predict text given images.

Annotation  Predict annotations of entire images using all information present.

Correspondence  Associate particular words with particular image substructures.
One possible way of using probability models is to predict text given images.

**Annotation** Predict annotations of entire images using all information present.

**Correspondence** Associate particular words with particular image substructures.

So, what specific models are there and how are Annotation and Correspondence done?
Data Preprocessing
Each image is segmented using Normalised Cuts (Shi and Malik, 2000).

Represent 8 largest regions in each image by computing a set of 40 features for each region.
Each image is segmented using Normalised Cuts (Shi and Malik, 2000).

Represent 8 largest regions in each image by computing a set of 40 features for each region.

What are the features?
Region features

- **Size**: Portion of the image covered by the region
- **Position**: Coordinates of the region center of mass normalized by the image dimensions.
- **Color**: The average and standard deviation of \((R,G,B), (L,a,b)\) and \((r=R/(R+G+B), g=G/(R+G+B))\) over the region.
- **Texture**: The average and variance of 16 filter responses.
- **Shape**: The ratio of the area to the perimeter squared, the moment of inertia and the ratio of the region area to that of its convex hull.

We will refer to a region, together with the features, as a **blob**.
Annotation Models
Annotation Models - Introduction

- Multi-Modal Hierarchical Aspect Model
- Mixture of Multi-Modal Latent Dirichlet Allocation
Figure: Multi-modal extension of a hierarchical model for text.

- Image regions are generated using a Gaussian distribution.
- Words are generated using a multinomial distribution.
A document is modeled by a sum over the clusters, weighted by the probability that the document is in the cluster.

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

\(l, c\) Levels, Clusters  \\
\(w\) Words in document \(d\)  \\
\(b\) The image regions in document \(d\)  \\
\(D\) The set of observations for document  \\
\(W, B\) The set of words and blobs for the document \(D = W \cup B\)  \\
\(\frac{N_w}{N_{w,d}}, \frac{N_b}{N_{b,d}}\) Normalisation constants for differing numbers of words and blobs in each image.

We will refer to this as **model I-0** in the results later on.
We also experimented with allowing a cluster dependent level structure where \( p(l|d) \) is replaced with \( p(l|c, d) \), and refered this as model I-1:

\[
p(D|d) = \sum_c p(c) \prod_{w \in W} \left[ \sum_l p(w|l, c)p(l|c, d) \right]^{N_w} \prod_{b \in B} \left[ \sum_l p(b|l, c)p(l|c, d) \right]^{N_b}
\]
We also experimented with allowing a cluster dependent level structure where $p(l|d)$ is replaced with $p(l|c, d)$, and referred this as model I-1:

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

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

Both model I-0 and model I-1 are not generative, they are specific to the documents in the training set.
Two alternatives for generalization:

- Marginalize out the training data
- Estimate the mixing weights using a cluster specific average computed during training (model I-2).
Two alternatives for generalization:

- Marginalize out the training data
- Estimate the mixing weights using a cluster specific average computed during training (model I-2).

\[
p(D) = \sum_c p(c) \prod_{w \in W} \left[ \sum_l p(w|l, c)p(l|c) \right]^{\frac{N_w}{N_w,d}} \prod_{b \in B} \left[ \sum_l p(b|l, c)p(l|c) \right]^{\frac{N_b}{N_b,d}}
\]
**Model Fitting** All models are fit using EM algorithm.

**Image Based Word Prediction**

\[
p(w|B) \propto p(w, B) \\
\propto \sum_c p(c) p(w|c) p(B|c) \\
= \sum_c p(c) \left[ \sum_l p(w|l, c)p(l|c) \right] \prod_{b \in B} \left[ \sum_l p(b|l, c)p(l|c) \right]^{N_b}_{N_{b,d}}
\]
Latent Dirichlet Allocation (LDA)

A generative probabilistic model for independent collections of data where each collection is modelled by a randomly generated mixture over latent factors.
Latent Dirichlet Allocation (LDA)

A generative probabilistic model for independent collections of data where each collection is modelled by a randomly generated mixture over latent factors.

Example

Text modelling: A LDA model with two topics - **CAT, DOG**

**CAT**: milk, meow, kitten

**DOG**: puppy, bark, bone

A document is generated by picking a distribution over topics, and given this distribution, picking the topic of each specific word. Then words are generated given their topics.
Choose one of $J$ mixture components $c \sim \text{Multinomial}(\eta)$. 
1. Choose one of $J$ mixture components $c \sim \text{Multinomial}(\eta)$.

