

Blog Article Summarization with Image-Text Alignment Techniques

Wei-Ta Chu

National Chung Cheng University, Taiwan

Email: wtchu@ccu.edu.tw

Ming-Chih Kao

National Chung Cheng University, Taiwan

Email: a49906207a@gmail.com

Abstract—We propose an image-text alignment framework to match images with text, and take blog article summarization as the main application. Objects in an image are first detected, from them deep features are extracted and transformed into a space commonly shared with the text. On the other hand, sentences of a blog article are represented as vectors, and are also embedded into the common space. With these processes, cross-modal matching can be achieved. A blog article is then summarized in the representation of images and their matched sentences. In evaluation, we demonstrate the effectiveness of the proposed method, and show that the generated summary makes more sense.

I. INTRODUCTION

Writing blog articles to express life styles gets more and more popular in recent years. According to Wikipedia, in 2011, there were over 156 million public blogs on the internet. In 2014, there were around 172 million Tumblr and 75.8 million WordPress blogs worldwide. Enabling blog article writing has been important social networking service. Recently, there may be many pictures embedded in a long blog article. Content of blog articles becomes richer; however, lengthy articles impede efficient access and often prevent users from reading articles on mobile devices. People may need to scroll again and again to read articles, which is quite annoying. Therefore, we want to take blog article summarization as the main target.

Traditional document summarization methods mostly rely on natural language processing. Although the role of images is undoubted, visual information was rarely considered in document summarization before. Therefore, motivated by [1], we propose to summarize blog articles by associating each image in the article with a sentence with closest semantics. A blog summary is then constructed by a set of image-sentence pairs, as shown in Fig. 1, which can be efficiently presented on mobile devices with limited display size.

Obviously, the main challenge is to associate image and text in a systematic and semantic way. We develop a deep neural network to embed images and text into a common space, so that different modalities are comparable. Ideally, after embedding, entities with similar semantics would be mapped to close positions. Currently, the techniques of image captioning attract much attention, and many researches have proposed exciting results. However, our goal is to summarize the original article in the representation of the author's words. Fig. 2 shows the comparison between the sentence selected by the proposed framework, and the sentence generated by an




Article title : 5 great reasons to include Tarangire National Park on a Tanzania safari	
	1) Unlike Lake Manyara, you are far less-likely to encounter a long wait at the registration desk or to battle crowds of vehicles fighting for position to give their passengers the best vantage point for gawking at one leopard in a tree.
	2) There are heavily wooded areas spotted with the trademark baobab trees, as well as an abundance of abandoned termite mounds.
	3) Additionally, the park is home to resident wildebeest, zebra, giraffes, leopards, lions, buffalo, warthogs and delightful Vervet monkeys.

Fig. 1. One sample summarization result.


	our framework	Sit adjacent to rice paddies or dine by the river; four of Chiang Mai's most beautiful hotels are waiting to enchant your tastebuds.
	image caption	A close up of a picnic table with a cake.

Fig. 2. Comparison of the sentence selected by our framework (top), and the sentence generated by an image captioning engine [2].

image captioning engine [2]. We see that selecting sentences by the proposed approach more matches with the context of the original article.

The rest of this paper is organized as follows. We give brief literature survey in Section II. In Section III, we describe how to construct the proposed image-text alignment framework based on deep neural networks. Experimental results are presented in Section IV, followed by conclusion in Section V.

II. RELATED WORKS

In [3], a pure text-based method was proposed to summarize documents for mobile devices. Similarity between sentences is first computed, and then sentences are clustered. Important sentences from each cluster are selected to form document summaries. Visual information was not used in this work. Feng et al. [1] proposed an approach to automatically generate captions for images in news documents. They adopted the Latent Dirichlet Allocation (LDA) model to discover latent topics in images and text of news documents. Based on latent

