상호 관계 기반 자동 이미지 주석 생성
(Correlation-based Automatic Image Captioning)

양형정†  Jia-yu Pan‡‡  Pinar Duygulu‡‡‡  Christos Falout‡‡‡‡
(Hyungieong Yang)  (Jia-Yu Pan)  (Pinar Duygulu)  (Christos Falout)

요 약 본 논문에서는 상호 관계에 기반한 자동 이미지 주석 생성 방법을 보인다. 새로운 실험 이미지 를 위한 자동 주석의 생성은 훈련 데이터 내의 주석과 함께 주어진 이미지를 이용하여 이미지의 시각적 속성과 텍스트 속성의 상호 관계를 발견해 뒤로 수행된다. 본 논문에서 제시하는 상호 관계 기반 자동 주석 생성 모델은 1) 시각적 속성의 적절한 군집화, 2) 시각적 속성과 텍스트 속성의 가중치 부여, 3) 노이즈 제거를 위한 차원 축소 등의 요소를 고려하여 설계 된다. 실험은 680 MB의 Corel 이미지 데이터를 이용하여 각 10개의 테이터 집합에 대해 수행 되었으며, 실험 결과, 시각적 속성과 텍스트 속성에 대한 가중치 부여와 시각적 속성의 적절한 군집화가 모델의 성능을 향상 시키며, 본 논문에서 제시한 상호 관계 기반 모델이 기존의 EM을 이용한 자동 주석 생성 모델에 비해 45%의 상대적 성능 향상을 보인다.

키워드 : 이미지 주석, 상호 관계, 고유값 분해, 클러스터링

Abstract This paper presents correlation-based automatic image captioning. Given a training set of annotated images, we want to discover correlations between visual features and textual features, so that we can automatically generate descriptive textual features for a new unseen image. We develop models with multiple design alternatives such as 1) adaptively clustering visual features, 2) weighting visual features and textual features, and 3) reducing dimensionality for noise supression. We experiment thoroughly on 10 data sets of various content styles from the Corel image database, about 680MB. The major contributions of this work are: (a) we show that careful weighting visual and textual features, as well as clustering visual features adaptively leads to consistent performance improvements, and (b) our proposed methods achieve a relative improvement of up to 45% on annotation accuracy over the state-of-the-art, EM approach.

Key words : Image annotation, correlation, Singular Value Decomposition, Clustering

1. 서 론

Content-based image retrieval (CBIR) systems, matching images based on visual similarities, have some limitations due to missing semantic information[1-4]. Manually annotating images with words could provide such semantic information. There are some collections where images or videos are annotated with descriptive texts (e.g., the Corel data set, some museum collections, news photographs on the web with captions, etc.). Integration of the textual and visual features provided by these annotated collections improves the performance of search and retrieval[5-8]. However, manual annotation is time consuming and error-prone. Recently, automatic image annotation, which derives words from image content, has achieved promising results. Leveraging the existing text retrieval systems, automatic image annotation could be useful for the construction of content-based image retrieval systems supporting semantic information.

Several automatic image annotation methods have been proposed for better indexing and retrieval in large image databases[5,7,9-11]. Some of these approaches generate keywords for an image by
mapping image regions to terms. In other words, captioning is conducted by finding the association between constituent regions of images and given terms for the images. Mori et al. [12] use co-occurrence statistics of image grids and words for modelling the association. Duygulu et al. [10] view the mapping as a translation of image regions to words, and learn the mapping between region groups and words by using an EM algorithm. Recently, probabilistic models such as the cross-media relevance model [7] and latent semantic analysis (LSA) based models [28] are also proposed for captioning.

In this paper, we present correlation-based automatic image captioning. Given a training set of annotated images, we want to discover the correlations between visual features and textual features, so that we can automatically generate descriptive textual features for a new unseen image. The framework of automatic image captioning consists of two parts: constructing a model for the association, and annotating new images. Images in the annotated image set are first segmented and numerical feature vectors are extracted. The segmented image feature vectors (each could be a blob or a grid) are clustered into K clusters. The K cluster centers together form a visual vocabulary for the image content.

A model is constructed to capture the association between terms and the visual vocabulary. Model parameters are trained using the given annotated image set. When a new image arrives, the new image is segmented. Each segmented region is labeled by a token in the visual vocabulary, based on some similarity function. Captioning terms for the new image will most likely describe the content of the image. The likelihood of each term is determined by the model trained in the first step, given the visual tokens of the new image. We develop models with multiple design alternatives such as 1) adaptively clustering visual features, 2) weighting visual features and textual features, and 3) reducing dimensionality for noise suppression, for the better association model. We experiment thoroughly on 10 data sets of various content styles from the Corel image database, about 680MB. The major contributions of this work are: (a) we show that careful weighting on visual and textual features, as well as adaptive visual feature clustering, leads to consistent performance improvements, and (b) our proposed methods achieve a relative improvement of up to 43% on annotation accuracy over the state-of-the-art, EM approach.

The rest of the paper is organized as follows: Section 2 gives the related work, followed by section 3 where an adaptive method for obtaining image region groups is explained. The proposed uniqueness weighting scheme and correlation-based image annotation methods are given in Section 4. Section 5 shows the experimental results on the Corel data set. Several discussions are given in Section 6. Section 7 concludes the paper.

2. Related Work

There have been several attempts at captioning images automatically. Basically, the essential question is how we associate the visual content of an image with its semantics (expressed by the annotated terms). Previous approaches differ in how the image's visual content is represented (e.g., in blobs or regions) and in the particular models which are used to capture the association (e.g., language translation model or conditional random field).

Image captioning In this study, we are interested in linking the visual and textual features for annotating the images automatically. Automatic image captioning is useful since manual annotation of these collections is subjective and requires a huge amount of human effort.

Maron et al. [13] use multiple-instance learning to train classifiers to identify particular keywords from image data using labeled bags of examples. In their approach, an image is a "positive" if it contains an object (e.g., tiger) in the image but "negative" if it doesn't. Wenin et al. [14] propose a semi-automatic strategy for annotating images utilizing users' feedback of the retrieval system. The query keywords which receive positive feedback are collected as possible annotation words to the retrieved images. Li and Wang [11] model image concepts with a 2-D multiresolution Hidden Markov Model and label images with the concepts
that best fit the content.