2. Conditioned on the mixture component, choose a mixture over $K$ factors, $\theta \sim \text{Dir}(\alpha_c)$. 
Choose one of $J$ mixture components $c \sim \text{Multinomial}(\eta)$.

Conditioned on the mixture component, choose a mixture over $K$ factors, $\theta \sim \text{Dir}(\alpha_c)$.

For each of the $N$ words:

1. Choose one of $K$ factors $z_n \sim \text{Multinomial}(\theta)$.
2. 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.
Choose one of $J$ mixture components $c \sim \text{Multinomial} (\eta)$.

Conditioned on the mixture component, choose a mixture over $K$ factors, $\theta \sim \text{Dir}(\alpha_c)$.

For each of the $N$ words:

1. Choose one of $K$ factors $z_n \sim \text{Multinomial}(\theta)$.
2. 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.

For each of the $M$ blobs:

1. Choose a factor $s_m \sim \text{Multinomial}(\theta)$.
2. 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 $c$. 
Correspondence Models
It’s natural to want to build models that can predict words for specific image regions rather than for a whole image, there are several simple ways:

- The simplest: Discrete Data Translation
- Hierarchical Clustering Model
- MoM-LDA
It is inspired by machine translation: a lexicon links discrete objects (words in one language) to discrete objects (words in the other language).

- Use K-mean to Vector-quantize representations of image regions. Each region then get a single label (blob token).
- Predict words: construct a joint probability table linking word token to blob token.
Hierarchical clustering models encode this correspondence through **Co-occurrence**

**Co-occurrence example**

The word ”tiger” always co-occurs with an orange stripy region and never otherwise.
Hierarchical clustering models encode this correspondence through **Co-occurrence**

**Co-occurrence example**

The word ”tiger” always co-occurs with an orange stripy region and never otherwise.

**Word prediction**

\[
p(w|b) \propto \sum_c p(c) \sum_l p(l)p(w|l, c)p(b|l, c) \quad \text{(Region only)}
\]

\[
p(w|b) \propto \sum_c p(c|B) \sum_l p(l)p(w|l, c)p(b|l, c) \quad \text{(Region cluster)}
\]
Integrating correspondence and Hierarchical clustering
Problem in translating from image region to word is similar with machine translation.

<table>
<thead>
<tr>
<th>Problem in translating from English to French</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun → soleil</td>
</tr>
<tr>
<td><strong>But if</strong> the surrounding words are computer related</td>
</tr>
<tr>
<td>sun → sun</td>
</tr>
</tbody>
</table>
Problem in translating from image region to word is similar with machine translation.

<table>
<thead>
<tr>
<th>Problem in translating from English to French</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun → soleil</td>
</tr>
<tr>
<td><strong>But if</strong> the surrounding words are computer related</td>
</tr>
<tr>
<td>sun → sun</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build explicit correspondence information into existing hierarchical clustering model</td>
</tr>
<tr>
<td>1. Linking Word Emission and Region Emission Probabilities with Mixture Weights</td>
</tr>
<tr>
<td>2. Paired Word and Region Emission at Nodes</td>
</tr>
</tbody>
</table>
To implement this strategy by having the vertical mixture weights for the regions carries over to that for words.

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

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

where

\[
p(l|B, c, d) \propto \sum_{b \in B} p(l|b, c, d)\]
Linking Word Emission and Region Emission Probabilities with Mixture Weights

To implement this strategy by having the vertical mixture weights for the regions carries over to that for words.

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

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

where

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

**Dependence Models**

This model is **Model D-0**

\[p(l|c, d) \Rightarrow p(l|d)\text{get D-1 cluster dependent level distributions}\]

\[p(l) \Rightarrow p(l|d)\text{ get D-2 drop the dependency on training set}\]
This method further tightens the relationship between the regions and words. Then the observed words and regions are emitted in pairs

\[ D = \{(w, b)\} \]

\[
p(D|d) = \sum_c p(c) \prod_{(w, b) \in D} \left[ \sum_l p((w, b)|l, c)p(l|d) \right]
\]
This method further tightens the relationship between the regions and words. Then the observed words and regions are emitted in pairs \( D = \{(w, b)\} \)

