Ranking and Retrieval of Image Sequences from Multiple Paragraph Queries

Gunhee Kim
Seoul National University
gunhee@cs.cmu.edu

Seungwhan Moon
Carnegie Mellon University
seungwhm@cs.cmu.edu

Leonid Sigal
Disney Research Pittsburgh
lsigal@disneyresearch.com

Abstract

We propose a method to rank and retrieve image sequences from a natural language text query, consisting of multiple sentences or paragraphs. One of the method's key applications is to visualize visitors' text-only reviews on TRIPADVISOR *or* YELP*, by automatically retrieving the most illustrative image sequences. While most previous work has dealt with the relations between a natural language sentence and an image or a video, our work extends to the relations between paragraphs and image sequences. Our approach leverages the vast user-generated resource of blog posts and photo streams on the Web. We use blog posts as text-image parallel training data that co-locate informative text with representative images that are carefully selected by users. We exploit large-scale photo streams to augment the image samples for retrieval. We design a latent structural SVM framework to learn the semantic relevance relations between text and image sequences. We present both quantitative and qualitative results on the newly created* DISNEYLAND *dataset.*

1. Introduction

Textual and visual forms of communication are complementary and synergetic in many aspects (*e.g.* news articles, blogs). A system that can take passages of free form text and automatically illustrate them with relevant imagery would constitute a significant step forward toward joint understanding of natural language descriptions and visual content of images. Recently there has been steady progress of chipping away at this challenging problem by either one-sentence (*i.e.* subject-verb-object style) text generation methods [7, 8, 9, 17, 24, 29] for image description, or image/video retrieval from sentence queries [10, 37]. In this paper, we make the next leap towards retrieving a sequence of images (as opposed to a single image) to illustrate much longer in terms of content, passages of text, that may consist of multiple sentences or even paragraphs (as opposed to a single sentence). One of the challenges, however, is obtaining appropriate text-image parallel corpus from which semantic relationships between text and images can be learned.

Figure 1. A depiction of our problem statement with *Disneyland* examples. We leverage (a) blog posts to learn the mapping between sentences and images, and (b) photo streams to augment the image samples. (c) Our objective is to rank and retrieve image sequences that best describe a given text query consisting of multiple sentences or paragraphs.

As social media sites continue to proliferate, a growing number of individuals willingly share their own experiences in the form of images, videos, or text, over a multitude of Web platforms. For example, many people who visit *Disneyland* take large streams of photos about their unique experience, and upload them onto photo hosting sites such as FLICKR. Some of more enthusiastic users also craft blog posts to document their trips on weblog publishing sites such as BLOGGER or WORDPRESS, or evaluate their experiences on review sites like TRIPADVISOR or YELP.

In this paper, we address a problem of *ranking and retrieval of image sequences for a natural language query of multiple sentences*, by leveraging a large corpus of online images and text that describe common events or activities in different forms. Fig.1 illustrates an intuition of our problem statement with an example of tourism (*e.g.* visiting *Disneyland*). Given a text query consisting of multiple sentences or paragraphs, our goal is to automatically retrieve image se-

*This work has been done when Gunhee Kim was a postdoctoral researcher, and Seungwhan Moon was an intern at Disney Research.

quences that best describe the essence of the query text. In order to more densely populate image samples, we also exploit a large set of photo streams; each stream is a sequence of images that are taken by a single user in one day. Our approach is developed based on the structural ranking SVM with latent variables (*e.g.* [11, 18, 35]), in order to learn the relevance relation between text and image sequences.

Our research can enable a number of web service applications, especially in the domain of tourism. For example, we can visualize the visitors' text-only reviews on TRIPADVISOR or YELP, by automatically retrieving the most illustrative image sequences from the Web. This application is significant because general users can understand key concepts and sentiment much easier and quicker with images. Moreover, the visuals are more useful for a new visitor. For example, a user who has never visited Disneyland may not fully understand the reviews about *Bug's land*, without illustration of attractions, which our approach can generate.

As a problem domain, we focus on a theme park, specifically *Disneyland*, because it is easy to obtain abundant visual and textual data. However, our approach and problem formulation are much broader and are applicable to any domain that has a large set of blog posts with images (more broadly, any mixed image and text media). A concrete example is tourist websites that discuss or review museums, restaurants, cities or countries. In such setting, representative illustrations can be created for the flow of sentiments in the reviews. Our approach is unsupervised and is applicable to any of these domains if data is available.

We evaluate the image retrieval performance of our method on a newly collected *Disneyland* dataset, which consists of more than 10K blog posts with 120K associated images, and 6K photo streams of more than 540K unique images. We present comprehensive empirical studies comparing the retrieval accuracies of image sequences between five text segmentation, three text description, and two text-to-image embedding methods and their combinations. Our approach using latent structural SVM can efficiently integrate multiple algorithmic outputs in a unified way. We also perform the visualization of users' reviews on TRIPADVISOR or YELP, and evaluate them using crowdsourcing-based user studies via Amazon Mechanical Turk.

1.1. Related work

We now discuss some representative works that learn relations between images and natural language text.

Sentence generation from images/videos. The goal of this line of work is to automatically create or retrieve a concise descriptive sentence for a given image [8, 9, 17, 24, 34]. Among them, [8] and [9] are the most relevant to ours because their methods directly leverage a large collection of raw (possibly noisy) online data, such as multimodal database of news articles (images and their captions) in [8] and Flickr images annotated with noisy tags, titles, and descriptions in [9]. In our work, in addition to Flickr photo streams, we exploit blog posts and consumers' reviews, which have not been explored prior. Moreover, our work focuses on the extended problem of retrieving image sequences for a query of multiple sentences.