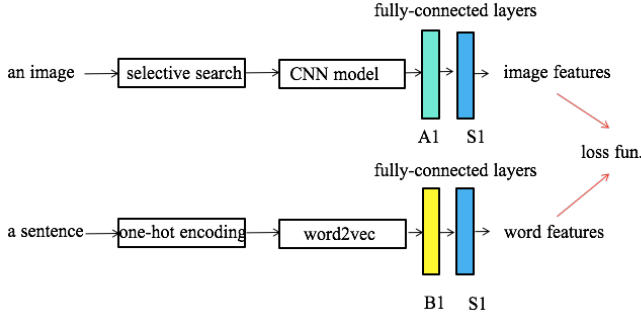


Fig. 3. The proposed image-text alignment framework.

topic distributions, they calculated the KL divergence between an image and each sentence. In [4], they aligned video scenes with chapters of a novel. They first recognized actors, and then align dialogue inside the video with dialogue inside the novel. The visual information they used is appearance of actors, rather than semantic analysis.

The New2Image system [5] summarized news documents in the representation of images associated with sentences. Each word is represented by a vector generated by word2vec [6], and the vectors of words in the same sentence are averaged to be the representation of a sentence. With such representation, they clustered sentences of a document. By computing similarity between sentences and the document title, from each cluster one key sentence is selected. They then computed the distance between a key sentence and an image’s predefined caption. The most matched key sentences and the corresponding images finally form the news summary. This work pretty matches with our goal, and we will compare our work with it in the evaluation section.

III. IMAGE-TEXT ALIGNMENT FRAMEWORK

The proposed image-text alignment framework is shown in Fig. 3. Two pipelines are designed to process images and text, respectively. For images, we extract image features by a convolutional neural network [7]. Based on the codebook defined in [6], we represent each word by one-hot representation, and then transform it into a vector by the word2vec embedding [6]. Finally, we design fully-connected layers to map visual features and text features into a common space. Details of this framework are described as follows.

A. Image Representation

One image may convey several concepts that are located at different positions or presented at multiple scales. To consider multiple objects, we first use the selective search scheme [8] to detect object regions. From each region we extract Convolutional Neural Network (CNN) features based on the VGG-f model [7]. This CNN has five convolutional layers and three fully-connected layers. We take output of the last fully-connected layer as the representation of each object, which is a 4096-dimensional vector.

Given a CNN feature vector \mathbf{x}_i extracted from the i th object region, we attempt to learn a transformation matrix \mathbf{W}_A and a bias vector \mathbf{b}_A , and transform \mathbf{x}_i into a h -dimensional vector as

$$\mathbf{x}'_i = \mathbf{W}_A \mathbf{x}_i + \mathbf{b}_A, \quad (1)$$

where $\mathbf{W}_A \in \mathbb{R}^{h \times 4096}$ and \mathbf{b}_A is h -dimensional. This transformation is for reducing noises or correlation between dimensions. To embed the transformed vector \mathbf{x}'_i into a space that is common for images and text, we learn an embedding matrix \mathbf{W}_S and a bias vector \mathbf{b}_S to do embedding:

$$\mathbf{x}''_i = \mathbf{W}_S \mathbf{x}'_i + \mathbf{b}_S, \quad (2)$$

where $\mathbf{W}_S \in \mathbb{R}^{h \times h}$ and \mathbf{b}_S is h -dimensional. After embedding vectors extracted from all considered object regions, the set of vectors $X = \{\mathbf{x}''_1, \mathbf{x}''_2, \dots, \mathbf{x}''_M\}$ is used to represent an image. Currently, the value h is set as 1000.

B. Text Representation

To represent text, we first filter out (manually-defined) stop words, and then each word from a sentence is encoded as a one-hot representation \mathbf{c}_i based on the predefined codebook. One sentence is thus represented as a set of one-hot vectors $\{\mathbf{c}_i\}$, $i = 1, \dots, N$, where N is the number of words in the sentence. We then use word2vec to embed \mathbf{c}_i into \mathbf{y}_i , which dimension is 1000.