Recently, probabilistic models are proposed to capture the joint statistics between image regions and caption terms. Mori et al. [12] use co-occurrence statistics collected for words and image areas which are defined by a fixed grid. Duygulu et al. [10] utilize machine translation approach proposed by Brown et al. [15] to find the correspondences between words and types of image regions. Jeon et al. [7] propose a cross-media relevance model for words and types of image regions, which takes the advantage that an image can be described both using image features and words. Monay et al. [28] use latent semantic analysis (LSA) to find the association. These methods quantize or cluster the image features into discrete tokens and find correlations between the tokens and captioning terms. The quality of tokenization could affect the captioning accuracy.

Other works model directly the association between words and the numerical features of the regions. Barnard et al. [6,9,16] propose a generative hierarchical model, inspired by Hofmann’s aspect model for text [17], for integrating the semantic information provided by the text and visual information provided by image features. Blei and Jordan [5] propose correspondence Latent Dirichlet Allocation (Corr-LDA) model that finds conditional relationships between latent variable representations of sets of image regions and sets of words. The continuous-space relevance model (CRM) [18], and the contextual model which models spatial consistency by Markov random field [19] are also proposed to find the actual association between image regions and terms for image annotation and a greater goal of object recognition.

While most previous approaches are complex and delicate, we want to explore simpler yet superior methods, motivated by some approaches employed for efficient document retrievals.

**Clustering:** One important preprocessing step is the construction of the visual and textual vocabularies. The model is then constructed to link the visual and textual tokens in the vocabularies. Many clustering algorithms can be used for the vocabulary construction [20], where mostly used ones are K-means, K-Harmonic means [21], OPTICS [22] and [23]. However, all these algorithms need the user to specify the number of desirable clusters K.

There are works which try to adaptively determine the number of clusters K. X-means [24] and G-means [25] determine the optimal K for the K-means algorithm by evaluating the quality of clusters using different criteria, namely BIC and normality statistical test. They start with small K, and split the clusters of poor quality (effectively increases K) until the criteria are met. Among all these algorithms, we choose the G-means algorithm in this paper.

3. **Adaptive Visual Vocabulary Generation**

The common approach for automatic image captioning is to find the association between the visual elements and the caption terms of an image. At first, two sets of vocabularies, namely the vocabulary of visual information (visual vocabulary) and that for the content semantics (content vocabulary), are constructed. Usually, a set of terms are used as the vocabulary for content semantics. The visual vocabulary consists of tokens representing visual information on either a sub-region (a grid or an image segment) or the entire image. Then, a model is used to capture the association between tokens from the two vocabularies. In this work, we follow the work in [10] and use a term set as the content vocabulary, and a blob-token set as the visual vocabulary. Let us begin with the definition of an annotated image set.

**Definition 1** (Annotated image set) An annotated image set is a set of images \( I = \{I_1, ..., I_k\} \), where each image \( I_i \) is annotated with a set of \( W_i \) terms \( \{w_{i1}, ..., w_{i|W_i|}\} \), where \( W_i \) is the number of annotated terms.

Figure 1 gives two examples of annotated images along with their captioning terms. Let the two images be \( I_1, I_2 \), then \( W_1=4 \) and \( W_2=4 \).

In this paper, use different font styles for different types of symbols, namely: bold and italic symbols for sets (e.g., the image annotation set \( I \)); bold, uppercase symbols for matrices (e.g., \( D \)); bold, lowercase symbols for vectors (e.g., \( q \)); italic symbols for set sizes (e.g., \( W \)).
Definitions 2 (Term set of an annotated image set) The term set of an annotated image set $I = \{I_1, ..., I_B\}$, denoted as $W = \{w_1, ..., w_W\}$, is defined as the collection of all $W$ terms used as annotating terms for the images in $I$.

Definition 3 (Blob) A blob of an image is a contiguous, homogeneous region of the image, given by an image segmentation algorithm.

A blob is usually represented as a continuous-valued vector of features describing the characteristic of the region. Figure 2 illustrates the blobs of the two example images in Figure 1, along with their captioning terms. We use the normalized cuts algorithm in [26] to break an image into regions, and then map each region into a 30-d feature vector. We used features such as the mean and standard deviation of its RGB values, average responses to various texture filters, its position in the entire image layout, and some shape descriptors (e.g., major orientation and the area ratio of the bounding region to the real region). All features are normalized to have zero-mean and unit-variance. For a given image $I_i$, the number of blobs $B_i$ is not necessarily equal to the number of captioning terms $W_i$. For example, in Figure 2, the numbers of blobs in each image are $B_1=5$ and $B_2=2$ while the numbers of words are $W_1 = 4$ and $W_2 = 4$, respectively.

Definition 4 (Blob-token) A set of continuous-valued blobs represented as feature vectors can be clustered. Each cluster is labeled as a blob-token.

Definition 5 (Blob-token set of an annotated image set) The blob-token set of an annotated image set $I$, denoted as $B = \{b_1, ..., b_B\}$, is defined as the collection of all $B$ blob-tokens which appear in the individual images of $I$.

Figure 3 shows three examples of the blob-tokens. For presentation purpose, these blob-tokens are semantically labeled as "cat", "sky" and "sun", however, the actual labels to the blob-tokens are not crucial. The consistency among the member blobs of a blob-token is more important for applications. In other words, we would like to have all blobs of a blob-token similar to each other, and dissimilar to those not belong to this blob-token.

The quality of blob-tokens would affect the accuracy of image captioning. In [10], the blob-tokens are generated by applying K-means algorithm on all the raw blobs in an annotated image set, with the number of blob-tokens, $B$, set at 500. However, the choice of $B=500$ is by no means optimal. Intuitively, if the optimal $B=625$, then setting $B=500$ would inevitable mixing red blobs with blue blobs together (i.e., cluster them as the same blob-token). On the other hand, if $B=325$, then setting $B=500$ would generate clusters which are too fine and hurts the algorithm's ability on generalization.