Mapping between images and text. Prior work has also looked at learning a mapping between sentences and images or retrieving one from the other (*e.g.* [7, 10, 29, 37]). The main focus in this line of work is on definition of a common semantic space that embeds both images and sentences. Some successful ideas include triplets of objects, actions, and scenes in [7], kernel canonical correlation analysis in [10], dependency tree-recursive neural networks (DT-RNN) in [29], and conditional random field models on abstract scenes in [37]. The key novelty of our work is that we focus on the relation between multiple paragraphs and image sequences, instead of between sentences and images.

Recently, multimodal recurrent neural networks [14, 16, 29] have been extensively used for mapping between text and images. Our latent structural SVM framework is appealing because of its flexibility that enables us to learn the weights of combinations of different model components in a unified way, including text segmentations, text descriptors, and text-to-image mapping methods.

Image/video retrieval from structured queries. This direction of work goes beyond conventional keyword-based image/video retrieval, and addresses structured queries. Some notable examples include video search for a sentence in the context of autonomous driving [22], image ranking and retrieval based on visual phrases [25], multiple-attribute queries [27], and graph-structured object queries [18]. The work of [28] proposed a method of merging three different query modalities (*e.g.* text, sketch, and images) into a common semantic space for image retrieval. Our work is unique in two aspects; first, our query structure is natural language paragraphs, and second, the retrieval targets are image sequences rather than individual images.

One of the most relevant works is [13], which also develops a method for automated story picturing. However, there are key differences. First, given a query passage, we aim to retrieve image sequences that emphasize the progression of the passage, as opposed to similar image retrieval. Second, our approach leverages unstructured online blogs and photo streams, as opposed to dataset created by experts.

1.2. Contributions

(1) To the best of our knowledge, this work is the first to address the ranking and retrieval of image sequences for natural language queries of multiple paragraphs. We extend both input and output to more complex forms in relation to previous research: paragraphs instead of sentences and image sequences instead of individual images.

(2) We develop an image sequence retrieval method, built upon a structural ranking SVM with latent variables. We show that our method can flexibly incorporate different pieces of information about the text and image structure.

(3) We evaluate our method with a large unstructured *Disneyland* dataset, consisting of 10K blog posts with 120K associated images, and 6K photo streams of 540K images. With quantitative evaluation and user studies via Amazon Mechanical Turk, we show our approach is practical in visualizing natural language text written by actual users.

2. Problem Formulation

We have three types of input data. The first input is a set of visitors' blog posts $\mathcal{B} = \{B^1, \cdots, B^N\}$. We assume that each blog post B^n consists of a sequence of images and associated text $B^n = \{(I_1^n, T_1^n), \cdots, (I_{|B^n|}^n, T_{|B^n|}^n)\}$. The set of blog posts $\mathcal{B}$ is used as an image-text parallel corpus for training, from which we can learn a joint image-text embedding into a common latent space.

The second input is a large set of visitors' photo streams $\mathcal{P} = \{P^1, \cdots, P^L\}$. We define a photo stream as a set of images that are taken in sequence by the same photographer within a single day. The main use of photo streams is to populate image samples for retrieval. We embed photo streams in the same latent space using the transformation learned from the blog data, and then for a given query text we can also return the images from photo streams.

The third input is a set of text-only posts $\mathcal{Q}$ where each post $Q \in \mathcal{Q}$ consists of multiple sentences or paragraphs. They include users' reviews or blog posts without images. We use $\mathcal{Q}$ as a set of text queries.

We formulate the retrieval of image sequences for a multiple paragraph query as follows. Given a query text $Q \in \mathcal{Q}$, we rank a set of image sequences, $\mathcal{S} = \{(S^1, w^1), \cdots, (S^K, w^K)\}$ where S^k is the k-th ranked image sequence, w^k is a ranking score, and K is the number of sets to retrieve. Each sequence S^k consists of images of blog posts $\mathcal{B}$ or photo streams $\mathcal{P}$. We assume that the size of a retrieved sequence S^k, denoted κ^k, is a user input, because it is rather subjective to individual's preference. For instance, some blog authors would upload tens of images for a short blog post, while others would be more verbose and use only a few images per post.

2.1. Text-Image Parallel Corpus

We assume that each image in a blog post is semantically associated with some portion of the text in the same post. The challenge in creating a text-image parallel training corpus is that the text in a blog post is often unstructured, and thus the canonical association between text and images is unknown a priori. Acquiring annotator labels is one option, but is not scalable to a large corpus. Therefore, our approach is to segment the text of a post such that each text segment is coherent in semantics, and let each image have some association with each of the text segments. We then use text-to-image block distance to determine the varying degree of image-text association (*i.e.* a text segment closer to an image has a higher degree of association than other text segments located farther in a document). Once the text segmentation and image-text association is obtained, each blog post B^n can be decomposed into a sequence of images and associated text: $B^n = \{(I_1^n, T_1^n), \cdots, (I_{|B^n|}^n, T_{|B^n|}^n)\}$.

2.2. Text Segmentation

We assume that a blog author augments, with images, text segments of the post in a semantically meaningful manner. The purpose of text segmentation is therefore to divide the input text into coherent groups of sentences such that each segment is expected to be associated with a single image. We apply several automatic text segmentation methods in NLP literature [5, 6, 32], based on its syntactic structure or semantic distribution. Each of the resulting segments obtained from the following methods represents either a sequence of sentences with the highest content centrality and coherence, or an individual topic that is latent in the post. We implement one syntactic segmentation (1) and four semantic segmentation methods (2)–(5). Semantic-based segmentation methods represent each sentence in terms of its semantic centrality, and group a sequence of sentences such that each segment maintains its coherency.

(1) **Paragraph tokenizer**. One of simplest segmentation methods is to divide the passage by its syntactic structure, such as by paragraphs. Each paragraph often carries a unique topic or event, thus likely to be associated with an image. We apply a standard paragraph tokenizer (NLTK [23]) that uses rule-based regular expressions to detect paragraph divisions.