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

Correspondence Models

This model is **Model C-0**
Model I-1 and I-2 are similarly modified to get **C-1** and **C-2**
This method further tightens the relationship between the regions and words. Then the observed words and regions are emitted in pairs:

\[ D = \{(w, b)\} \]

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

**Correspondence Models**

This model is **Model C-0**

Model I-1 and I-2 are similarly modified to get **C-1** and **C-2**

The correspondence is not provide in training data, we need to estimate correspondence as part of the training process.
All methods for computing the correspondence assume that the probability that a word and a segment correspond can be estimated by the probability that they are emitted from the same node. Using $w \leftrightarrow b$ to denote that the word, $w$, and the region, $b$, correspond, in case C-0

$$p(w \leftrightarrow b) \approx \sum_c p(c) \sum_l p((w, b)|l, c)p(l|d)$$
Correspondence Issues

Issues

The choice of correspondence model, Should there be a one-one map between regions and words (Usually impossible.), there is no option of deciding that a region corresponds to no word.

Solution

- A special word NULL
- Fertility: one to many
- Refusal to predict: when the annotation with the highest probability given the region has too low a probability
Evaluation
Evaluation

- Measuring Annotation Performance:
  Compare the words predicted by models with words actually present for held-out data

- Measuring Correspondence Performance:
  Examine whether we predict appropriate words for each particular region


**Empirical word frequency**

- Relative to prediction performance
- Matching the performance empirical density
- A sensible indicator
Empirical word frequency

Quality of word posterior distribution
Empirical word frequency

Quality of word posterior distribution

- Estimate the Kullback-Leibler divergence between predictive distribution $q(w|B)$, and target distribution $p(w)$. 
Empirical word frequency

**Quality of word posterior distribution**

- Estimate the Kullback-Leibler divergence between predictive distribution \(q(w|B)\), and target distribution \(p(w)\).
- \(E_{KL}^{(model)} = \sum_{w \in \text{vocabulary}} p(w) \log \frac{p(w)}{q(w|B)}\)
Empirical word frequency

**Quality of word posterior distribution**

- Estimate the Kullback-Leibler divergence between predictive distribution $q(w|B)$, and target distribution $p(w)$.

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

$$= \text{constant} - \frac{1}{K} \sum_{w \in \text{observed}} \log q(w|B).$$
Empirical word frequency

Quality of word posterior distribution

**Goodness of the model:**
Evaluation-Annotation Models

Empirical word frequency

Quality of word posterior distribution

**Goodness of the model: loss function**
Evaluation-Annotation Models

Empirical word frequency

Quality of word posterior distribution

**Goodness of the model: loss function**

- Traditional zero-one loss is highly misleading
Evaluation-Annotation Models

Empirical word frequency

Quality of word posterior distribution

**Goodness of the model: loss function**

- Traditional zero-one loss is highly misleading
- \( E_{NS}^{model} = \frac{r}{n} - \frac{w}{(N - n)} \)
  - \( N \) the vocabulary size
  - \( n \) the number of actual words for the image
  - \( r \) the number of words predicted correctly
  - \( w \) the number of words predicted incorrectly
Empirical word frequency

Quality of word posterior distribution

**Goodness of the model: loss function**

- Traditional zero-one loss is highly misleading
- \( E_{NS}^{model} = \frac{r}{n} - \frac{w}{(N - n)} \)
  
  - \( N \) the vocabulary size
  - \( n \) the number of actual words for the image
  - \( r \) the number of words predicted correctly
  - \( w \) the number of words predicted incorrectly

- \( E_{PR}^{model} = \frac{r}{n} \), based on the n best words.
Empirical word frequency

Quality of word posterior distribution

\[ E_{KL}^{(model)} = constant - \frac{1}{K} \sum_{w \in \text{observed}} \log q(w|B). \]

Goodness of the model: loss function

\[ E_{NS}^{model} = r/n - w/(N - n) \]
\[ E_{PR}^{model} = r/n, \text{ based on the n best words.} \]
Empirical word frequency

Quality of word posterior distribution

\[
E_{KL}^{(model)} = constant - \frac{1}{K} \sum_{w \in \text{observed}} \log q(w|B).
\]

