Text-to-Image Generation based on Crossmodal Association with Hierarchical Hypergraphs

Jung-Woo Ha
School of Computer Sci. and Eng.
Seoul National University
Gwanak-ro 1, Gwanak-gu, Seoul, Korea
jwha@bi.snu.ac.kr

Byoung-Tak Zhang
School of Computer Sci. and Eng.
Seoul National University
Gwanak-ro 1, Gwanak-gu, Seoul, Korea
btzhang@bi.snu.ac.kr

Abstract

In this paper, we propose a novel framework for text-to-image generation based on association between text and image modalities. As an association model, we use hierarchical hypergraphs which consist of two layers including heterogeneous hypergraphs. While the first layer is composed of two hypergraphs: a text hypergraph and an image hypergraph, a hypergraph in the second layer associates two modalities by merging two hypergraphs in the first layer. In our model, hypergraphs are learned by self-organizing method based on random sampling. With multimodal association represented in the learned model, an intermediate image is generated by cross-modal inference when text keywords are given as a query. We use Korean magazine articles as a text-image data for experiments and we illustrate generated intermediate images and retrieved images similar to the intermediates as experimental results.

1 Introduction

Text-to-image generation [1] is a challenging issue in the field of vision and language integration due to semantic gap and the difference of granularity between two modalities. When compared with image-to-text generation such as image annotation [2], text-to-image generation has been considered as a more difficult task because it includes synthesis of image information. In this paper, we propose a novel framework for text-to-image generation based on cross-modal association [3] and use a hierarchical hypergraph model for associating text and image modalities.

Our model has a hierarchical structure with two layers consisting of hypergraphs which represent heterogeneous information. While two hypergraphs for each text and image modality exist in the first layer, the hypergraph in the second layer represents inter-modal relationships by merging hyperedges of each modality-hypergraph in the first layer. Goal of learning is to build these hypergraphs both to abstract features from raw variables of text and image and to maximize the probability of generating stored patterns by reflecting the association from patterns in the given multi-modal data. Learning in our model consists of two phases: the first layer learning and the second one. In the first phase, hypergraphs for each modality are learned by generating hyperedges, calculating weight of hyperedges, and removing hyperedges with low-weight. In the second phase of learning, multimodal

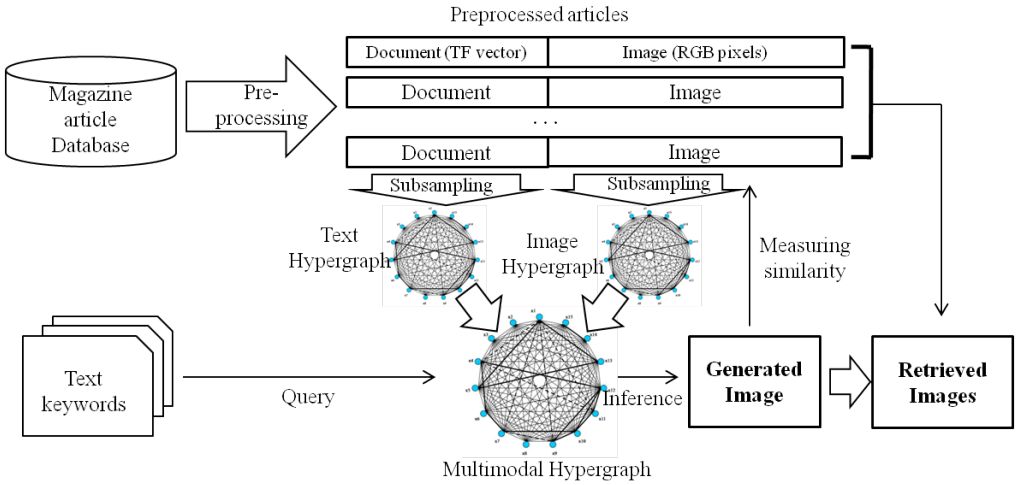

Figure 1. Framework of text-to-image generation and image retrieval with hierarchical hypergraphs. Hierarchical hypergraphs are trained with magazine article data and an image is generated by inference with the learned model when text keywords are given as a query. Finally, original images are retrieved from article database based on similarity between the generated image and original images.

hypergraph is generated by merging two modality-dependant hypergraphs based on meta-information.

