Enhancing Bidirectional Association between Deep Image Representations and Loosely Correlated Texts

Qiuwen Chen, Qinru Qiu
Department of Electrical Engineering and Computer Science, Syracuse University, NY 13244, USA
Email: {qchen14, qiqiu}@syr.edu

***Abstract*—The problem of bridging the gap between image and natural language has gained more and more attention in recent years. This paper continues to push the study and improves the bidirectional retrieval performance across the modalities. Unlike previous works that target at single sentence densely describing the image objects, we extend the focus to associating deep image representations with noisy texts that are only loosely correlated. Based on text-image fragment embedding, our model employs a sequential configuration, connects two embedding stages together. The first stage learns the relevancy of the text fragments, and the second stage uses the filtered output from the first one to improve the matching results. The model also integrates multiple convolutional neural networks (CNN) to construct the image fragments, in which rich context information such as human faces can be extracted to increase the alignment accuracy. The proposed method is evaluated with both synthetic dataset and real-world dataset collected from picture news website. The results show up to 50% ranking performance improvement over the comparison models.**

I. Introduction

Learning to associate images with loosely correlated texts is an important feature for many retrieval applications. From textual input, we can search for images with natural language, or intelligently assign illustrations to news articles. Given an image, it is possible to generate caption automatically, or to locate relevant documents from a text database. In this paper, we seek to improve the cross-modal matching performance given that the text-image parallel datasets are not specially constructed for query purposes.

There has been extensive researches on bidirectional mapping between images and words/sentences [9], [15]–[18], [24]. Learning an embedding space of different modalities is proved to be effective for high-quality datasets with descriptive sentences. But the task becomes much more challenging when tightly coupled text-image pairs are not available. The parallel text may contain many contents that do not directly describe the image, or conversely, the image could show objects that are otherwise not discriminative without proper contexts. Compare the example text-image pairs in Fig. 1, while most of the words in the Flickr8k [12] sentence are densely corresponded to the objects in the image, only a small portion of the Reuters Picture News [32] paragraph is explicitly describing the contents of its illustration. The rest of the paragraph is co-occurring just for the news background and may even cause overfitting to learning-based methods. Topic modeling has been studied to summarize text corpus [3], [4], [8], but they are mostly tuned for automatic annotation applications

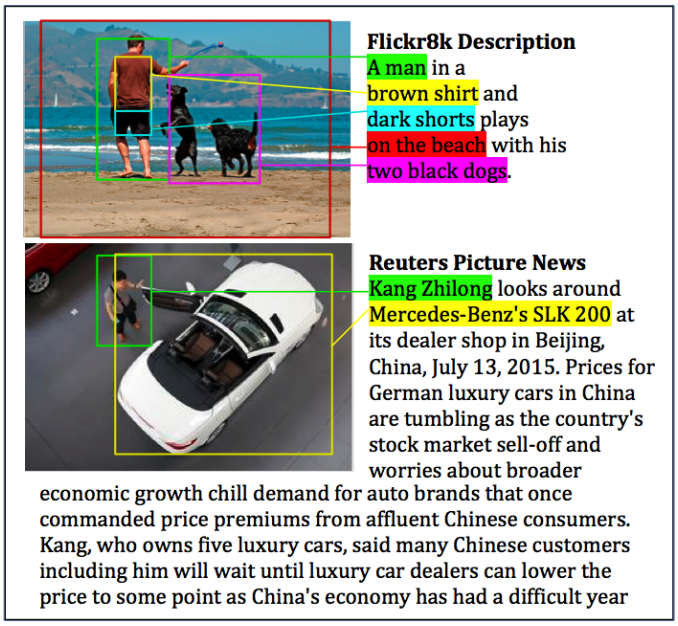


Fig. 1. Comparison between descriptive text-image pair and picture news

and are difficult to associate with continuous image features as embedding-based methods do. Even if the descriptive part of the news is somehow extracted, it is still hard to accurately map the words to the image components without recognizing the person's identity or the car model. Therefore, in order to successfully match the noisy text-image pairs, we need to extract the useful portion of the paragraphs and to enrich the image representations.