In this study, we determine the number of blob tokens $B$ adaptively using the idea of G-means
[26]. Essentially, G-means is a wrapper around the K-means algorithm, it runs K-means starting from a small number of B, and split clusters (thus, increases B) which are not gaussian. The gaussianity of a cluster is checked by a statistical test (e.g., Kolmogorov–Smirnov test) on the distribution of the data points in that cluster. In our work, the blob-tokens adaptively found by G-means are the labels of the clusters. The number of blob-tokens generated for the training set are all less than 500, ranging from 339 to 495, mostly around 400. We refer the reader to [25] for the details.

4. Correlation-based Image Captioning

In this section, we propose correlation-based methods with proper weighting assignments on the terms and blob-tokens and dimension reduction for noise suppression. The common goal among the proposed methods is to have an estimate for \( p(w|b_i) \), the conditional probability of a term \( w \) given a blob-token \( b_i \). Since the number of terms and blob-tokens are fixed and finite, the goal is to estimate a table whose \((i, j)\) item is the desired \( p(w|b_i) \), which we called the association table. In this section, we propose 4 methods to obtain such estimates, namely, method Corr, Cos, SvdCorr, and SvdCos.

Table 1 shows the symbols and terminology we used in the paper.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I )</td>
<td>annotated image set of ( N ) images ( {I_1, ..., I_N} )</td>
</tr>
<tr>
<td>( W )</td>
<td>term set of ( W ) terms ( {w_1, ..., w_W} )</td>
</tr>
<tr>
<td>( B )</td>
<td>blob-token set of ( B ) tokens ( {b_1, ..., b_B} )</td>
</tr>
<tr>
<td>( W_i )</td>
<td>the number of captioning terms for image ( I_i )</td>
</tr>
<tr>
<td>( B_i )</td>
<td>the number of blob-tokens in image ( I_i )</td>
</tr>
<tr>
<td>( D )</td>
<td>data matrix, ([D_{uw}D_{ub}])</td>
</tr>
<tr>
<td>( D_w )</td>
<td>image-to-term data matrix</td>
</tr>
<tr>
<td>( D_b )</td>
<td>image-to-blob-token data matrix</td>
</tr>
<tr>
<td>( T_{Corr} )</td>
<td>correlation-based association table</td>
</tr>
<tr>
<td>( T_{Cos} )</td>
<td>cosine-similarity association table</td>
</tr>
<tr>
<td>( d_w )</td>
<td>the ( i )-th column of the matrix ( D_w )</td>
</tr>
<tr>
<td>( d_b )</td>
<td>the ( i )-th column of the matrix ( D_b )</td>
</tr>
</tbody>
</table>

The correlation between terms and blob-tokens is computed based on their co-occurrence relation in the given annotated image set. Recall that each image in the annotated image set has a set of blob-tokens, as well as a set of annotated terms. We can represent each image by a vector of counts on terms and blob-tokens. If there are \( W \) possible terms and \( B \) possible blob-tokens, the entire annotated image set of \( N \) images can be represented by a data matrix \( D_{uw} \) \([D_{uw}d_{uw}]\). We now define two matrices: one is unweighted, the other is uniqueness-weighted as initial data representation.

**Definition 6 (Unweighted data matrix)** Given an annotated image set \( I = \{I_1, ..., I_N\} \) with the term set \( W \) and the blob-token set \( B \), the unweighted data matrix \( D_{uw} \) \([D_{uw}d_{uw}]\) is a \( N \)-by-\((W+B)\) matrix, where the \((i, j)\)-element of the \( N \)-by-\( W \) matrix \( d_{uw} \) is the count of term \( w_j \) in image \( I_i \), and the \((i, j)\)-element of the \( N \)-by-\( B \) matrix \( d_{ub} \) is the count of blob-token \( b_j \) in image \( I_i \).

The data matrix \( D \) is weighted according to the "uniqueness" of each term(blob-token). If a term appears only once in the image set, say with image \( I_i \), the term is only associated with the blob-tokens of \( I_i \). The more common a term is associated with the more blob-tokens. The uncertainty of finding the correct term-and-blob-token association goes up. In other words, common terms are "noisy". Similarly, these arguments hold for blob-tokens. The idea is to give higher weight to terms (blob-tokens) which are more "unique" in the training set, and low weights to noisy, common terms (blob-tokens).

**Definition 7 (Uniqueness weighting and weighted data matrix)** Given an unweighted data matrix \( D_{uw} \) \([D_{uw}d_{uw}]\). Let \( z_i \) \((y_i)\) be the number of images which contain the term \( w_j \) (the blob-token \( b_j \)). The weighted data matrix \( D = [D_{uw}D_{ub}] \) is constructed from \( D_{uw} \), where the \((i, j)\)-element of \( D \) \((D_{uw}) \), \( d_{uw} \) \((d_{ub}) \), is

\[
\begin{align*}
  d_{uw} &= \frac{d_{uw} \times \log N/z_i}{y_i} , \\
  d_{ub} &= \frac{d_{ub} \times \log N/y_i}{z_i} ,
\end{align*}
\]

where \( N \) is the total number of images in the set.