(2) **Latent Semantic Analysis (LSA)**. The LSA applies the singular value decomposition (SVD) to obtain the concept dimension of sentences [19, 30]. Assuming that a passage is composed of multiple concepts or topics, each of which is represented by a few terms in the passage, the LSA based method recursively finds the most representative sentence or a group of sentences (segment) that maximizes the inter-sentence similarity value for each topic (*e.g.* the most prominent topic boundary) [32, 5].

(3) **LexRank**. The LexRank algorithm detects key sentences in the text based on their lexical centrality [2, 6]. We construct a graph by creating a vertex for each sentence in the blog, and connecting as edges the semantically similar sentences using the intra-sentence cosine similarity of TF-IDF vectors. We obtain the top sentences by their semantic importance that is estimated via random walk and eigenvector centrality. For each top centroid sentence, we build a text segment such that its boundary is within the equal sentence distance between two centroids.

(4-5) **Summary-based LSA and LexRank**. The LSA and LexRank algorithm can perform not only segmentation but also summarization [6, 15]. Thus, using the two base algorithms, we jointly perform segmentation and summarization, and treat each resulting summary sentence as a segment to be associated with an image. That is, while the segmentation (2)–(3) represents each segment as multiple sentences, the segmentation (4)–(5) selects only a single sentence that has the highest semantic centrality from each text segment, and can remove less representative portions of the segments.

2.3. Text Description

After performing text segmentation, we extract features from each text segment. We first pre-process the text corpus by normalizing each tokenized sentence and removing stop words. We use the three standard text descriptors.

(1) **Bag-of-Words (BOW)**. The bag-of-words approach is a simplified representation of text that treats each input text as a multiset of words, disregarding its complex semantics or grammar [26].

(2) **TF-IDF**. TF-IDF improves upon BOW by weighting each term with both term frequency and inverse document frequency, thus being able to identify the key terms that are unique to the given text [1]. TF-IDF can efficiently capture the characteristics of the given text especially if the text is long (*e.g.* multiple paragraphs). Because TF-IDF vector can be very sparse, we reduce the feature dimension to 20,000 by picking only the most frequent term frequencies.

(3) **LDA Topic Distribution**. An LDA model can represent each given text as mixing proportions for latent components, which are often interpreted as "topics" [3, 4]. Thus an LDA model can project text in a much more compact dimension, which has been reported to be effective in many tasks including text categorization [21, 31, 33]. In our experiment, we train a topic model with 50 topics, 2 passes over our corpus with 10K blog documents.

2.4. Image Description

For image description, we use dense feature extraction with vector quantization, which is one of the standard methods in recent computer vision research. We densely extract HSV color SIFT and histogram of oriented edge (HOG) features on a regular grid of each image at steps of 4 and 8 pixels, respectively. Then, we form 300 visual words for each feature type by applying K-means to randomly selected descriptors. Finally, the nearest word is assigned to every node of the grid. As image or region descriptors, we build L_1 normalized spatial pyramid histograms to count the frequency of each visual word in three levels of regular grids. We define the image descriptor $\boldsymbol{v}$ by concatenating the two spatial pyramid histograms of color SIFT and HOG features.

2.5. Text-to-Image Embedding

The *text-to-image embedding* aims to obtain a mapping between images and their associated text, and allows us to retrieve the closest images for a given text, and vice versa. We implement two methods, which include one parametric method using NCCA (Normalized Canonical Correlation Analysis) [9], and one nonparametric method using simple K-nearest neighbor search. Suppose that each of the training blog posts is segmented as a sequence of text and images (*e.g.* $B^n = \{(I_1^n, T_1^n) \cdots, (I_{|B^n|}^n, T_{|B^n|}^n)\}$), and finally we have M image and text pairs. Using the image and text descriptors in previous sections, we represent each image and text as $\mathbf{x}$ and $\mathbf{y}$, respectively. We then repeat the embedding for each image and text descriptor and each segmentation method separately.

(1) **NCCA**. We present a set of M image and text pairs in matrices $\mathbf{X} \in \mathbb{R}^{M \times d}$ and $\mathbf{Y} \in \mathbb{R}^{M \times D}$, respectively. Then the goal of text-to-image embedding is to find matrices $\mathbf{U} \in \mathbb{R}^{d \times c}$ and $\mathbf{V} \in \mathbb{R}^{D \times c}$ to map images and text to a common c-dimensional latent space by $\mathbf{XU}$ and $\mathbf{YV}$. The objective of the CCA is to find $\mathbf{U}$ and $\mathbf{V}$ such that

$$\max_{\mathbf{U},\mathbf{V}} \text{tr}(\mathbf{U}^T\mathbf{X}^T\mathbf{YV}) \quad (1)$$
$$\text{s.t. } \mathbf{U}^T\mathbf{X}^T\mathbf{XU}=\mathbf{I}, \mathbf{V}^T\mathbf{Y}^T\mathbf{YV}=\mathbf{I}.$$

The CCA optimization is solved as a generalized eigenvalue problem as in [9]:

$$\begin{pmatrix} \mathbf{C}_{xx} & \mathbf{C}_{xy} \\ \mathbf{C}_{xy}^T & \mathbf{C}_{yy} \end{pmatrix} \begin{pmatrix} \mathbf{z}_x \\ \mathbf{z}_y \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{C}_{xx} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{yy} \end{pmatrix} \begin{pmatrix} \mathbf{z}_x \\ \mathbf{z}_y \end{pmatrix}, \quad (2)$$

where $\mathbf{C}_{xx} = \mathbf{X}^T\mathbf{X}$, $\mathbf{C}_{xy} = \mathbf{X}^T\mathbf{Y}$, and $\mathbf{C}_{yy} = \mathbf{Y}^T\mathbf{Y}$. We form projection matrices $\mathbf{U}$ and $\mathbf{V}$ from the top c eigenvectors corresponding to each $\mathbf{z}_x$ and $\mathbf{z}_y$, respectively. The NCCA proposes to compute the similarity $\sigma(\mathbf{x}, \mathbf{y})$ between image $\mathbf{x}$ and text $\mathbf{y}$ by