Also, we generate intermediate images used to retrieve images with the learned model. When text keywords are given as a query, an intermediate image is generated with abstracted image features related to the query. By measuring the difference of generated images and original images, we apply our framework to retrieve images.

For experiments, we use Korean magazine articles consisting of pairs of a document and an image. Magazine articles are suitable data for associating text and image information because they consist of a document and high-quality images related to the topic of the document. We use whether two hyperedges are originated from same article as a meta-information for building the multimodal hypergraph. A document and an image are converted to a term frequency vector of predefined vocabularies with 4,062 keywords and a vector of 60 by 40 pixels with 24bit RGB scale. Experimental results show that images related to given text query are generated by our method and it can be applied to image retrieval. Figure 1 illustrates overall flow of the proposed framework for text-to-image generation.

2 Hierarchical hypergraphs

2.1 Learning hierarchical hypergraphs

A hypergraph is an extended graph with generalized representation power [4]-[5]. A hypergraph consists of vertices and hyperedges, edges connecting to more than two vertices at the same time. Therefore, hypergraphs have been applied to dealing with higher-order relationships among heterogeneous features in various domains such as bioinformatics and multimedia mining [6]-[9]. This property allows hypergraphs to represent relationships among text and image features effectively. Most previous works using hypergraphs focus on analyzing relationships among vertices for clustering or categorizing and use prior knowledge such as defined relations [7]-[8], measured similarity between objects [9], and gene regulatory networks [10] for building a hypergraph.

In this study, we propose a novel data-driven method for building and learning hypergraphs. Vertices and hyperedges in our model denote data variables and arbitrary combinations of variables, respectively. With this definition, our model can represent higher-order relationships between variables. Also, we formulate a hypergraph to a probabilistic graphical model and introduce a hierarchical structure with two layers into our model for efficient

learning. The first layer includes two hypergraphs with higher-order features of text and image and a hypergraph in the second layer consists of hyperedges which are intermodal relationships between abstract information of text and image by combining hyperedges from each modality hypergraph based on given data patterns.

Formally, a hypergraph H is defined to $H = (V, E)$ where V and E denote a vertex set and a hyperedge set. Also, $w(e)$ denotes weight of a hyperedge e. Let it call "e matches $\mathbf{x}$" that all vertices of e are equal to the values of corresponding variable in $\mathbf{x}$. When the information of the n-th stored pattern $\mathbf{x}^{(n)}$ in the i-th hyperedge e_i is represented with energy function, $\varepsilon(e_i;\mathbf{x}^{(n)})$ is defined as follow:

$$\varepsilon(e_i;\mathbf{x}^{(n)}) = w(e_i)I(\mathbf{x}^{(n)}, e_i) \tag{1}$$

where $I(\mathbf{x}^{(n)}, e_i)$ denotes an indicator function which yields 1 if e_i matches $\mathbf{x}^{(n)}$ and 0, otherwise. Then, the probability of generating data $D = \left\{\mathbf{x}^{(n)}\right\}_{n=1}^{N}$ with a hypergraph H, $P(D|H)$, is given as a Gibbs distribution:

$$\begin{aligned} P(D \mid H) &= \prod_{n=1}^{N} P(\mathbf{x}^{(n)} \mid H) = \prod_{n=1}^{N} \frac{1}{Z(H)} \exp\left(-\varepsilon(E;\mathbf{x}^{(n)})\right) \\ &= \prod_{n=1}^{N} \frac{1}{Z(H)} \exp\left(-\sum_{i=1}^{|E|} w(e_i) I(\mathbf{x}^{(n)}, e_i)\right) \end{aligned} \tag{2}$$

where Z(H) is a partition function.