Following the same idea mentioned for images, we first transform text vectors and then embed them into a space common for text and images:

$$\mathbf{y}'_i = \mathbf{W}_B \mathbf{y}_i + \mathbf{b}_B, \quad (3)$$

$$\mathbf{y}''_i = \mathbf{W}_S \mathbf{y}'_i + \mathbf{b}_S, \quad (4)$$

where $\mathbf{W}_B \in \mathbb{R}^{h \times 1000}$, and \mathbf{b}_B is h -dimensional. The matrix \mathbf{W}_S and the bias vector \mathbf{b}_S are the same as that mentioned in eqn. (2). Similarly, the set of embedded h -dimensional vectors $Y = \{\mathbf{y}''_1, \mathbf{y}''_2, \dots, \mathbf{y}''_N\}$ is used to represent a sentence.

C. Model Learning

Based on the common feature space, we measure the similarity between image representation and text representation to find image-text alignment. We use dot product as the similarity measure. For the image I and the sentence L , similarity between them is defined as

$$Sim_{IL} = \frac{\sum_{\mathbf{x}''_i \in X} \exp(\max_{\mathbf{y}''_j \in Y} \mathbf{x}''_i{}^T \mathbf{y}_j)}{M}. \quad (5)$$

Every object region \mathbf{x}''_i has its best matched word, and the average dot product Sim_{IL} over all regions is calculated to find the best matched sentence for the image I .

On the other hand, to find the best matched image to a sentence L , the similarity value is defined as

$$Sim_{LI} = \frac{\sum_{\mathbf{y}''_j \in Y} \exp(\max_{\mathbf{x}''_i \in X} \mathbf{x}''_i{}^T \mathbf{y}_j)}{N}. \quad (6)$$

Every word has its best matched object region, and the average dot product Sim_{LI} is calculated to find the best matched

image for the sentence L . Finally, we define the degree of image-text alignment between the image I and the sentence L as $Sim = Sim_{IL} + Sim_{LI}$.

We use TensorFlow to build the model. Each training sample consists of one image and two sentences, where one sentence is positive, i.e., it really describes the image, and another sentence is negative. From the positive image-sentence pairs, we calculate the similarity value $Sim^{(pos)}$. Similarly, from negative image-sentence pairs, we calculate the similarity value as $Sim^{(neg)}$. In each mini-batch we sum all similarity values obtained from samples, and the loss function we would like to minimize is defined as

$$L(\theta) = \sum \frac{Sim^{(neg)}}{Sim^{(pos)}} \quad (7)$$

where $\theta = \{W_A, W_B, W_S, b_A, b_B, b_S\}$.

The activation function is Rectified Linear Unit, min-batch size is 10, the optimizer is Adam, and learning rate is 0.01. To mitigate overfitting, we employ dropout with ratio 0.2 for each fully-connected layer.

D. Blog Summarization

After embedding, sentences and images are transformed into a common feature space. By measuring distances between them, we find the most matched sentence for each image and use these image-sentence pairs to be the blog summary.

IV. EVALUATION RESULTS

A. Datasets

Training a deep neural network usually needs a large volume of training data. We therefore use the MSCOCO dataset [9] to train the proposed image-text alignment framework. The MSCOCO dataset consists of about 80,000 images, and each image is described by five sentences generated by crowd-workers on Amazon Mechanical Turk. We will construct the proposed model based on the MSCOCO dataset, with the settings determined in Sec. IV-B, and then use the model to do blog article summarization.

To evaluate performance of blog summarization, we collect a blog article dataset consisting of 48 articles, with the ground truth of image-text alignment labeled by manually. These blog articles are collected from A Luxury Travel Blog¹, which is one of the top 50 travel blogs in year 2016. Each article averagely has 5.6 images and 38.16 sentences. Based on the proposed framework, we select one sentence for each image to form the summary. If the selected sentence matches with the ground truth, we say this is a correct selection.

B. Finding Settings

The MSCOCO dataset is huge, which is good to achieve good performance, but training a complex model based on such dataset needs much time. At the stage of finding better settings for our model, we sample a subset of the MSCOCO dataset, including 10,000 images for training and 200 images for testing. At testing, for each image we randomly select one

TABLE I
SELECTION ACCURACY OBTAINED BASED ON DIFFERENT STRUCTURES.