$$\frac{\left(\mathbf{x}\mathbf{U}\text{diag}(\lambda_{x1}^t, \ldots, \lambda_{xc}^t)\right)\left(\mathbf{y}\mathbf{V}\text{diag}(\lambda_{y1}^t, \ldots, \lambda_{yc}^t)\right)^T}{\|\mathbf{x}\mathbf{U}\text{diag}(\lambda_{x1}^t, \ldots, \lambda_{xc}^t)\|_2 \|\mathbf{y}\mathbf{V}\text{diag}(\lambda_{y1}^t, \ldots, \lambda_{yc}^t)\|_2} \quad (3)$$

where $\lambda_{x1}, \ldots, \lambda_{xc}$ are the top c eigenvalues that correspond to eigenvectors $\mathbf{z}_x$, and t is the power to which the eigenvalues are taken. We use $c = 96$ and $t = 4$ as done in [9]. Using the similarity metric of Eq.(3) we can retrieve the closest images for any given text and vice versa.

(2) **KNN**. The KNN is a technique of lazy learning in which we retain all M training pairs of images and text. The similarity $\sigma(\mathbf{x}, \mathbf{y})$ from image $\mathbf{x}$ to text $\mathbf{y}$ is $\sigma(\mathbf{x}, \mathbf{y}) = \cos(\mathbf{x}, \mathbf{x}')$ where $\mathbf{x}' = \text{argmax}_{(\mathbf{x}', \mathbf{y}') \in \mathcal{T}} \cos(\mathbf{y}, \mathbf{y}')$. That is, we first find the closest text $\mathbf{y}'$ to $\mathbf{y}$ from the training set, and compute the cosine similarity between $\mathbf{x}$ and $\mathbf{x}'$ that is associated with $\mathbf{y}'$. Hence, the values of $\sigma(\mathbf{x}, \mathbf{y})$ and $\sigma(\mathbf{y}, \mathbf{x})$ are not identical, because the former is computed from an image space while the latter is from a text space.

In the experiments of Section 4, we compare the retrieval performance of these two embedding methods.

3. The Retrieval Model

We design our ranking and retrieval approach based upon the structural SVM with latent variables (*e.g.* [12, 35]). Our discriminant function is defined as a real-valued function $\mathcal{F}(Q,\mathcal{S}) : \mathcal{Q}\times\mathcal{S} \rightarrow \mathbb{R}^{+}$, which measures the compatibility of a query text $Q \in \mathcal{Q}$ and an image sequence $S \in \mathcal{S}$.

Naturally, the mapping from text to an image sequence depends on how to segment the given passage. We call this task *text segmentation*, which defines the function $\mathcal{G}(Q,\kappa,H)$ that partitions a query paragraph Q into a sequence of κ segments $H = \{h_1,\cdots,h_\kappa\}$, so that each text segment h_i is topically coherent and maps to a single image. Thus the image sequence $S = \{s_1,\cdots,s_\kappa\}$ has the same length with H, where each s_i corresponds to h_i. In practice, each h_i can be single or multiple sentences.

We treat the text segmentation output as a *latent variable*, because its correct answer is not available at both training and test stage. If we denote one instance of paragraph segmentation by $H \in \mathcal{H}$, the retrieval objective is to find the optimal image sequences S^* for a query text Q:

$$S^* = \underset{(S,H)\in\mathcal{S}\times\mathcal{H}}{\operatorname{argmax}}\ \mathcal{F}(Q,S,H) = \underset{(S,H)\in\mathcal{S}\times\mathcal{H}}{\operatorname{argmax}}\ \mathbf{w}\cdot\Psi(Q,S,H) \quad (4)$$

where as usual the discriminant function is linear in the feature vector $\Psi(Q,S,H)$, which describes the relation among query input Q, image sequence output S, and a segment instance H as a latent variable.

3.1. Feature Spaces

We decompose the feature vector into two components:

$$\mathbf{w}\cdot\Psi(Q,S,H) = \boldsymbol{\alpha}\cdot\Phi(Q,S,H) + \boldsymbol{\beta}\cdot\Pi(Q,S,H). \quad (5)$$

The first component $\Phi(Q,S,H)$ includes a set of features describing a one-to-one relation between each pair of (s_i,h_i), whereas $\Pi(Q,S,H)$ consists of feature vectors relating S to H as a set.

The first feature set $\Phi(Q,S,H)$, that measures one-to-one compatibility between text and images, concatenates the mean similarities of all combinations between two image features in Section 2.4 and three text features in Section 2.3. Thus, $\Phi(H,S) \in \mathbb{R}^6$ is defined by

$$\Phi(H,S) = \frac{1}{\kappa}\left[\textstyle\sum_{i=1}^{\kappa}\sigma(\mathbf{x}_i^1,\mathbf{y}_i^1)\cdots\sum_{i=1}^{\kappa}\sigma(\mathbf{x}_i^2,\mathbf{y}_i^3)\right]^T, \quad (6)$$

where $\mathbf{x}_i^1$ and $\mathbf{y}_i^1$ are the first type of image and text descriptors for h_i and s_i, respectively. For image-text similarity $\sigma(\mathbf{x},\mathbf{y})$, we use one of the two methods (*i.e.* NCAA or KNN) defined in Section 2.5.

The second feature set $\Pi(Q,S,H)$ describes compatibility between the sequences of text segments H and images S as a whole. We use the two popular similarity measures: one orderless metric, Hausdorff similarity, and one ordered one, Dynamic Time Warping (DTW) similarity. We consider both orderless and ordered metrics because sequential relation may not be always consistent in different blogs where the image-text ordering often does not match. The final feature dimension becomes $\Pi(Q,S,H) \in \mathbb{R}^{12}$ (*i.e.* 2 metrics $\times$ 3 text $\times$ 2 image features).