Leaning hypergraphs is finding H of maximizing the above probability and is maximizing log-likelihood function:

$$\prod_{n=1}^{N} P(\mathbf{x}^{(n)} \mid H) = \left(\frac{1}{Z(H)}\right)^{N} \exp\left\{-\sum_{n=1}^{N} \varepsilon(E;\mathbf{x}^{(n)})\right\}, \tag{3}$$

$$\arg\max_{H}\left[\log\left\{\prod_{n=1}^{N} P(\mathbf{x}^{(n)} \mid H)\right\}\right] = \arg\max_{H}\left\{\sum_{n=1}^{N}\sum_{i=1}^{|E|} w(e_i) I(\mathbf{x}^{(n)}, e_i) - N \log Z(H)\right\}. \tag{4}$$

To maximize the probability, therefore, learning the model includes find optimal set of hyperedges with high weight value. By learning, an optimal multimodal hypergraph $\hat{H}$ is generated by merging optimal text hypergraph $\hat{H}^T$ and image hypergraph $\hat{H}^I$ based on meta-information of relationships documents and images:

$$\hat{H} = \arg\max_{H} P(\mathbf{x} \mid H) = \prod_{k=1}^{K} \frac{1}{Z(H)} \exp\left(-\sum_{i=1}^{|E|} w(e_i) I(\mathbf{x}^{(k)}, e_i)\right), \tag{5}$$

$$H = H^T \oplus H^I \quad , \tag{6}$$

$$\hat{H}^T = \arg\max_{H^T} P(T \mid H^T) = \prod_{n=1}^{N} \frac{1}{Z(H^T)} \exp\left(-\sum_{i=1}^{|E^T|} w(e_i^T) I(\mathbf{x}^{T(n)}, e_i^T)\right), \tag{7}$$

$$\hat{H}^I = \arg\max_{H^I} P(I \mid H^I) = \prod_{m=1}^{M} \frac{1}{Z(H^I)} \exp\left(-\sum_{i=1}^{|E^I|} w(e_i^I) I(\mathbf{x}^{I(m)}, e_i^I)\right), \tag{8}$$

where $\oplus$ denotes an operator of merging two hypergraphs and it will be explained later.

Learning hierarchical hypergraphs has two phases such as the first layer learning and the second layer learning and these phases repeat every epoch. The first layer learning consists of generating hyperedges, calculating weight of hyperedges, and eliminating low-weighted hyperedges for each modality. Figure 2 explains algorithm for learning a hierarchical hypergraph model.

learn_hierarchical_hg(D)

D: a data set, $D = \{T, I\}$ where T and I are a text set and an image set.
$|E|$: the size of a hyperedge set E
H^T: text hypergraph
H^I: image hypergraph
H: multimodal hypergraph
K: the maximum number of epoch
for $i \leftarrow 1$ to $|E^T|$
  $E^T \leftarrow$ generate_hyperedge(E^T, T)
end for
for $i \leftarrow 1$ to $|E^I|$
  $E^I \leftarrow$ generate_hyperedge(E^I, I)
end for
for k$\leftarrow 1$ to K
  $H^T \leftarrow$ calculate_weight(E^T, T)
  $H^I \leftarrow$ calculate_weight(E^I, I)
  $E \leftarrow$ merge_hypergraphs(E^T, E^I)
  $H \leftarrow$ calculate_weight(E, D)
  $H^T \leftarrow$ replace_low_hyperedges(E^T)
  $H^T \leftarrow$ replace_low_hyperedges(E^T)
end for

Figure 2. Algorithm of learning hierarchical hypergraphs. generate_hyperedge(.) is explained in Figure 3. calculate_weight(.) and replace_low_hyperedges(.) is implemented by (9), (12), and (13)

A hyperedge is generated based on random sampling from given data and Figure 3 shows algorithm of generating hyperedges. This sampling method allows the model to approximate the distribution of the data. When generating a hyperedge from image data, in this study, eight neighboring pixels of randomly selected pixel are sampled together to be a patch and the patch becomes a vertex in a hyperedge like Figure 4. Therefore, an image hyperedge includes several image patches and degree of image hyperedges is defined to the number of patches dissimilar to the one of text hyperedges.