Goodness of the model: loss function

\[
E_{NS}^{model} = \frac{r}{n} - \frac{w}{(N-n)}
\]

\[
E_{PR}^{model} = \frac{r}{n}, \text{ based on the } n \text{ best words.}
\]

\[
E_{KL} = \frac{1}{K} \sum_{data} (E^{empirical}_{KL} - E^{model}_{KL})
\]

\[
E_{NS} = \frac{1}{K} \sum_{data} (E^{empirical}_{NS} - E^{model}_{NS})
\]

\[
E_{PR} = \frac{1}{K} \sum_{data} (E^{empirical}_{PR} - E^{model}_{PR})
\]

Negative → worse; positive → better
Manual correspondence scoring
Manual correspondence scoring

- Limit the size of the dataset
- Contain significant noise
Manual correspondence scoring

- Limit the size of the dataset
- Contain significant noise

Assumption

A method that cannot predict annotations accurately is unlikely to predict correspondence well.
Manual correspondence scoring

- Limit the size of the dataset
- Contain significant noise

Using annotation as a proxy.

Assumption

A method that cannot predict annotations accurately is unlikely to predict correspondence well.
Using annotation as a proxy

- Image based methods
- Region based methods
Using annotation as a proxy

- Image based methods
  Compute the annotation in the natural way.
- Region based methods
Using annotation as a proxy

- Image based methods
  Compute the annotation in the natural way.

- Region based methods
  Consider each region separately.
  Use image based annotation method to compute region word posteriors.
  Marginalize out the correspondence of each region from the model, and test it as an annotation model.
Experiments
Experiments

Use images from 160 CD’s from Corel image data set.

75% training sets, 25% “standard” held out sets.

Exclude words occur less than 20 times in test set.
Experiments

Annotation Results.

Correspondence Results
Experiments

Annotation Results.

- KL-divergence score measure
- Normalized score measure
- Comparison of models using different scores

Correspondence Results
Annotation Results

- KL-divergence score measure
- Normalized score measure.
- Comparison of models using different scores.
Annotation Results

- **KL-divergence score measure**
  - Check for overfitting.
  - Number of iteration.

- Normalized score measure.

- Comparison of models using different scores.
Annotation Results

- KL-divergence score measure
  - Check for overfitting:
  - Number of iteration:
Annotation Results

- KL-divergence score measure
  - Check for overfitting: not a big problem
  - Number of iteration:
Annotations Results

- KL-divergence score measure
  - Check for overfitting: not a big problem
  - Number of iteration: performance on training set improved with number of iterations.
Annotation Results

- KL-divergence score measure
  - Check for overfitting: not a big problem
  - Number of iteration: performance on training set improved with number of iterations.

- Normalized score measure
Annotation Results
Annotation Results

- KL-divergence score measure
  - Check for overfitting: not a big problem
  - Number of iteration: performance on training set improved with number of iterations.

- Normalized score measure
Annnotation Results

- KL-divergence score measure
  - Check for overfitting: not a big problem
  - Number of iteration: performance on training set improved with number of iterations.

- Normalized score measure
  Go from zero to some peak, and then to drop down to zero again.
Annotation Results

- KL-divergence score measure
  - Check for overfitting: not a big problem
  - Number of iteration: performance on training set improved with number of iterations.

- Normalized score measure
  Go from zero to some peak, and then to drop down to zero again.

- Comparison of models using different scores.
Annotation Results

Annotation results are compared between models. Measure PR:

<table>
<thead>
<tr>
<th>Method</th>
<th>Training data</th>
<th>Held out data</th>
<th>Novel data</th>
</tr>
</thead>
<tbody>
<tr>
<td>linear-I-0-doc-vert</td>
<td>0.130 (0.003)</td>
<td>0.095 (0.003)</td>
<td>0.057 (0.003)</td>
</tr>
<tr>
<td>binary-I-0-ave-vert</td>
<td>0.130 (0.005)</td>
<td>0.082 (0.004)</td>
<td>0.023 (0.005)</td>
</tr>
<tr>
<td>binary-I-0-doc-vert</td>
<td>0.152 (0.005)</td>
<td>0.094 (0.004)</td>
<td>0.034 (0.005)</td>
</tr>
<tr>
<td>binary-I-0-region-cluster</td>
<td>0.157 (0.005)</td>
<td>0.099 (0.004)</td>
<td>0.038 (0.005)</td>
</tr>
<tr>
<td>binary-I-0-region-only</td>
<td>0.140 (0.005)</td>
<td>0.099 (0.003)</td>
<td>0.037 (0.006)</td>
</tr>
<tr>
<td>binary-I-2-ave-vert</td>
<td>0.141 (0.005)</td>
<td>0.087 (0.003)</td>
<td>0.023 (0.004)</td>
</tr>
<tr>
<td>binary-I-2-doc-vert</td>
<td>0.137 (0.005)</td>
<td>0.090 (0.004)</td>
<td>0.036 (0.004)</td>
</tr>
<tr>
<td>binary-I-2-region-cluster</td>
<td>0.141 (0.005)</td>
<td>0.099 (0.004)</td>
<td>0.039 (0.004)</td>
</tr>
<tr>
<td>binary-I-2-region-only</td>
<td>0.133 (0.005)</td>
<td>0.104 (0.004)</td>
<td>0.040 (0.005)</td>
</tr>
<tr>
<td>linear-D-0-doc-vert</td>
<td>0.147 (0.002)</td>
<td>0.102 (0.002)</td>
<td>0.059 (0.004)</td>
</tr>
<tr>
<td>binary-D-0-ave-vert</td>
<td>0.126 (0.005)</td>
<td>0.081 (0.003)</td>
<td>0.024 (0.005)</td>
</tr>
<tr>
<td>binary-D-0-doc-vert</td>
<td>0.160 (0.005)</td>
<td>0.094 (0.003)</td>
<td>0.037 (0.005)</td>
</tr>
<tr>
<td>binary-D-0-region-cluster</td>
<td>0.166 (0.005)</td>
<td>0.100 (0.003)</td>
<td>0.040 (0.005)</td>
</tr>
<tr>
<td>binary-D-0-region-only</td>
<td>0.143 (0.005)</td>
<td>0.103 (0.003)</td>
<td>0.038 (0.005)</td>
</tr>
<tr>
<td>binary-D-2-ave-vert</td>
<td>0.143 (0.005)</td>
<td>0.088 (0.003)</td>
<td>0.021 (0.005)</td>
</tr>
<tr>
<td>binary-D-2-doc-vert</td>
<td>0.173 (0.005)</td>
<td>0.102 (0.003)</td>
<td>0.036 (0.005)</td>
</tr>
<tr>
<td>binary-D-2-region-cluster</td>
<td>0.177 (0.005)</td>
<td>0.108 (0.003)</td>
<td>0.039 (0.005)</td>
</tr>
<tr>
<td>binary-D-2-region-only</td>
<td>0.152 (0.005)</td>
<td>0.108 (0.003)</td>
<td>0.036 (0.005)</td>
</tr>
<tr>
<td>linear-C-0-region-only</td>
<td>0.103 (0.003)</td>
<td>0.067 (0.002)</td>
<td>0.035 (0.005)</td>
</tr>
<tr>
<td>binary-C-0-ave-vert</td>
<td>0.115 (0.004)</td>
<td>0.070 (0.003)</td>
<td>0.015 (0.005)</td>
</tr>
<tr>
<td>binary-C-0-doc-vert</td>
<td>0.137 (0.003)</td>
<td>0.078 (0.002)</td>
<td>0.025 (0.004)</td>
</tr>
<tr>
<td>binary-C-0-region-cluster</td>
<td>0.142 (0.003)</td>
<td>0.085 (0.002)</td>
<td>0.030 (0.005)</td>
</tr>
<tr>
<td>binary-C-0-region-only</td>
<td>0.128 (0.003)</td>
<td>0.085 (0.003)</td>
<td>0.032 (0.005)</td>
</tr>
<tr>
<td>discrete-translation</td>
<td>0.129 (0.004)</td>
<td>0.073 (0.003)</td>
<td>0.029 (0.005)</td>
</tr>
<tr>
<td>MoM-LDA</td>
<td>0.053 (0.002)</td>
<td>0.050 (0.002)</td>
<td>0.038 (0.002)</td>
</tr>
</tbody>
</table>
Annotation results are compared between models. Measure NS:

<table>
<thead>
<tr>
<th>Method</th>
<th>Training data</th>
<th>Held out data</th>
<th>Novel data</th>
</tr>
</thead>
<tbody>
<tr>
<td>linear-1-0-doc-vert</td>
<td>0.301 (0.005)</td>
<td>0.174 (0.007)</td>
<td>0.081 (0.007)</td>
</tr>
<tr>
<td>binary-1-0-ave-vert</td>
<td>0.294 (0.006)</td>
<td>0.154 (0.006)</td>
<td>0.064 (0.008)</td>
</tr>
<tr>
<td>binary-1-0-doc-vert</td>
<td>0.325 (0.006)</td>
<td>0.160 (0.007)</td>
<td>0.065 (0.008)</td>
</tr>
<tr>
<td>binary-1-0-region-cluster</td>
<td>0.332 (0.006)</td>
<td>0.168 (0.007)</td>
<td>0.068 (0.008)</td>
</tr>
<tr>
<td>binary-1-0-region-only</td>
<td>0.234 (0.006)</td>
<td>0.160 (0.006)</td>
<td>0.062 (0.008)</td>
</tr>
<tr>
<td>binary-1-2-ave-vert</td>
<td>0.331 (0.006)</td>
<td>0.164 (0.008)</td>
<td>0.068 (0.007)</td>
</tr>
<tr>
<td>binary-1-2-doc-vert</td>
<td>0.322 (0.006)</td>
<td>0.170 (0.008)</td>
<td>0.074 (0.008)</td>
</tr>
<tr>
<td>binary-1-2-region-cluster</td>
<td>0.324 (0.006)</td>
<td>0.179 (0.008)</td>
<td>0.076 (0.008)</td>
</tr>
<tr>
<td>binary-1-2-region-only</td>
<td>0.228 (0.006)</td>
<td>0.163 (0.006)</td>
<td>0.068 (0.007)</td>
</tr>
<tr>
<td>linear-D-0-doc-vert</td>
<td>0.321 (0.005)</td>
<td>0.167 (0.006)</td>
<td>0.076 (0.008)</td>
</tr>
<tr>
<td>binary-D-0-ave-vert</td>
<td>0.284 (0.007)</td>
<td>0.151 (0.007)</td>
<td>0.061 (0.008)</td>
</tr>
<tr>
<td>binary-D-0-doc-vert</td>
<td>0.321 (0.007)</td>
<td>0.157 (0.007)</td>
<td>0.064 (0.008)</td>
</tr>
<tr>
<td>binary-D-0-region-cluster</td>
<td>0.330 (0.006)</td>
<td>0.166 (0.008)</td>
<td>0.067 (0.008)</td>
</tr>
<tr>
<td>binary-D-0-region-only</td>
<td>0.239 (0.006)</td>
<td>0.162 (0.007)</td>
<td>0.064 (0.007)</td>
</tr>
<tr>
<td>binary-D-2-ave-vert</td>
<td>0.312 (0.005)</td>
<td>0.162 (0.003)</td>
<td>0.066 (0.005)</td>
</tr>
<tr>
<td>binary-D-2-doc-vert</td>
<td>0.358 (0.005)</td>
<td>0.172 (0.003)</td>
<td>0.069 (0.005)</td>
</tr>
<tr>
<td>binary-D-2-region-cluster</td>
<td>0.360 (0.005)</td>
<td>0.179 (0.003)</td>
<td>0.072 (0.005)</td>
</tr>
<tr>
<td>binary-D-2-region-only</td>
<td>0.248 (0.005)</td>
<td>0.167 (0.003)</td>
<td>0.066 (0.005)</td>
</tr>
<tr>
<td>linear-C-0-region-only</td>
<td>0.240 (0.005)</td>
<td>0.124 (0.007)</td>
<td>0.046 (0.006)</td>
</tr>
<tr>
<td>binary-C-0-ave-vert</td>
<td>0.252 (0.006)</td>
<td>0.143 (0.007)</td>
<td>0.060 (0.008)</td>
</tr>
<tr>
<td>binary-C-0-doc-vert</td>
<td>0.281 (0.006)</td>
<td>0.148 (0.006)</td>
<td>0.054 (0.007)</td>
</tr>
<tr>
<td>binary-C-0-region-cluster</td>
<td>0.290 (0.006)</td>
<td>0.157 (0.007)</td>
<td>0.064 (0.007)</td>
</tr>
<tr>
<td>binary-C-0-region-only</td>
<td>0.233 (0.006)</td>
<td>0.163 (0.006)</td>
<td>0.071 (0.006)</td>
</tr>
<tr>
<td>discrete-translation</td>
<td>0.318 (0.005)</td>
<td>0.111 (0.007)</td>
<td>0.016 (0.008)</td>
</tr>
<tr>
<td>MoM-LDA</td>
<td>0.125 (0.005)</td>
<td>0.107 (0.005)</td>
<td>0.041 (0.007)</td>
</tr>
</tbody>
</table>
Annotation results are compared between models.