3.2. Learning

In order to learn the model (*e.g.* computing the parameter vector $\mathbf{w}$ of Eq.(5)), we use blog data as training data. With a slight abuse of notation, we denote our training data as $\mathcal{B}_t = \{(Q^{(n)},S^{(n)})|n = 1,\cdots,N\}$, where $Q^{(n)}$ and $S^{(n)}$ are the text and image sequence of training blog n. The learning objective of structural latent SVM is

$$\min_{\mathbf{w},\xi\geq 0}\ \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{N}\sum_{n=1}^{N}\xi_n \quad (7)$$

$$\text{s.t.}\ \mathbf{w}\cdot\Big(\Psi(Q^{(n)},S^{(n)}) - \Psi(Q^{(n)},S)\Big) \geq \Delta(S^{(n)},S) - \xi_i, \quad (8)$$
$$\forall n, \forall S \in \mathcal{S}\backslash S^{(n)}$$

where ξ_n is a slack variable and C is a regularization parameter. The loss function is defined as $\Delta(S^{(n)},S) = 1 - \frac{1}{|S^{(n)}|}\sum_{s=1}^{|S^{(n)}|}\sigma(s_i^{(n)},s_i)$, which is the element-wise mean distance between $S^{(n)}$ and S. We let S to have the same length with $S^{(n)}$.

Note that we have no access to the correct segmentation for text $Q^{(n)}$ at both training and test time. Thus, we let the segmentation of each document be a latent variable. More specifically, in Section 2.2, we describes five different segmentation methods that are functions of Q and κ. As a latent variable of each training data, we introduce $\mathbf{h}^{(n)} \in \mathcal{R}^D$, where D is the number of possible segmentation methods. (*e.g.* $D = 5$ in our setting). Thus, $\mathbf{h}^{(n)}$ is a binary vector with only one nonzero element indicating which segmentation output is applied to a training document n. With the latent variables, the constraints of Eq.(8) can be specified as

$$\mathbf{w}_d\cdot\max_{\mathbf{h}_d^{(n)}\in\{1,\cdots,D\}}\Big(\Psi(Q^{(n)},S^{(n)},\mathbf{h}_d^{(n)}) - \Psi(Q^{(n)},S,\mathbf{h}_d^{(n)})\Big)$$
$$\geq \Delta(S^{(n)},S) - \xi_i. \qquad \forall n, \forall S \in \mathcal{S}\backslash S^{(n)}. \quad (9)$$

Now we have D different sets of $\mathbf{w}_d$ that are learned from the blogs that are segmented by method d.

In Eq.(9), $\mathcal{S}$ can be any possible image sequences with the size of κ. Since $S \in \mathcal{S}$ can be countlessly many, we limit the generation of negative S as follows. With a fixed η, for each training blog n, we generate negative S by randomly applying two operations to $S^{(n)}$, which are shuffling

the orders and replacing some of $S^{(n)}$ with other images. We set $\eta = 50$ in the experiments.

Optimization. We use alternating optimization that has been commonly used for the latent structural SVM (*e.g.* [18, 35]). In summary, we alternate between the optimization of SVM parameters $\{\mathbf{w}_d\}_{d=1}^D$, and segmentation labels $\mathbf{h}$. We first randomly initialize $\mathbf{h}^{(n)}$ for each training blog, and apply the segmentation method d accordingly. We then repeat the following two steps until convergence or for a pre-defined number of iterations.

(a) With a fixed segmentation method for each training data n with $\mathbf{h}^{(n)}$, we solve standard structural SVM of Eq.(7) to obtain $\{\mathbf{w}_d\}$. We use the n-slack algorithm with margin-rescaling presented in [12].

(b) With fixed SVM parameters, update the segmentation method by $\mathbf{h}^{(n)} = \text{argmax}_d \, \mathbf{w}_d \cdot \Psi(Q^{(n)}, S^{(n)}, H)$ for all training blogs $n \in \{1, \cdots, N\}$.

3.3. Inference

At test time, we are given a set of learned parameters $\{\mathbf{w}_d\}_{d=1}^D$, a query text Q, training blogs $\mathcal{B}_t$ and the database of photo streams $\mathcal{P}$, from which the best image S^* is selected. The retrieval is performed as follows. Suppose that the candidate image sequences $\mathcal{S}_{cand}$ are given. Then, for each $S \in \mathcal{S}_{cand}$, we compute the score s by $\max_d \mathcal{F}(Q, S, H) = \max_d \mathbf{w}_d \cdot \Psi(Q, S, H)$. That is, once we apply D different segmentation methods to Q, we compute the scores of each image sequence S by finding the maximum score s among D different segmentation outputs with their corresponding $\mathbf{w}_d$. Finally, we can sort $S \in \mathcal{S}$ according to the scores.

However, one major difficulty of this scenario is that there are exponentially many candidates in $\mathcal{S}_{cand}$, so we use an approximate strategy. Once the query text Q is segmented into $H = \{h_1, \cdots, h_\kappa\}$, we first find the K_h-nearest neighbor images of each h_i and constrain $S \in \mathcal{S}_{cand}$ to be generated from them. With this constraint, the size of $\mathcal{S}_{cand}$ becomes $K_h \times \kappa$. Optionally, to further improve the search speed for S^*, we can use a greedy algorithm; for example, once we choose an image for h_1, we select the next image for h_2 that maximizes the sore $\mathbf{w}_d \cdot \Psi(Q, S, H)$. The greedy algorithm has been widely used in the applications of structural SVM to subset selection problems [20, 36].

4. Experiments

We first conduct comprehensive experiments for the image sequence retrieval task to compare the contributions of different text features, segmentation methods, embedding methods, and their combinations. We then demonstrate the ability of our approach to visualize the text-only review entries of TRIPADVISOR and YELP. We perform user studies using Amazon Mechanical Turk to obtain general users' preferences.