Weight of text hyperedge is a function of the number of matching data samples and the occurrence frequency in data of each vocabulary:

$$w(e^T) = \frac{\sum_{n=1}^{N} I(x^{T(n)}, e^T)}{\sum_{k=1}^{\delta(e)} f(v_k, \mathbf{x}^T) + \alpha \sum_{k=1}^{\delta(e)} g(v_k, E) + \varepsilon}, \tag{9}$$

where v_k is the k-th vertex of e, ε denotes a small constant for preventing divide by zero, and f and g are functions returning the number of data instances whose v_k is positive value and the number

generate_hyperedge(E, D)

e_i: the i-th hyperedge
$\delta(e_i)$: degree of e_i (the number of vertices in e_i)

$\boldsymbol{x} \leftarrow \mathbf{x}^{(n)}$: the n-th instance of a data set D
$e_i \leftarrow \{\}$;
for $k \leftarrow 1$ to $\delta(e_i)$
  $idx \leftarrow$ random_sampling($|\boldsymbol{x}|$);
  $e_i \leftarrow e_i \cup \{\boldsymbol{x}_{\text{idx}}\}$; the value of the idx-th variable of x
end for
$E \leftarrow E \cup \{e_i\}$;

Figure 3. Algorithm of generating hyperedges. random_sampling() is implemented variously with considering the property of data.

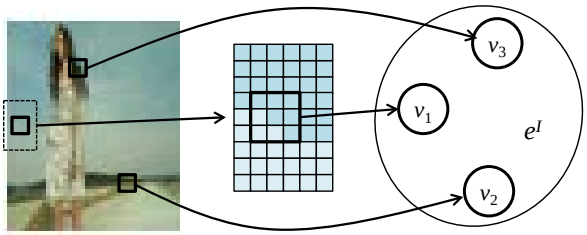

Figure 4. Flow of learning hierarchical hypergraphs

of hyperedges including v_k regardless of its value, respectively. These two terms are used for preventing from sampling a few vocabularies with high frequency only. To define weight of image hyperedges, we modify indicator function to $I'(\mathbf{x}, e)$ as follows:

$$I'(\mathbf{x}^I,\ e^I) = \begin{cases} 1 & (\Delta < \theta) \\ 0 & (\Delta \geq \theta) \end{cases}, \tag{10}$$

$$\Delta = \beta \left\| p^{\mathbf{x}} - p^e \right\|^2 + (1-\beta) \left\| \delta(p^{\mathbf{x}}) - \delta(p^e) \right\|^2, \tag{11}$$

where p^{x} and p^e denote pixel values of data and hyperedge, respectively and $\delta(p)$ is a value matrix of difference between neighboring two pixels comprising e^I. By considering both RGB pixel values and difference values between pixels, an image hyperedge can reflect color patterns better. Then, weight of image hyperedge can be formulated with two functions I' and g like (9):

$$w(e^I) = \frac{\sum_{n=1}^{N} I'(x^{I(n)}, e^I)}{\sum_{k=1}^{\delta(e)} g(v_k, E) + \varepsilon}. \tag{12}$$

The number of removed hyperedges with low weight is determined by epoch number:

$$R_t = \frac{R_{max} - R_{min}}{\exp(t/\kappa)} + R_{min}, \tag{13}$$

where R_{max} and R_{min} denote maximum and minimum boundary value of R_t, respectively, κ is a constant for controlling the speed from R_{max} to R_{min}. By repeating this replacement of hyperedges, a hypergraph is self-organzied

The second learning phase is merging two modality hypergraphs by combining two hyperedges from each modality hypergraph. Two hyperedges are merged based on meta-information document and images and we use whether the document and the image where hyperedges are originated are from a same article as meta-information. Repeating these two phase of learning, we can find the optimal model to associate text and image. Figure 5 shows overall flow of learning hierarchical hypergraphs with magazine article data.