In the following, whenever we mention the data matrix \( D \), it will be always the weighted data
**Example 1** Let the annotated image set \(I = \{I_1, I_2\}\), with term set \(W = \{w_1, w_2, w_3\}\) (e.g., "boat", "sea", "sky") and blob-token set \(B = \{b_1, b_2, b_3, b_4\}\) (e.g., "wood-like-token", "sea-token", "sky-token", "blue-token"). Let image \(I_1\) has annotated words \(w_1, w_2\) and blob-tokens \(b_1, b_2\); Then, the corresponding data matrix:

\[
D_{uw} = [D_{uw,w_1}D_{uw,w_2}] = \begin{pmatrix}
1 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 0
\end{pmatrix}
\]

The weighted data matrix:

\[
D = [Dw|D_b]
\]

\[
= \begin{pmatrix}
\log(2) & \log(2) & 0 & 0 & \log(2) & 0 & 0 & 0 \\
0 & 0 & \log(2) & 0 & 0 & \log(2) & 0 & \log(2) \\
0 & 0 & 0 & \log(2) & 0 & 0 & \log(2) & \log(2)
\end{pmatrix}
\]

**Definition 8** (Method Corr: correlation-based association table) Let table \(T_{UN,\text{Corr}} = D_w^T D_b\). The correlation-based association table \(T_{\text{Corr}}\) is defined by normalizing each column of \(T_{UN,\text{Corr}}\) such that each column sum up to 1. Note that the \((i, j)\)-element of \(T_{\text{Corr}(i,j)}\) can be viewed as an estimate to \(p(w_i|b_j)\), the conditional probability of term \(w_i\) given blob-token \(b_j\).

**Example 2** The table \(T_{UN,\text{Corr}}\) of the data matrix in Example 1 is:

\[
D_w^T D_b = \begin{pmatrix}
\log(2) & \log(2) & 0 & 0 & \log(2) & 0 & 0 & 0 \\
0 & 0 & \log(2) & 0 & 0 & \log(2) & 0 & \log(2) \\
0 & 0 & 0 & \log(2) & 0 & 0 & \log(2) & \log(2)
\end{pmatrix}
\]

The correlation-based association table \(T_{\text{Corr}}\) by normalizing each column of \(T_{UN,\text{Corr}}\) is:

\[
\begin{pmatrix}
0.5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
\]

**Example 3** (Without doing weighting on \(D\)) Continue from Example 2, if we define a correlation table \(T_{UN,\text{Corr}}\) with the unweighted \(D_b\), i.e., we define \(F_{UN,\text{Corr}} = D_w^T D_{uw,b}\) and let \(F_{\text{Corr}}\) be \(F_{UN,\text{Corr}}\), with columns normalized which each sums to 1.

We have \(F_{\text{Corr}} = \begin{pmatrix}
0.5 & 0.33 & 0 & 0 \\
0.5 & 0.33 & 0 & 0 \\
0.5 & 0.33 & 1 & 1
\end{pmatrix}\).

Notice that \(T_{\text{Corr}}\) is not confused by the "noisy" blob-token "b" and would not annotate terms \(w_1\) and \(w_2\) for the image \(I_2\). On the other hand, \(F_{\text{Corr}}\) will have probability of 0.66 of annotating the wrong terms (\(w_1\) or \(w_2\)) for the image \(I_2\).

\(T_{\text{Corr}}\) measures the association between a term and a blob-token by the co-occurrence counts.

Another possible measurement could be to see how similar the overall occurrence pattern (over the training images) of a term and a blob-token is. Such occurrence patterns are in fact the columns of \(D_w\) or \(D_b\), and the similarity can be taken as the cosine value between pairs of column vectors.

**Definition 9** (Method Cos: cosine-similarity association table) Let the \(i\)-th column of the matrix \(D_w(D_b)\) be \(d_w(d_b)\). Let \(Cos_{ij}\) be the cosine similarity between column vectors \(d_w(d_b)\), which is:

\[
Cos_{ij} = \frac{d_w(d_b)^{T}d_w(d_b)}{|d_w(d_b)| \cdot |d_w(d_b)|}
\]

Let the table \(T_{UN,\text{Cos}}\) be a \(W\)-by-\(B\) matrix whose \((i, j)\)-element \(T_{UN,\text{Cos}(i,j)} = Cos_{ij}\). Normalize the columns of \(T_{UN,\text{Cos}}\) such that each column sums up to 1, and we get the cosine-similarity association table \(T_{\text{Cos}}\).

Note that the cosine-similarity table \(T_{\text{Cos}} = \tilde{D}_w^{T}\tilde{D}_b\) where \(\tilde{D}_w(\tilde{D}_b)\) is the matrix \(D_w(D_b)\) with each column normalized to unit length. Like the correlation-based association table, the \((i, j)\)-element of \(T_{\text{Corr}(i,j)}\) can also be viewed as an estimate to the conditional probability of term \(w_i\) given blob-token \(b_j\), \(p(w_i|b_j)\).

**Example 4** Continue from Example 1, we have the column-normalized matrix \([\tilde{D}_w(\tilde{D}_b)] = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0
\end{pmatrix}\).

Hence, the table \(T_{UN,\text{Cos}}\) is:

\[
\begin{pmatrix}
1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0
\end{pmatrix}
\]

and the cosine-similarity table \(T_{\text{Cos}}\) is:

\[
\begin{pmatrix}
0.5 & 0 & 0 & 0 \\
0.5 & 0 & 0 & 0 \\
0 & 0 & 1 & 1
\end{pmatrix}
\]

The low rank representation by Singular Value Decomposition (SVD) reveals latent semantics in a given matrix [27]. Specifically, SVD is used to clean up the observed (noisy) term-document matrix. They showed that estimating the term-term correlation after SVD gives better retrieval performance than the one of the observed (noisy) term-document matrix. In this paper, we propose to use SVD to suppress the noise in the data matrix before learning the association.

**Definition 10** (Singular Value Decomposition) SVD decomposes a given matrix into a product of
three matrices U, Λ, V^T. That is, X = UAV^T, where
U=[u_1, ..., u_n] and V = [v_1, ..., v_m] are orthonormal,
and Λ is a diagonal matrix. Note that u_i (v_i) are
columns of the matrix U(V). Let Λ=diag(σ_1, ..., σ

Note that the SVD of a matrix X can also be
written as

\[ X = \sum_{j=1}^{\text{rank}(X)} \sigma_j u_j v_j^T \]

(2)

The latter terms in the summation contribute less
to X, as the corresponding σ_j's become smaller and
smaller.

The following example shows that SVD is used to
clean up noise and reveals informative structure
in a matrix, by omitting the smallest terms in the
summation (equation 2).