4.1. Datasets of Photo Streams, Blogs, and Reviews

We crawl image and text data for the two parks at California: *Disney California Adventure* and *Disneyland Park*.

Photo Stream Data. We collect 542,217 unique images of 6,026 photo streams from Flickr by querying keywords related to *Disneyland*. We only consider photo streams that contain more than 30 images, and remove noisy ones that are not relevant to the park.

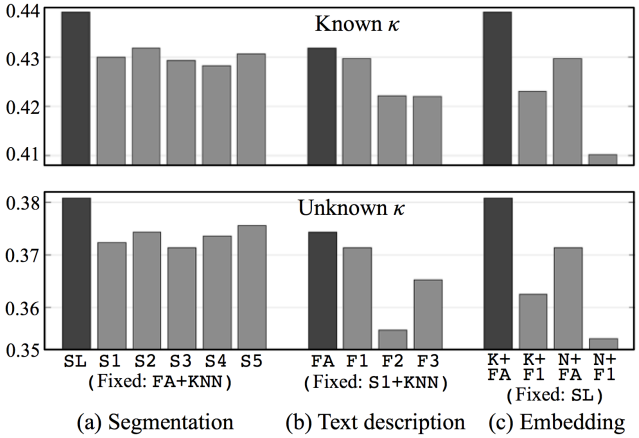
Blog Data. We first crawl 53,091 unique blog posts and 128,563 associated pictures from three popular blog-publishing sites, BLOGSPOT, WORDPRESS, and TYPEPAD by changing query terms from Google search. Then, the blogs are manually classified into three groups by Disneyland experts: *Travelogues*, *Disney* and *Junk*. The *Travelogue* label indicates the blog posts of our interest, which describe stories and events with multiple images in Disneyland. The *Disney* label is applied to the posts that are Disney-related but not travelogues, such as history of Disneyland, Disney films, or merchandise. We use only the blogs of the *Travelogue* group, whose size is 10,075 posts and 121,251 associated images.

TripAdvisor and Yelp dataset. TripAdvisor and Yelp curate traveler reviews for specific venues registered on their sites. We manually pick 100 reviews from each website (200 in total), under venue names *Disney California Adventure* and *Disneyland Park*. We pick reviews that are neither too short nor too long. We use traveler review data for evaluation in order to demonstrate that the knowledge of image-text association obtained from the blog data can be flexibly applied to other types of text entries under the same general domain.

4.2. Results on Image Sequence Retrieval

For quantitative evaluation, we randomly select 80% of blog posts as a training set and the others as a test set. For each test blog post, we use the text portion as a query text Q and the sequence of images as groundtruth S_G. The goal of each algorithm is to retrieve the best image sequence from training blogs or photo streams $\mathcal{P}$. Since the training and test data are disjoint, each algorithm can only retrieve similar (but not identical) images at best. We perform the experiment under two different settings: with or without the groundtruth size (κ) of the image sequence as it appears in the original blog post given to the algorithm. We test with 10 different sets of training and test partitions.

To measure performance, we need to define how close the estimated $S = \{s_1, \cdots, s_\kappa\}$ is to the groundtruth $S_G = \{s_{G1}, \cdots, s_{G|S_G|}\}$ for a query Q. Because the retrieved sequence can only have similar images to the groundtruth and may not have the same number of images, we define similarity-based Jaccard index as an evaluation metric as follows. We first represent each image s by the L_1-normalized descriptor as in Section 2. Since S and S_G are



(a) Segmentation (b) Text description (c) Embedding

Figure 2. Comparison of different (a) segmentation methods, (b) text representation methods, and (c) embedding methods on the image retrieval accuracy with the similarity-based Jaccard Index measure in Eq.(10), when the ground truth image sequence size κ is given (top), and unknown (bottom) to the algorithm.

two vector sequences with possibly different lengths, we align S and S_G using the dynamic time warping (DTW) algorithm, which finds a set of correspondences $\mathcal{C}$ between them. Then, we define the similarity-based Jaccard index as

$$J(S, S_G) = \frac{\sum_{(s_i,s_j)\in\mathcal{C}} \sigma(s_i, s_j)}{\max(|S|, |S_G|)} \qquad (10)$$

where $\sigma(s_i, s_j)$ is the cosine similarity. Note that $J(S, S_G)$ is higher for a more similar pair of S and S_G in both appearance and lengths.

Since the problem of retrieving image sequences from multiple paragraph queries has not been addressed in previous literature, we instead present comprehensive comparison results of different combinations of methods we propose. Specifically, as baselines we vary among five text segmentation methods (Syntactic (S1), LSA (S2), LSA-based-summary (S3), LexRank (S4), and LexRank-based-summary (S5)), three text representation methods (BOW (F1), TF-IDF (F2), and LDA topic distribution (F3)), and two text-to-image embedding methods ((KNN) and (NCCA)). These baselines are tested against our latent structural SVM model that dynamically assigns the best segmentation method per input text (SL) and that jointly optimizes the feature weights (FA).

Comparison of text segmentation methods. We test whether our latent model improves the retrieval performance against the fixed segmentation methods. For this experiment, we fix the embedding to (KNN) and the text description to (FA).

Fig.2.(a) shows that our latent model (SL) can assign the best segmentation method to each input blog post, outperforming all the baselines that use one fixed segmentation method. Since every blog is written in a different style, the results make an intuitive sense that finding and applying the best segmentation algorithm per input blog improves the performance. Obviously, algorithms perform better when the groundtruth image sequence size (κ) is given, retrieving image sequences that deviate less from the groundtruth images. Note that even a small numerical increase in accuracy with the Jaccard Index measure indicates a significant qualitative improvement, as evident in Section 4.3.

vs. (FA+NCCA)	# Votes	5	4	3	2	1	0
60.6% (303/500)	# Samples	15	26	21	26	9	3
vs. (F1+KNN)	# Votes	5	4	3	2	1	0
59.0% (295/500)	# Samples	16	19	29	19	14	3
vs. (S2+KNN)	# Votes	5	4	3	2	1	0
57.4% (287/500)	# Samples	11	21	32	18	16	2

Table 1. The results of pairwise preference tests on visualization of consumer reviews via AMT between our method and three baselines. The numbers indicates the percentage of responses that our visualization is better than that of baseline for a given sentence.