2.2 Text-to-image generation

Text-to-image generation is to generate an intermediate image with image patches consisting of RGB pixels in hyperedges including vocabularies given as a text query. Text-to-image generation consists of three processes: text query expansion, raw image generation, and similar original image retrieval. The text query is expanded by making a set of vocabularies in hyperedges including given text words as vertices. An intermediate image is 60 by 40 pixels with 24bit RGB scale and the value of each pixel $p_{i,j}$ is calculated by weighted summation of pixels in hyperedges including given query and expanded query:

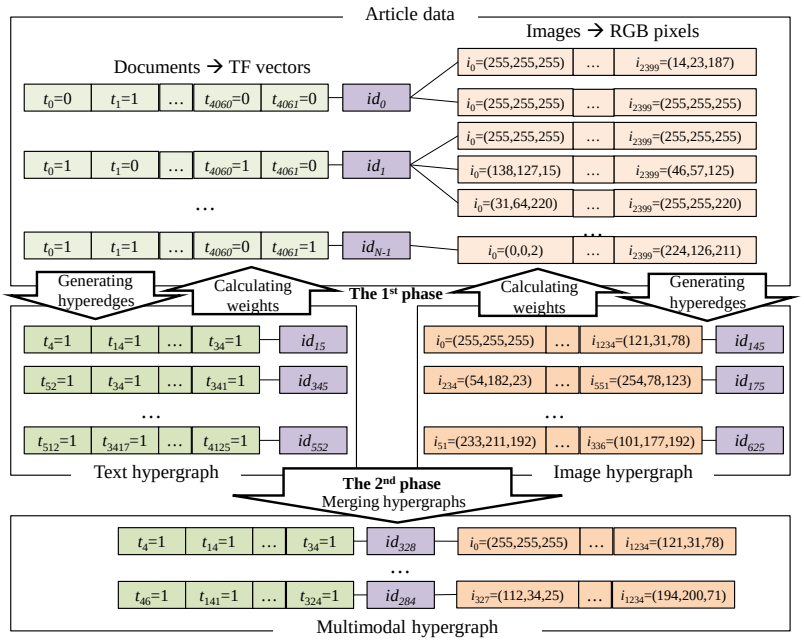

Figure 5. Flow of learning hierarchical hypergraphs

$$p_{i,j} = \frac{1}{\sum_{e \in E^Q, v \in e} \{q \cdot w(e)\}} \cdot \sum_{e \in E^Q, v \in e} \left\{ q \cdot w(e) \sum_{v \in e} p_v^{(q)} \right\}, \tag{14}$$

$$p_{i,j}^{(q)} = \frac{\alpha}{\sum_{e \in E^{Q(q)}} \{w(e)\}} \cdot \sum_{e \in E^{Q(q)}} \left\{ w(e) \sum_{v \in e} p_v \right\} + \frac{1-\alpha}{\sum_{e \in E^{E(q)}} \{w(e)\}} \cdot \sum_{e \in E^{E(q)}} \left\{ w(e) \sum_{v \in e} p_v \right\} \text{ s.t. } q = h(Q,e) \quad , \tag{15}$$

$$p_v = \begin{cases} v & (Idx(v) = 40 \times i + j) \\ 0 & (\text{otherwise}) \end{cases}, \tag{16}$$

Where $h(Q,v)$ is the number of text word elements of query Q in e, E^Q and E^E denote a set of given query vocabulary and expanded words, respectively, $Idx(v)$ is the index in the image pixel vector of vertex v, and α is a constant in (0, 1) for weighting image patches directly related to given query and we set α to 0.1. Finally, original images are retrieved by measuring the distance $Diff(I, I')$ to the intermediate image I':

$$Diff(I, I') = \sum_{e \in E^Q, v \in e} \left\{ q^{\gamma} \cdot \left\| I - I^{(q)} \right\|^2 \right\} \quad , \tag{17}$$

$$\left\| I - I^{(q)} \right\|^2 = \sum_{i=1}^{60} \sum_{j=1}^{40} \left\{ (R_{i,j} - R_{i,j}^{(q)})^2 + (G_{i,j} - G_{i,j}^{(q)})^2 + (B_{i,j} - B_{i,j}^{(q)})^2 \right\}, \tag{18}$$

where R, G, and B are 8bit values of red, green, blue pixels. Introducing q into calculating the distance allows hyperedges with more text keywords of the given query to have stronger influences on measuring the distance.