The approach in this paper follows the method of learning an embedding layer between texts and images. An image is partitioned into multiple regions of objects and has the region features extracted using convolutional neural networks (CNN) [19], [21], [29]. For a text paragraph, its dependent word pairs are used as the semantic fragments. Both image regions and word pairs are treated as bags of fragments and matched in the embedding space [15], [16]. To aid the mapping between images and noisy paragraphs, we propose two improvements to the fragment space. First, instead of learning a single level of embedding, we cascade embedding optimizers. The result from the upstream embedding is analyzed to determine the relevancy and discriminative power of the text fragments. Then the information is forwarded to the downstream embedding

to suppress the noisy text portion and improve the secondary learning process. Second, we integrate multiple CNN's that are tuned for different contexts to construct the image fragments. The new fragments help the embedding not just match for the object-level image features, but also adopt diversified information such as facial characteristics of persons.

The contributions of this work are as the followings. In section III, we analyze the problem of matching noisy paragraph and images, select the proper optimization objectives, and also propose an equivalent implementation of the alignment scores for accelerating the computation. In section IV we design an cascade configuration of text-image matchers to refine the discrimintive set of text. In section V, we integrate multiple CNN's to enrich the image representations and use facial recognition network as the demonstration. Finally in section VI, we evaluate the proposed methods with both synthetic dataset and real datasets of picture news collected from Reuters Picture News [32].

II. Related Work

It is an emerging topic of learning to bridge the gap between image and natural languages. Some works [20], [26], [35] have focused on generating novel captions from query images. A typical pipeline in Vinyals et al. [34] was that the image was first passed to the CNN [33] and had its compact representation extracted. Then the image representation was treated as the initial word input to the semantic space and used to generate a sentence label using a long-short term memory (LSTM) [11] predictor. Other works [15], [17], [18], [24] have focused on learning an embedding space for bidirectional mapping. Frome et al. [9] converted the whole images and the word labels into a common embedding space and defined a hinge rank loss to align the correct pairs. Instead of using a common embedding space, Karpathy et al. [16] broke the images into multiple objects using regional CNN [10] and the sentences into dependent word pairs using Stanford CoreNLP toolkit [25], and then learned to compute the similarity scores based on the visual-semantic fragment embedding. Most of the existing works have been focused on query-like text-image datasets such as Flickr8k [12] and Pascal1k [31], and achieved state-of-the-art accuracy. Only a small body of studies considered loosely correlated pairs, such as picture news.

To obtain neural descriptors of images, many studies have been conducted for different applications. For instance, the networks in Krizhevsky et al. [19] and Szegedy et al. [33] were dedicated to object classification for ImageNet challenge [7]. The VGG Face Descriptor [29] was tuned for celebrity identifications. Zhou et al. [36] specialized for scene classification. For the text representation, works have been conducted to convert words or sentences into vector space [1], [13], [28].

Topic modeling such as Latent Dirichlet Allocation (LDA) [3] has been an effective way to extract the essential part of large text bodies. There has been studies based on LDA for word sense disambiguation [22] and semantic category classification [5]. As for news media, Cano et al. [4] explored different methods in finding keywords from Twitter messages. Feng et al. [8] connected the image and text modalities by clustering the SIFT features [23] of image regions into discrete words, and building a mixed LDA model with both visual and semantic words. They performed image annotation on BBC news dataset. The discretization of images may impose information loss compared to the embedding-based methods, but the key idea of extracting essential texts could be beneficial.

III. Visual-Semantic Embedding

The bidirectional retrieval task is essentially a ranking problem. For each text-image pair, an alignment score is calculated to indicate how closely correlated a text sample and an image sample are. The scores of all pairs in the searching space are ranked among the image peers or the text peers. The top-ranked images are the search result of a text query, or vice versa. For the datasets which are targeted by this paper, the text queries are not short descriptive sentences that frequently refer to the image contents. They could be long paragraphs with only parts of them strongly connected to the images.

A. Text and Image Representations

Following the deep embedding approach [15], [16], both the texts and images are broken into fine fragments. For images, RCNN [10] and Caffe [14] are used to detect the object regions. Each region forms an *image fragments*. The network is pre-trained with ImageNet [7] data and fine-tuned towards 200 object classes. Every image is represented by a bag of regions containing the whole image and up to 19 RCNN detections. The detection regions are selected by highest classification probabilities. The embedding of the i^{th} image fragment v_i is calculated as in equation (1).

$$v_i = W_v[\text{CNN}(R_i)] + b_v \tag{1}$$

where R_i is the pixels in region i and CNN(.) outputs the 4096-dimensional features of the fully-connected hidden layer (*fc7*) immediately before the RCNN classifier. W_v and b_v are learnt parameters. When the size of the embedding space is d, W_v is a $d \times 4096$ matrix.