Comparison of text features. We test whether our model improves the retrieval performance compared to the baselines where one text descriptor is fixed. For this experiment, we fix the segmentation and embedding method to (S1)+(KNN). Fig.2.(b) shows that the aggregated feature set (FA) outperforms the single best-performing feature set. Among the text features, the TF-IDF (F2) performs the best. The results clearly show that textual information gained from different representations is complementary, and our structural SVM model successfully adjusts the importance of each pair of image and text features (FA).

Comparison of text-to-image embedding methods. Fig.2.(c) compares the performance of the two embedding methods when applied with two different feature descriptors (FA) and (F2). We fix the segmentation to (SL). In our experiments, the nonparametric method (KNN) learns better mapping between images and text than the parametric (NCCA). We observe that the best pair of an embedding method and a text representation is (KNN)+(FA).

In conclusion, our approach consistently outperforms baselines by 2.5% to 8%. The accuracy improvement quantitatively appear moderate, mainly because the similarity-based metric of Eq.(10) encodes correct trends but suppresses perceptual differences (similar to the BLEU score).

Qualitative results. Fig.3 shows examples of image sequence retrieval. We compare the results of our algorithm and baselines with the groundtruth. Our algorithm illustrates the main theme of documents by retrieving relevant images in terms of attractions (*e.g.* Mickey's toontown), events (*e.g.* parades), and locations (*e.g.* restaurants).

4.3. Visualization of Customers' Reviews

We evaluate the ability of our algorithm to visualize the general users' reviews. We build a set of query text $\mathcal{Q}$ by selecting 100 story-like reviews that mainly describe

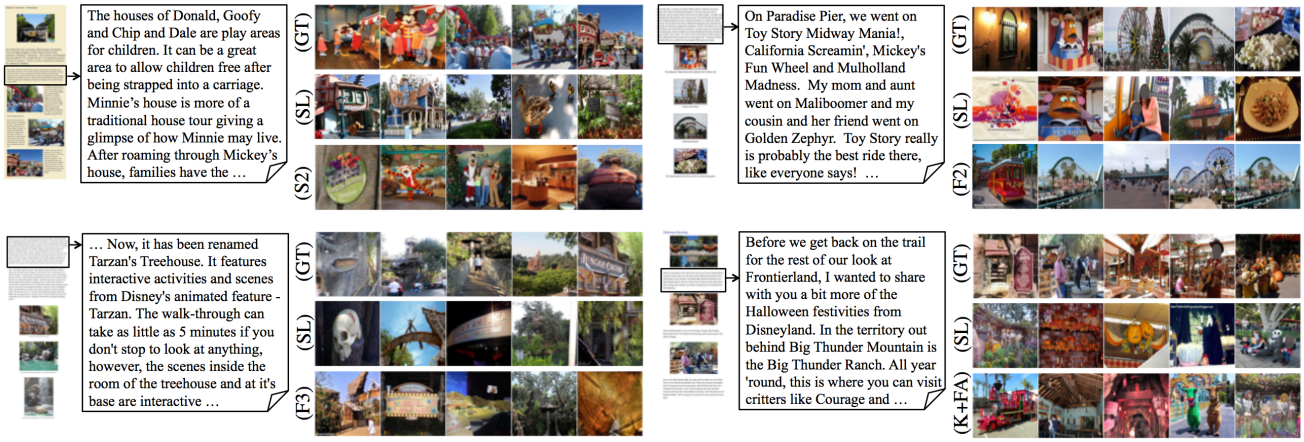


Figure 3. Qualitative comparison of the image sequence retrieval. On the left, we show downsized original blog posts and parts of text queries. On the right, we show the groundtruth image sequences (GT), the prediction of our algorithm (SL), and that of one baseline from top to bottom. More similar the predicted sequences are to groundtruth, more accurate the retrieval algorithm is for given text queries. Note that actual blogs and image sequences are much longer; we show only parts of them for illustration purpose.

Figure 4. Examples of visualization of consumers' reviews on TripAdvisor and Yelp. We show relevant parts of the documents with bold fonts for the words that are visualized in the retrieved images. For each text query, we show three top-ranked image sequences. Note that actual query reviews and image sequences are much longer; we show only parts of them for illustration purpose.

the flow of the trip sequentially, from each of TripAdvisor and Yelp datasets. Since the query reviews are text-only and have no groundtruth image sequences, we employ crowdsourcing-based evaluation via Amazon Mechanical Turk (AMT). We first train our method and baselines using the blog data. We then show each query review and a pair of image sequences predicted by our algorithm and one of the baselines in a random order, and ask a turker to choose the one that visualizes the passage best. We obtain these preference answers from five different turkers per query.

Table 1 shows the results of pairwise AMT preference tests. We compare with three baselines, each of which uses a different text feature, segmentation, and embedding method. Although the question is highly subjective in nature and has a variety of equally good answers, our results are favored by the general turkers by larger margins.

Fig.4 shows three top-ranked image sequences for TripAdvisor and Yelp reviews. Although actual query reviews and image sequences are much longer, we select some parts of them for illustration purpose. We highlight the terms that are visualized by our algorithm. Disneyland provides a diverse sets of entertainment, activities, events, and attractions, which are highly co-located in both text and images. Our approach helps build cross-reference between them, which can enable a wide range of promising and engaging Web applications.