Example 5 Let X be
\[
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

SVD gives matrices U, Λ, V such that X = UAV^T.
Cleaning up X by representing it using only the
first two σ's, we get the clean up version X

\[
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

the cleaned up version X shows

\[
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

the structure of the original X which is hidden by
the noise before applying SVD.

We keep only the first r terms of Equation (2) to
preserve the 90% variance of the distribution. More
specifically, let S90=0.9 \times \sum \sigma_j^2 \Rightarrow S_{90}
then r is de-
determined as

\[ r = \arg \min_j \sum \sigma_j^2 \leq S_{90} \].

In other words,
r is determined capturing 90% of the total variance.

In the following, we denote the data matrix after
SVD as D_{svd}=D_{w_{svd}}D_{b_{svd}}. Now, let us define
correlation-based association tables with SVD.

Definition 11 (Method SvdCorr) correlation-based
association table with SVD) Method SvdCorr
generates the correlation-based association table
T_{svdcorr} following the procedure outlined in Def-
nition 8, but instead of starting with the weighted
data matrix D, here the matrix D_{svd} is used.

Definition 12 (Method SvdCos) cosine-similarity
association table with SVD) Method SvdCos
generates the cosine-similarity association table
T_{svdcos} following the procedure outlined in Definition
9, but instead of starting with the weighted data
matrix D, here the matrix D_{svd} is used.

Given an association table by the proposed
methods, an image is annotated by the following
algorithm.

Algorithm 1 (Association table based annotation)
Given an association table T_{W×10} (W: total number
of terms; B: total number of blob-tokens), and also
the number of captioning terms needed k, an image
with 1 blob-tokens set B' = {b', ..., b'}, can be
captioned through the following steps:
1. Form a query vector q = [q_1, ..., q_n], where q_i is
the count of the blob-token b_i in the set B'.
2. Compute the term-likelihood vector p = Tq, where
p=[p_1, ..., p_B]^T, and p_i is the predicted likelihood
of the term w_i.
3. If k captioning terms are to be generated, select
the terms corresponding to the top k p_i's in the
p vector.

5. Experimental Result

The experiments are performed on 10 Corel
image data sets. Each data set contains about 5200
training images and 1750 testing images. The sets
cover a variety of themes ranging from urban
scene to natural scene, and from artificial objects
like jet/plane to animals. Each image has in average 3 annotated terms and 9 blobs.

We apply G-means and uniqueness weighting to
show the effects of clustering and weighting. We
compare our proposed methods, namely Corr, Cos,
SvdCorr and SvdCos, with the state-of-the-art
machine translation approach [10], namely EM
approach as the comparison baseline. Each method
constructs an association table as an estimated
conditional probability of a term w_i given a blob-
token b_j, \{p(w_i,b_j)\}. These association tables are
then used in Algorithm 1 for annotation. Parti-
cularly, we would like to answer the following
questions:
1. How important is the clustering algorithm?
2. How does the proposed "uniqueness" weighting
effect the performance?
3. Which proposed method is best?
We measure the annotation accuracy on each test image as the percentage of correctly predicted words as a measurement of the quality of the association table [10]. Given an image with m true annotated terms (given by human annotators), we also predicted m terms for this image using Algorithm 1. The accuracy of the annotation is defined as $S_{\text{correct}} = \frac{m_{\text{correct}}}{m}$, where $m_{\text{correct}}$ is the number of correct terms annotated for the new image. The overall performance is expressed by the average accuracy over all images in a (test) set.

In the rest of the paper, we denote an experiment result by the design alternatives chosen along the process which generates the result. That is, each process is denoted with a string in the following format: "[method]-[nTokens]-[weighted]" with 3 fields to fill in the specific choices made at the 3 stages of the process. The choices for each stages are:

- field [method]: the 5 methods, EM, Corr, Cos, SvdCorr, and SvdCos. We also use All to denote all proposed methods (i.e., all except EM).
- field [nTokens]: 500, where the number of blob-tokens fixed at 500; AdaptB, where the number of blob-tokens is determined adaptively.
- field [weighted]: W, if the data matrix is weighted; UW, otherwise.

We first evaluate the performance of the proposed 4 methods with unweighted 500 blob-token data sets. Table 2 shows the annotation accuracy of the proposed methods and the baseline algorithm [10] denoted as EM-B500-UW (which means EM is applied to an unweighted matrix, denoted as UW, in which the number of blob tokens is 500, denoted as B500). With fixed 500 blob tokens, method Cos-B500-UW achieves an improvement around 2% in absolute accuracy over EM-B500-UW.

The adaptive blob-token generation improves the annotation accuracy shown, in Table 3. Cos-AdaptB-UW shows 9.4% absolute accuracy improvement over EM-B500-UW, the baseline method. Using the G-means algorithm (Section 3), the numbers of blob-tokens found for the 10 training set are all less than 500, ranging from 339 to 485, mostly around 400. In fact, we found that the improvement is not only on EM method, but also on our proposed methods. The annotation accuracy of All-B500-UW as well as EM-B500-UW are improved around 7% with adaptively generated blob-tokens.

Figure 4(a) illustrates the improvement of all proposed methods over the EM-B500-UW. Figure 4(b) compares the average annotation accuracy of fixed number of blob-tokens of 10 data sets versus the one of adaptively generated number of blob-tokens.