The text paragraphs are analyzed using Stanford CoreNLP [25] and have their word dependencies extracted. Each pair of dependent words is grouped as a *text fragment* and the paragraph is represented by a bag of such fragments. The embedding of the t^{th} fragment s_t is computed by equation (2).

$$s_t = f(W_s \begin{bmatrix} {w_t}^p \\ {w_t}^c \end{bmatrix} + b_s) \tag{2}$$

where ${w_t}^p$ and ${w_t}^c$ are the 200-dimensional vectors of the parent and child words of the dependent pair. The vector representations are learned by unsupervised objective [13]. W_s is a $d \times 400$ matrix that transforms the lumped word pairs to the embedding space. The activation function f is the Rectifying Linear Unit ($ReLU$).

B. Selection of Objectives

The correlation between text fragment t and image fragment i is computed as the dot product of their embedding vectors, ${v_i}{s_t}^T$. One way of defining the alignment score [15] between the j^{th} image and k^{th} text sample is in equation (3), and the global alignment objective in equation (4) drives the optimization.

$$A_{j,k} = \sum_{t \in T_k} \max_{i \in I_j} {v_i}{s_t}^T \tag{3}$$

$$\begin{aligned} loss_G = \sum_j [\sum_k \max(0, A_{j,k} - A_{j,j} + \Delta) \\ + \sum_k \max(0, A_{k,j} - A_{j,j} + \Delta)] \end{aligned} \tag{4}$$

Here, T_k is the set of dependent word pairs of the k^{th} text sample and I_j denotes the regions of the j^{th} image. Δ is a constant margin that valued 40 in our experiments. The loss function essentially maximizes the correct alignment against the other images and texts. Compared with their former objective [16], the formulation simplifies the model and improves the ranking performance. However, such formulation assumes that each text fragment can only align to one image region with the highest dot product as in equation (3). This assumption works well for descriptive sentences because they are always directly referring to image regions. However, noisy paragraphs do not hold the same property. From our observation, it is possible for a word in the news article to align with multiple image regions. Therefore, we choose to use the original formulation [16] that combines the local objective and the global objective. The loss is defined by equation (7) with the alignment calculation (5).

$$A_{j,k} = \sum_{t \in T_k} \sum_{i \in I_j} {v_i}{s_t}^T \tag{5}$$

$$loss_L = \sum_i \sum_t \max(0, 1 - y_{i,t} {v_i}{s_t}^T) \tag{6}$$

$$loss = \alpha loss_G + \beta loss_L \tag{7}$$

This formulation of alignment score allows a text fragment to align with multiple regions. In the early training epochs, $y_{i,t}$ is defined as +1 when v_i and s_t occur together in a correct image-text pair (i.e. $j = k$ for $i \in I_j, t \in T_k$), and -1 otherwise. In the later epochs, $y_{i,t}$ is adjusted by Multi-Instance Learning (MIL) [6]. $y_{i,t}$ is +1 only if in a correct pair, ${v_i}{s_t}^T > 0$ or $i = \text{argmax}_{i' \in I_j}({v_{i'}}{s_t}^T)$. The overall loss function is a weighted linear combination of the local loss (6) and the global loss (4) with biases $\alpha = 0.5$ and $\beta = 1.0$.

For testing, the alignment scores $A_{j,k}$ are calculated using the trained parameters (W_v, b_v, W_s and b_s). The image search (i.e. use a text sample to query the most likely image) is done by fixing a text sample k and ranking the alignment scores of all candidate images. And the text search is similar by ranking the text candidates with a fixed image j.