3 Experimental results

We use 2,477 Korean magazine articles and each article consists of a document and an image. As preprocessing, we define 4,062 words frequently occurred in data to a vocabulary set and

convert a document to a binary value vector of the defined vocabulary set. If a word appears at least one time in a document, the value is 1. Moreover, images are resized 60 by 40 pixels with 24bit RGB scale. As a parameter setting, we set up degree of hyperedges of text hypergraphs and image hypergraphs to 10 and 20, respectively. Also, each of text and image hypergraph has 12,385 hyperedges and a text hyperedge is merged with five image hyperedge originated from the same article.

Figure 6 depicts some text-to-image generation results for five text queries. According to Figure 6, it is not easy to understand contents from intermediate images but we can verify that some retrieved images are semantically related to the given query in first two keywords. The third and fourth keywords generate images a little associated to them while we cannot associate retrieved images with be the fifth query 'red'. This wrong association occurs because 'red' appears in too many documents and the generated intermediate images are averaged by many patches. Furthermore, retrieved images are similar to generated images in aspect to color due to the definition of the similarity. We can also find the values of difference value are smaller when the given keywords are more specific.

Text query	Given only	Expan-ded	Retrieved images				
			1	2	3	4	5
wagon							
			93.46	100.57	114.86	117.73	118.79
Financial crisis							
			81.26	103.57	115.22	118.41	122.1
desert							
			126.45	127.07	130.76	133.00	133.54
shoes							
			137.34	137.68	138.84	140.45	141.11
red							
			143.75	146.24	147.81	148.44	149.64

Figure 6. Result of text-to-image generation. 'given only' and 'expanded' denote generated intermediate images using given text query only and expanded query as well as given query, respectively. Numerical value under retrieved images is the measured difference.

Acknowledgments

This work was supported by the National Research Foundation (NRF) grants (2011-0016483-Videome, 2010-0018950-BrainNet), the IT R&D Program of KEIT (10035348-mLife), and the BK21-IT Program.

References

[1] L. Fraczak, (1995) "From route descriptions to sketches: a model for a text-to-image translator," *ACL 1995*.

[2] A. Hanbury (2008) "A survey of methods for image annotation," *Journal of Visual Languages & Computing* 19(5):617-627.

[3] T. Jiang et al. (2006), "Discovering Image-Text Associations for Cross-Media Web Information Fusion," *Lecture Notes in Computer Science* (*PKDD 2006*) 2006 4213:561-568.

[4] Zhou, D. et al. (2007) "Learning with hypergraphs: Clustering, classification, and embedding, *Advances in Neural Information Processing Systems (NIPS)* 19.

[5] B. -T. Zhang, (2008) "Hypernetworks: A molecular evolutionary architecture for cognitive learning and memory," *IEEE Computational Intelligence Magazine* **3**(3):49-63.

[6] S. Klamt et al. (2009) "Hypergraphs and Cellular Networks," *PLoS Comput Biol* **5**(5): e1000385.

[7] Q. Liu et al. (2011) "Hypergraph with sampling for image retrieval," *Pattern Recognition* **44** (2011):2255-2261.

[8] S. Tan et al. (2011) "Using Rich Social Media Information for Music Recommendation via Hypergraph Model," *Social Media Modeling and Computing Part 3*:213-237.

[9] Y. Huang et al. (2011) "Unsupervised Image Categorization by Hypergraph Partition,*" IEEE Trans on Pattern Analysis and Machine Intelligence* **33**(6):1266-1273.

[10] Z. Tian et al. (2009) "A hypergraph-based learning algorithm for classifying gene expression and arrayCGH data with prior knowledge," *Bioinformatics* (2009) **25** (21): 2831-2838.