Table 4 and Table 5 illustrate the annotation accuracy with "uniqueness" on B500 and AdaptB data sets, respectively. After applying the "uniqueness" weighting, the 4 proposed methods on the fixed number of blob-token data perform about 2% better and the performances on the adaptive number of blob-token data gives about 9% improve-

<table>
<thead>
<tr>
<th>Dataset ID</th>
<th>EM</th>
<th>Corr</th>
<th>Cos</th>
<th>SvdCorr</th>
<th>SvdCos</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>0.2199</td>
<td>0.2196</td>
<td>0.2445</td>
<td>0.2216</td>
<td>0.1810</td>
</tr>
<tr>
<td>002</td>
<td>0.2177</td>
<td>0.2183</td>
<td>0.2464</td>
<td>0.2212</td>
<td>0.1989</td>
</tr>
<tr>
<td>003</td>
<td>0.2279</td>
<td>0.2282</td>
<td>0.2423</td>
<td>0.2278</td>
<td>0.1881</td>
</tr>
<tr>
<td>004</td>
<td>0.1925</td>
<td>0.1941</td>
<td>0.2118</td>
<td>0.1950</td>
<td>0.1621</td>
</tr>
<tr>
<td>005</td>
<td>0.2280</td>
<td>0.2299</td>
<td>0.2594</td>
<td>0.2336</td>
<td>0.2136</td>
</tr>
<tr>
<td>006</td>
<td>0.2065</td>
<td>0.2072</td>
<td>0.2410</td>
<td>0.2085</td>
<td>0.1920</td>
</tr>
<tr>
<td>007</td>
<td>0.2065</td>
<td>0.2085</td>
<td>0.2312</td>
<td>0.2118</td>
<td>0.1714</td>
</tr>
<tr>
<td>008</td>
<td>0.2290</td>
<td>0.2308</td>
<td>0.2555</td>
<td>0.2314</td>
<td>0.1961</td>
</tr>
<tr>
<td>009</td>
<td>0.2223</td>
<td>0.2233</td>
<td>0.2414</td>
<td>0.2236</td>
<td>0.1916</td>
</tr>
<tr>
<td>010</td>
<td>0.2324</td>
<td>0.2332</td>
<td>0.2586</td>
<td>0.2337</td>
<td>0.2078</td>
</tr>
<tr>
<td>Average</td>
<td>0.2213</td>
<td>0.2193</td>
<td>0.2432</td>
<td>0.2206</td>
<td>0.1902</td>
</tr>
</tbody>
</table>
Table 3 Annotation accuracy on "AdaptB-UW" data sets

<table>
<thead>
<tr>
<th>DataSet ID</th>
<th># of BT</th>
<th>EM</th>
<th>Corr</th>
<th>Cos</th>
<th>SvdCorr</th>
<th>SvdCos</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>339</td>
<td>0.2786</td>
<td>0.2548</td>
<td>0.3076</td>
<td>0.2622</td>
<td>0.3021</td>
</tr>
<tr>
<td>002</td>
<td>416</td>
<td>0.3000</td>
<td>0.2791</td>
<td>0.3204</td>
<td>0.2875</td>
<td>0.2227</td>
</tr>
<tr>
<td>003</td>
<td>392</td>
<td>0.3045</td>
<td>0.2872</td>
<td>0.3121</td>
<td>0.2917</td>
<td>0.2193</td>
</tr>
<tr>
<td>004</td>
<td>438</td>
<td>0.2779</td>
<td>0.2583</td>
<td>0.3001</td>
<td>0.2959</td>
<td>0.2242</td>
</tr>
<tr>
<td>005</td>
<td>495</td>
<td>0.2845</td>
<td>0.2854</td>
<td>0.3232</td>
<td>0.2865</td>
<td>0.2581</td>
</tr>
<tr>
<td>006</td>
<td>353</td>
<td>0.3055</td>
<td>0.2751</td>
<td>0.3239</td>
<td>0.2775</td>
<td>0.2243</td>
</tr>
<tr>
<td>007</td>
<td>433</td>
<td>0.2873</td>
<td>0.2665</td>
<td>0.3041</td>
<td>0.2685</td>
<td>0.2078</td>
</tr>
<tr>
<td>008</td>
<td>384</td>
<td>0.3073</td>
<td>0.2833</td>
<td>0.3306</td>
<td>0.2870</td>
<td>0.2271</td>
</tr>
<tr>
<td>009</td>
<td>386</td>
<td>0.2808</td>
<td>0.3030</td>
<td>0.2978</td>
<td>0.2667</td>
<td>0.2028</td>
</tr>
<tr>
<td>010</td>
<td>386</td>
<td>0.3261</td>
<td>0.3270</td>
<td>0.3346</td>
<td>0.3037</td>
<td>0.2445</td>
</tr>
<tr>
<td>Average</td>
<td>402</td>
<td>0.2963</td>
<td>0.2751</td>
<td>0.3157</td>
<td>0.2802</td>
<td>0.2232</td>
</tr>
</tbody>
</table>

Figure 4 Comparison of annotation accuracy on AdaptB vs. B500 data sets

(a) EM-B500-UW vs. ALL-AdaptB-UW
(b) All(EM)-B500-UW vs. All(EM)-AdaptB-UW

Table 4 Annotation accuracy on "B500-W" data sets

<table>
<thead>
<tr>
<th>Dataset ID</th>
<th>Corr</th>
<th>Cos</th>
<th>SvdCorr</th>
<th>SvdCos</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>0.2439</td>
<td>0.2445</td>
<td>0.2366</td>
<td>0.2219</td>
</tr>
<tr>
<td>002</td>
<td>0.2446</td>
<td>0.2464</td>
<td>0.2433</td>
<td>0.2274</td>
</tr>
<tr>
<td>003</td>
<td>0.2406</td>
<td>0.2423</td>
<td>0.2499</td>
<td>0.2202</td>
</tr>
<tr>
<td>004</td>
<td>0.2137</td>
<td>0.2118</td>
<td>0.2103</td>
<td>0.1933</td>
</tr>
<tr>
<td>005</td>
<td>0.2567</td>
<td>0.2594</td>
<td>0.2559</td>
<td>0.2406</td>
</tr>
<tr>
<td>006</td>
<td>0.2358</td>
<td>0.2410</td>
<td>0.2364</td>
<td>0.2265</td>
</tr>
<tr>
<td>007</td>
<td>0.2273</td>
<td>0.2312</td>
<td>0.2304</td>
<td>0.2062</td>
</tr>
<tr>
<td>008</td>
<td>0.2517</td>
<td>0.2556</td>
<td>0.2520</td>
<td>0.2318</td>
</tr>
<tr>
<td>009</td>
<td>0.2392</td>
<td>0.2414</td>
<td>0.2400</td>
<td>0.2239</td>
</tr>
<tr>
<td>010</td>
<td>0.2576</td>
<td>0.2586</td>
<td>0.2543</td>
<td>0.2576</td>
</tr>
<tr>
<td>Average</td>
<td>0.2417</td>
<td>0.2432</td>
<td>0.2408</td>
<td>0.2230</td>
</tr>
</tbody>
</table>