Measure KL:

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

Comparison of models using different scores.

- Words are generally associated with pieces of images.
- Methods using clustering are reliant on having images which are close to the training data.
- The result for the MoM-LDA model are worth nothing.
Study the keyword prediction error, $E_{PR}^{\text{empirical}} - E_{PR}^{\text{model}}$
Correspondence Result

- Study the keyword prediction error, $E_{PR}^{\text{empirical}} - E_{PR}^{\text{model}}$
Correspondence Result

- Study the keyword prediction error, $E_{PR}^{\text{empirical}} - E_{PR}^{\text{model}}$

<table>
<thead>
<tr>
<th>Method</th>
<th>PR measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>linear-I-0-region-only</td>
<td>0.099 (0.02)</td>
</tr>
<tr>
<td>binary-I-0-region-cluster</td>
<td>0.101 (0.01)</td>
</tr>
<tr>
<td>binary-I-0-region-only</td>
<td>0.103 (0.01)</td>
</tr>
<tr>
<td>binary-I-2-region-cluster</td>
<td>0.101 (0.01)</td>
</tr>
<tr>
<td>binary-I-2-region-only</td>
<td>0.093 (0.01)</td>
</tr>
<tr>
<td>linear-D-0-region-only</td>
<td>0.132 (0.01)</td>
</tr>
<tr>
<td>binary-D-0-region-cluster</td>
<td>0.096 (0.01)</td>
</tr>
<tr>
<td>binary-D-0-region-only</td>
<td>0.104 (0.01)</td>
</tr>
<tr>
<td>binary-D-2-region-cluster</td>
<td>0.103 (0.01)</td>
</tr>
<tr>
<td>binary-D-2-region-only</td>
<td>0.092 (0.01)</td>
</tr>
<tr>
<td>linear-C-0-region-only</td>
<td>0.101 (0.01)</td>
</tr>
<tr>
<td>discrete-translation</td>
<td>0.066 (0.01)</td>
</tr>
</tbody>
</table>
Correspondence Result

- Study the keyword prediction error, $E_{PR}^{empirical} - E_{PR}^{model}$

<table>
<thead>
<tr>
<th>Method</th>
<th>PR measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>linear-I-0-region-only</td>
<td>0.099 (0.02)</td>
</tr>
<tr>
<td>binary-I-0-region-cluster</td>
<td>0.101 (0.01)</td>
</tr>
<tr>
<td>binary-I-0-region-only</td>
<td>0.103 (0.01)</td>
</tr>
<tr>
<td>binary-I-2-region-cluster</td>
<td>0.101 (0.01)</td>
</tr>
<tr>
<td>binary-I-2-region-only</td>
<td>0.093 (0.01)</td>
</tr>
<tr>
<td>linear-D-0-region-only</td>
<td>0.132 (0.01)</td>
</tr>
<tr>
<td>binary-D-0-region-cluster</td>
<td>0.096 (0.01)</td>
</tr>
<tr>
<td>binary-D-0-region-only</td>
<td>0.104 (0.01)</td>
</tr>
<tr>
<td>binary-D-2-region-cluster</td>
<td>0.103 (0.01)</td>
</tr>
<tr>
<td>binary-D-2-region-only</td>
<td>0.092 (0.01)</td>
</tr>
<tr>
<td>linear-C-0-region-only</td>
<td>0.101 (0.01)</td>
</tr>
<tr>
<td>discrete-translation</td>
<td>0.066 (0.01)</td>
</tr>
</tbody>
</table>

- Paired word-blob emission approach improves correspondence performance over annotation performance.

- For both correspondence and annotation, linear-D-0-region-only method appears to be the best overall choice.
Discussion
Discussion

- The Corel data set is simple
  - annotations are nouns
  - vocabulary is small
- Performance is almost certainly affected by the correspondence model used
- Have little information about the effect of supervision
- Large scale evaluation of correspondence models is genuinely difficult
  the key issue seems to be the entropy of the labels.
Thank you!
Any Questions?