Architecture	Selection Accuracy
[A1, B1, S1]	43.4%
[A1+A2, B1+B2, S1]	32.7%
[A1+A2, B1+B2, S1+S2]	23%

TABLE II
SELECTION ACCURACY OBTAINED BASED ON DIFFERENT LOSS FUNCTIONS.

	Sim_{IL}	Sim_{LI}	$Sim_{IL} + Sim_{LI}$
Accuracy	33.6%	37.7%	43.4%

of its corresponding positive sentences, and randomly select nine negative sentences from other images to for the test set. The goal of the constructed model is, for an image, to select the positive sentence from the test set. That is, the selection accuracy is averagely 10% if we just randomly guess.

Fig. 3 shows the framework consisting of one fully-connected layer for image feature transformation (with W_A and b_A), one fully-connected layer for text feature transformation (with W_B and b_B), and one fully-connected layer common for both modalities for embedding (with W_S and b_S). This setting is denoted as [A1, B1, S1] in Table I. We also try other configurations, e.g., two fully-connected layers for transforming two modalities, followed by one fully-connected layer for embedding (denoted as [A1+A2, B1+B2, S1] in Table I), or followed by two embedding layers [A1+A2, B1+B2, S1+S2]. Table I shows that the structure shown in Fig. 3 yields the best selection accuracy. We thus use this structure in the following experiments.

We also evaluate selection accuracy obtained based on different loss functions. Table II shows performance variations when only Sim_{IL} , only Sim_{LI} , or both are considered in calculating the loss defined in eqn. (7). As can be seen, jointly considering Sim_{IL} and Sim_{LI} works better.

To train the framework shown in Fig. 3, we need image-sentence pairs where each pair contains one image, one positive sentence, and one negative sentences. In the MSCOCO dataset, each image has five corresponding positive sentences. Therefore, we totally draw $80,000 \times 5 = 400,000$ image-sentence pairs to train the proposed framework.

C. Image-Text Alignment

We evaluate the proposed framework based on the blog article dataset. In the experiments, the alignment accuracy is calculated as the ratio of the number of images that really match with the truth sentences to the number of all test images.

There are often many words in a sentence. Averaging the similarity values of all words to an image may diminish the influence of important words. Therefore, we evaluate performance obtained based on different numbers of ‘‘best words’’ to represent a sentence. Table III shows performance variations when only the top p closest words are used in similarity measurement. As can be seen, better performance can be obtained when only the top one or top two words are

¹<http://www.aluxurytravelblog.com>

TABLE III
ALIGNMENT ACCURACY OBTAINED BASED ON CONSIDERING DIFFERENT NUMBERS OF WORDS TO REPRESENT A SENTENCE.

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
Accuracy	13.75%	13.75%	12.64%	11.52%	11.52%

TABLE IV
ALIGNMENT ACCURACY OBTAINED BASED ON DIFFERENT METHODS.

	Random	[5]	CaptionBot+word2vec	Our model
Accuracy	4.83%	7.06%	13.01%	13.75%

used to calculate similarity.

With the best settings mentioned above, we now compare the proposed framework with two methods. In the first method, for some image I , we generate its image caption by CaptionBot [2]. We then calculate the similarity values between this image caption and sentences in the blog article, and finally find the most matched sentence for the image I . The word2vec module is again used to represent sentences or image caption as vectors. The second comparison method is News2Image [5]. They compute the distance between extracted key sentences and existing image caption. We again utilize CaptionBot to generate image caption, and implement the method in [5].

Table IV shows that the proposed method achieves performance better than others. The News2Image system uses average information of key sentences, and match it with image captions. The obtained performance is not very good. Directly calculating similarity between sentences and the generated image captions yields better performance, but relatively more performance improvement can be made by our method.