ment. As in case of the B500 data set, applying uniqueness weighting on the AdaptB data set also raises the performance of methods Corr, SvdCorr and SvdCos to the level of Cos. Method Corr and SvdCorr even outperform Cos. The uniqueness weighting improves the performance of all proposed methods except Cos (Figure 4(b)). Note that the method Cos always perform better than the baseline method. Intuitively, the uniqueness weighting multiplies each column $d_w$ or $d_b$ by some constant, which effectively changes the lengths of each column. However, the angles between them remain unchanged, so do the cosine values which are measured by the Cos method.
Table 5 Annotation accuracy on "AdaptB-W" data sets

<table>
<thead>
<tr>
<th>Dataset ID</th>
<th>Corr</th>
<th>Cos</th>
<th>SvdCorr</th>
<th>SvdCos</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>0.3062</td>
<td>0.3076</td>
<td>0.3076</td>
<td>0.2618</td>
</tr>
<tr>
<td>002</td>
<td>0.3171</td>
<td>0.3204</td>
<td>0.3184</td>
<td>0.2685</td>
</tr>
<tr>
<td>003</td>
<td>0.3176</td>
<td>0.3121</td>
<td>0.3202</td>
<td>0.2585</td>
</tr>
<tr>
<td>004</td>
<td>0.3081</td>
<td>0.3031</td>
<td>0.3071</td>
<td>0.2616</td>
</tr>
<tr>
<td>005</td>
<td>0.3218</td>
<td>0.3232</td>
<td>0.3224</td>
<td>0.2901</td>
</tr>
<tr>
<td>006</td>
<td>0.3248</td>
<td>0.3239</td>
<td>0.3296</td>
<td>0.2833</td>
</tr>
<tr>
<td>007</td>
<td>0.3170</td>
<td>0.3041</td>
<td>0.3158</td>
<td>0.2636</td>
</tr>
<tr>
<td>008</td>
<td>0.3293</td>
<td>0.3006</td>
<td>0.3354</td>
<td>0.2809</td>
</tr>
<tr>
<td>009</td>
<td>0.2986</td>
<td>0.2978</td>
<td>0.2999</td>
<td>0.2579</td>
</tr>
<tr>
<td>010</td>
<td>0.3362</td>
<td>0.3346</td>
<td>0.3440</td>
<td>0.2844</td>
</tr>
<tr>
<td>Average</td>
<td>0.3180</td>
<td>0.3157</td>
<td>0.3200</td>
<td>0.2712</td>
</tr>
</tbody>
</table>

(a) B500 data set  
(b) AdaptB data set

Figure 5 Annotation accuracy of unweighted vs. weighted data sets

Table 6 Average recall, precision, and the number of used words

<table>
<thead>
<tr>
<th># of used words</th>
<th>EM</th>
<th>Corr</th>
<th>Cos</th>
<th>SvdCorr</th>
<th>SvdCos</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Recall</td>
<td>0.0425</td>
<td>0.1718</td>
<td>0.1820</td>
<td>0.1567</td>
<td>0.2128</td>
</tr>
<tr>
<td>Avg. Precision</td>
<td>0.0411</td>
<td>0.1131</td>
<td>0.1445</td>
<td>0.1197</td>
<td>0.2079</td>
</tr>
</tbody>
</table>

Figure 5 shows the effect of the proposed "uniqueness" weighting on the captioning accuracy comparing B500 data sets and AdaptB data sets. We also observed that weighting does not affect the result of EM method.

Another measurement of the performance is the recall and precision values for each word. Given a word w, let the set Rw contains r test images captioned with the word w by the method we are evaluating. Let r' be the actual number of test images that have the word w (set R'w), and r' be size of the intersection of Rw and R'w. Then, the precision of word w is r'/r, and the recall is r'/r'. Note that some words could never be used in the automatic captioning, if they are never used or are always used for the wrong images (un-annotatable words). We prefer a method which has fewer unused words, since it could generalize better to unseen images. Table 6 shows that the proposed methods use two to three times more predictable words on average than the baseline EM approach dose. In Figure 6 which illustrates recall and precision for each word, SvdCorr and SvdCos show more words are located in non-zero points than Corr and Cos. EM approach captions the frequent words with high precision and recall, but misses many words compare to SvdCorr/SvdCos. That is, EM approach is biased to the training set.

Figure 7 illustrates recall and precision scores of the top 20 frequent words in the test set. SvdCorr
(white bars) gives more general performance than baseline EM approach (black bar).

As an example of how well the captioning is performed, we show annotation words for the image in Figure 1(a) and Figure 1(b) in Table 7. EM-B500-UW and SvdCorr-AdaptB-W both give “sky”, “cloud”, “sun” and “water” for the image in Figure 1(a). EM-B500-UW gives “grass”, “rocks”,

<table>
<thead>
<tr>
<th>Table 7 Annotation examples of the proposed methods</th>
<th>Figure 1(a)</th>
<th>Figure 1(b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM-B500-UW</td>
<td>sun, clouds, sky, water</td>
<td>grass, rocks, sky, snow</td>
</tr>
<tr>
<td>Corr-AdaptB-W</td>
<td>sun, sky, sunset, clouds</td>
<td>grass, cat, tiger, tree</td>
</tr>
<tr>
<td>Cos-AdaptB-W</td>
<td>sun, sunset, sky, sea</td>
<td>grass, tiger, cat, leaves</td>
</tr>
<tr>
<td>SvdCorr-AdaptB-W</td>
<td>sun, clouds, sky, water</td>
<td>grass, cat, tiger, water</td>
</tr>
<tr>
<td>SvdCos-AdaptB-W</td>
<td>sunset, sun, sea, light</td>
<td>tiger, grass, cat, bengal</td>
</tr>
<tr>
<td>True caption</td>
<td>sea, sun, sky, waves</td>
<td>cat, forest, grass, tiger</td>
</tr>
</tbody>
</table>
"sky" and "snow" for the image in Figure 1(b), while \textit{SvdCorr-AdaptB-W} gives "grass", "cat", "tiger", and "water". Although the captions do not match the truth perfectly, they describe the content quite well. This indicates that the "truth" caption may be just one of the many ways to describe the image.