5. Conclusion

We proposed a method to retrieve image sequences for a text query of multiple paragraphs. Using the blogs and photo streams on the Web, we built an image-text parallel corpus where the association was learned. We formulated a latent structural SVM to learn their semantic relations, and presented a comprehensive evaluation against a number of baselines as well as a user study via AMT.

References

[1] A. Aizawa. An Information-Theoretic Perspective of TF-IDF Measures. *Info. Proc. Manag.*, 39(1):45–65, 2003. 4

[2] D. Beeferman, A. Berger, and J. Lafferty. Statistical Models for Text Segmentation. *Mach. Learn.*, 34(1-3):177–210, 1999. 3

[3] D. M. Blei and J. D. Lafferty. Dynamic Topic Models. In *ICML*, pages 113–120, 2006. 4

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *JMLR*, 3:993–1022, 2003. 4

[5] F. Y. Y. Choi, P. Wiemer-Hastings, and J. Moore. Latent Semantic Analysis for Text Segmentation. In *EMNLP*, 2001. 3

[6] G. Erkan and D. R. Radev. LexRank: Graph-based Lexical Centrality As Salience in Text Summarization. *JAIR*, 22(1):457–479, 2004. 3, 4

[7] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every Picture Tells a Story: Generating Sentences from Images. In *ECCV*, 2010. 1, 2

[8] Y. Feng and M. Lapata. Automatic Caption Generation for News Images. *IEEE PAMI*, 35(4):797–812, 2013. 1, 2

[9] Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik. Improving Image-Sentence Embeddings Using Large Weakly Annotated Photo Collections. In *ECCV*, 2014. 1, 2, 4

[10] M. Hodosh, P. Young, and J. Hockenmaier. Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics. *JAIR*, 47:853–899, 2013. 1, 2

[11] T. Joachims. Training Linear SVMs in Linear Time. In *KDD*, 2006. 2

[12] T. Joachims, T. Finley, and C. N. J. Yu. Cutting-plane Training of Structural SVMs. *Mach. Learn.*, 77:27–59, 2009. 5, 6

[13] D. Joshi, J. Z. Wang, and J. Li. The Story Picturing Engine: A System for Automatic Text Illustration. *ACM TOMM*, 2(1):68–89, 2006. 2

[14] A. Karpathy, A. Joulin, and L. Fei-Fei. Deep Fragment Embeddings for Bidirectional Image Sentence Mapping. In *NIPS*, 2014. 2

[15] K. Kireyev. Using Latent Semantic Analysis for Extractive Summarization. In *TAC*, 2008. 4

[16] R. Kiros, R. Salakhutdinov, and R. Zemel. Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models. In *TACL*, 2015. 2

[17] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby Talk: Understanding and Generating Image Descriptions. In *CVPR*, 2011. 1, 2

[18] T. Lan, W. Yang, Y. Wang, and G. Mori. Image Retrieval with Structured Object Queries Using Latent Ranking SVM. In *ECCV*, 2012. 2, 6

[19] T. A. Letsche and M. W. Berry. Large-scale Information Retrieval with Latent Semantic Indexing. *Information Sciences*, 100(1–4):105–137, 1997. 3

[20] L. Li, K. Zhou, G.-R. Xue, H. Zha, and Y. Yu. Enhancing Diversity, Coverage and Balance for Summarization through Structure Learning. In *WWW*, 2009. 6

[21] C. Lin and Y. He. Joint Sentiment/Topic Model for Sentiment Analysis. In *CIKM*, pages 375–384, 2009. 4

[22] D. Lin, S. Fidler, C. Kong, and R. Urtasun. Visual Semantic Search: Retrieving Videos via Complex Textual Queries. In *CVPR*, 2014. 2

[23] E. Loper and S. Bird. NLTK: The Natural Language Toolkit. In *ETMTNLP*, 2002. 3

[24] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2Text: Describing Images Using 1 Million Captioned Photographs. In *NIPS*, 2011. 1, 2

[25] A. Sadeghi and A. Farhadi. Recognition Using Visual Phrases. In *CVPR*, 2011. 2

[26] G. Salton, E. A. Fox, and H. Wu. Extended Boolean Information Retrieval. *CACM*, 26(11):1022–1036, 1983. 4

[27] B. Siddiquie, R. S. Feris, and L. S. Davis. Image Ranking and Retrieval based on Multi-Attribute Queries. In *CVPR*, 2011. 2

[28] B. Siddiquie, B. White, A. Sharma, and L. S. Davis. Multi-Modal Image Retrieval for Complex Queries using Small Codes. In *ICMR*, 2014. 2

[29] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. Grounded Compositional Semantics for Finding and Describing Images with Sentences. In *TACL*, 2013. 1, 2

[30] J. Steinberger and K. Jezek. Using Latent Semantic Analysis in Text Summarization and Summary Evaluation. In *ISIM*, 2004. 3

[31] S. Tasci and T. Gungor. LDA-based Keyword Selection in Text Categorization. In *ISCIS*, 2009. 4

[32] Y. Wang and J. Ma. A Comprehensive Method for Text Summarization Based on Latent Semantic Analysis. *NLPCC*, 400:394–401, 2013. 3

[33] Z. Wang and X. Qian. Text Categorization Based on LDA and SVM. In *CSSE*, pages 674–677, 2008. 4

[34] B. Z. Yao, X. Yang, L. Lin, M. W. Lee, and S.-C. Zhu. I2T: Image Parsing to Text Description. *IEEE Proc*, 98(8):1485–1508, 2010. 2

[35] C.-N. J. Yu and T. Joachims. Learning Structural SVMs with Latent Variables. In *ICML*, 2009. 2, 5, 6

[36] Y. Yue and T. Joachims. Predicting Diverse Subsets Using Structural SVMs. In *ICML*, 2008. 6

[37] C. L. Zitnick, D. Parikh, and L. Vanderwende. Learning the Visual Interpretation of Sentences. In *ICCV*, 2013. 1, 2