Furthermore, we observe that an image usually only relates to sentences around it. Only matching sentences around an image can largely reduce search space. In the following, we develop two variants to consider such spatial information. (1) If one blog article has K images, we equally divide the article into K segments. The i th image is only matched with the i th segment of the article. (2) Centered by an image, we only match the image with E nearby sentences.

Table V shows the obtained results by considering spatial information. We see that spatial information is very useful in image-text alignment. The accuracy of alignment is largely improved when only ten nearby sentences are used.

We also conduct a subjective test to do performance comparison. We invited twenty subjects in the evaluation. Each subject was given twenty images associated with two sentences, where one sentence is selected by the proposed method, and another sentence is selected by the News2Image method. The subjects were asked to examine which sentence is more appropriate to describe the corresponding image. Fig. 4 shows the compari-

TABLE V
ALIGNMENT ACCURACY OBTAINED BY CONSIDERING SPATIAL INFORMATION.

	(1)	(2) $E = 8$	(2) $E = 10$	(2) $E = 12$	w/o spatial
Acc.	21.56%	46.10%	47.58%	46.84%	13.75%

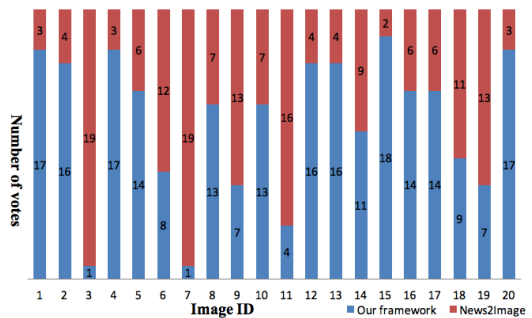


Fig. 4. Subjective performance comparison for twenty sample images.

son results, where the x axis denotes the twenty test images, and the y axis denotes the frequency that the better sentence is selected by which method. Overall, our method outperforms the News2Image method in thirteen of twenty images.

V. CONCLUSION

We propose a blog article summarization system based on a deep-based image-text alignment framework. Visual information of images is first extracted by a CNN, and then embedded into a space common with the text information. Similarly, text information is transformed into vectors, and then embedded into the common space. With this embedding, we calculate similarity between any image-sentence pair, and then select the sentence that is semantically most similar to the image. With such alignment, we summarize a blog article into image-sentence pairs. Comparing with image captioning, our result is more readable and matches with the author's context well.

ACKNOWLEDGMENT

This work was partially supported by the Ministry of Science and Technology of Taiwan under the grant MOST 105-2628-E-194-001-MY2 and MOST 106-3114-E-002-009.

REFERENCES

- [1] Y. Feng and M. Lapata, "Automatic caption generation for news images," *IEEE TPAMI*, vol. 35, no. 4, pp. 797–812, 2013.
- [2] K. Tran, X. He, L. Zhang, J. Sun, C. Carapcea, C. Thrasher, C. Buehler, and C. Sienkiewicz, "Rich image captioning in the wild," in *Proc. of CVPR Workshops*, vol. 26, 2016, pp. 434–441.
- [3] R. Yazhini and P. V. Raja, "Automatic summarizer for mobile devices using sentence ranking measure," in *Proc. of International Conference on Recent Trends in Information Technology*, 2014.
- [4] M. Tapaswi, M. Bauml, and R. Stiefelhagen, "Book2movie: Aligning video scenes with book chapters," in *Proc. of CVPR*, 2015, pp. 1827–1835.
- [5] J.-W. Ha, D. Kang, H. Pyo, and J. Kim, "News2images: Automatically summarizing news articles into image-based contents via deep learning," in *Proc. of RecSys*, 2015, pp. 27–32.
- [6] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. of NIPS*, 2013, pp. 3111–3119.
- [7] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. of the BMVC*, 2014.
- [8] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *IJCV*, vol. 104, no. 2, pp. 154–171, 2013.
- [9] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, P. D. D. Ramanan, and C. Zitnick, "Microsoft coco: Common objects in context," in *Proc. of ECCV*, 2014, pp. 740–755.