We summarize the results of our experiments as follows:

- Method \textit{SvdCorr} has the best captioning accuracy over the 10 testing sets.
- The uniqueness weighting has no effect on method Cos (neither improve nor deteriorate).
- Methods Cos and SvdCorr have the same level of performance as Cos after applying the uniqueness weighting.
- Setting the size of the blob-token set is crucial for achieving good captioning accuracy. All proposed methods and the baseline method improve their performance, when working on the \textit{AdaptB} data sets.

6. Discussions

The task of captioning images automatically is difficult, due to factors ranging from the property of the data set and the proposed framework for solving this problem (in our case, we caption an image via captioning the image's blobs).

In our experiments, we found that the Corel data set has several properties which introduce noise to our process. The ideal situation is each term has unique visual counterpart (blob-token), such as "cat" is always a yellow blob, and "sky" is always a blue blob. However, in the data set, a general term like "cat" appears with specific terms "tiger" or "lion" which have different blobs (one with strips, one without); and, "sky" is not always blue, there are sky during sunset which is yellow or orange.

Since our proposed methods caption an image through captioning its constituent blob-tokens, the quality of the blob-tokens in an image are critical. In practice, we have more blob-tokens than the captioning terms for each image. The extra blob-tokens may correspond to small objects in the background, or common objects such as "sky", "sea" which are captioned by human experts for some images but are not captioned for some other images which also contain the sky or sea. These uncaptioned blob-tokens introduce noise to our proposed methods and effect the performance.

For our experiments, we found that our proposed methods achieve a 45% relative improvement over the state-of-the-art EM approach. Why do the proposed methods perform better? What else can we do to do even better? The proposed methods weigh different terms and blob-tokens according to their power of discrimination (Definition 7). If a term (blob-token) is common among many images, it is likely to be mixed up with many different blob-tokens (terms). As a result, it should get lower weight, to constrain the possibility of our estimate being messed up by it. In other words, weighting suppresses the noise in the data matrix. The ongoing work is to incorporate this idea of weighting into the EM approach, which we suspect may as well boost its performance.

Despite the success of our proposed methods, we applied context-aware captioning to further improve the performance, where the relation among the blobs of an image are taking into consideration. For example, if an image is a seascape scene which contains three blob-tokens, with a blob-token suggesting the term "sea" and another suggesting "sky". Then, the term "table" is less likely to be correct than the term "boat" for the third blob-token (e.g., a wood-like blob-token which is shared by both "table" and "boat"), even the term "table" has greater likelihood than "boat" as indicated in the term-likelihood vector. We model this inter-blob-token relation by a term-term association table, which is estimated based on the co-occurrence of the terms in the annotated image set. It successfully boosts the performance of the inferior ones of our proposed methods to the same level of the best proposed method. However, surprisingly, the proposed context-aware captioning does not boost the best proposed method further. This may due to the inherent limitation of the correlation-based approach which uses only co-occurrence information. We believe adding extra information or assumptions into the process might help.
7. Conclusions

In this paper, we studied the problem of automatic image captioning. The problems we are interested in are: "Given an image, give terms which describe its content." "Find images which can be described by the word "tiger". The technique developed will be useful for image retrieval applications. Our main contribution is the proposed correlation-based methods (Corr, Cos and SvdCorr) that consistently outperform the state of the art (EM) by up to a 45% relative improvement in captioning accuracy. SvdCos shows the best performance with recall and precision measurement that is SvdCos is general to unseen images. Specifically, in this paper,

- we do thorough experiments on large datasets of different image content styles, and examine all possible combinations of the proposed techniques for improving captioning accuracy;
- the proposed uniqueness weighting scheme on terms and blob-tokens boosts the captioning accuracy;
- our improved, "adaptive" clustering (to form blob-tokens) consistently leads to performance gains;
- dimension reduction by SVD reveals latent structures between visual vocabulary and content vocabulary so that SvdCorr and SvdCos are generalized for captioning of the unseen images.

The proposed techniques can be applied to other domains. For example, given a set of microscopic images with descriptions (e.g. the location of the cells, the symptoms of some diseases shown in the images) [29], the proposed methods can automatically give medical suggestions given a microscopic image of a new patient.

References

[18] Lavrenko, V., Manmatha, R. and Jeon, J., "A

Jia-yu Pan
Jia-Yu Pan received an M.S. degree from National Taiwan University, and a B.S. degree from National Chiao Tung University, Taiwan, both in Computer Science. He is currently a Ph.D. student at Carnegie Mellon University, working on video data mining and multimedia correlation discovery.

Pinar Duygulu
Pinar Duygulu has received her BSc, MSc and PhD degrees from Department of Computer Engineering at Middle East Technical University, Ankara, Turkey. During her PhD, she was a visiting scholar at University of California at Berkeley. After being a post-doctoral researcher at Informedia Project at Carnegie Mellon University, recently she joined to Department of Computer Engineering at Bilkent University, Ankara, Turkey. She is the co-director of RETINA Vision and Learning Group at Bilkent. She received the best paper in Cognitive Vision award at European Conference on Computer Vision in 2002. Her current research interests include computer vision and multimedia data mining, specifically object recognition and semantic analysis of large image and video collections.

Christos Faloutsos
Christos Faloutsos is a Professor at Carnegie Mellon University. He has received the Presidential Young Investigator Award by the National Science Foundation (1989), four "best paper" awards, and several teaching awards. He is a member of the executive committee of SIGKDD; he has published over 120 refereed articles, one monograph, and holds four patents. His research interests include data mining for streams and networks, fractals, indexing methods for spatial and multimedia bases, and data base performance.