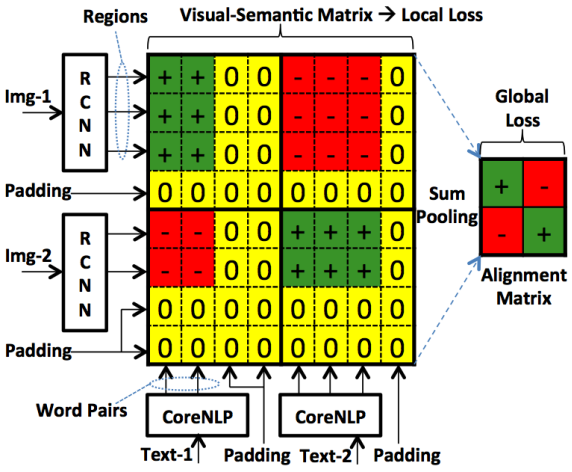


Fig. 2. Computation of Alignment Matrix

C. Speed-up with Fragment Padding

For an optimization mini-batch H, the inner products of all image and text fragments (${v_i}{s_t}^T, i \in I_j, t \in T_k$ and $j, k \in H$) form the *visual-semantic matrix*. We call the stacked $A_{j,k}$ of all text-image pairs the *alignment matrix*, from which the global loss can be quickly obtained with a few matrix operations. Essentially, an entry in the alignment matrix ($A_{j,k}$) is computed as the sum of all elements in its corresponding visual-semantic sub-matrix (a *patch*). Since images may have different number of regions and paragraphs are also diversified in the number of words, the sizes of the patches differ from each other. To calculate these alignments using theano [2], a straightforward implementation is to use a *scan* node to loop over the dimensions. However, it results in slow computation.

To improve the performance, we insert padding fragments (Fig. 2) to both the images and the texts. The j^{th} image fragment bag ${B_j}^v$ and the k^{th} text fragment bag ${B_k}^s$ are padded with zero fragments as in equation (8), in which all the texts and images will have the same number of fragments. The resultant patches in the visual-semantic matrix are of size $(N^v \times N^s)$.

$$\begin{aligned} {B_j}^v &= \{R_i | i \in I_j\} + \{0\} \times {N_j}^v, \text{ s.t. } |{B_j}^v| = N^v \\ {B_k}^s &= \{\begin{bmatrix} {w_t}^p \\ {w_t}^c \end{bmatrix} | t \in T_k\} + \{0\} \times {N_k}^s, \text{ s.t. } |{B_k}^s| = N^s \end{aligned} \tag{8}$$

These padding fragments produce inner products of zeros, and thus will not contribute to the local loss or the global loss. But with the equally sized patches, we can use a standard sum-pooling process supported by theano to obtain the alignment matrix. The pooling operations are optimized in software implementations and better for vectorization than the loops do. Therefore, the padding helps accelerate the computation by removing the need of handling differently sized patches.

IV. Text Fragment Filtering

A single stage of match embedding works well for short sentences that densely correlate with the images. However,

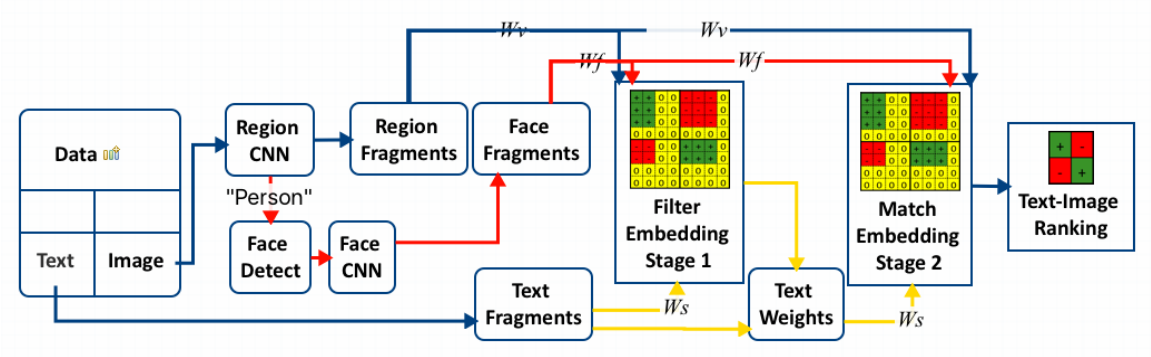


Fig. 3. Configuration of fragment filtering and fragment enrichment

loosely coupled texts such as picture news pose new challenges to the model. Since a lot of words in the news articles are not explicitly describing the images, they may cause overfitting and divert the optimization from those text fragments that really differentiate. In order to filter out the interfering fragments, we propose a fragment importance measure, and use a sequential architecture to improve the text-image association.

A. Fragment Importance Measure

The i, t^{th} entry in the visual-semantic matrix indicates how well the image fragment i correlate with the text fragment t. When the text body contains noises, the dot products $(v_i {s_t}^T)$ may not produce the optimal matching, but they are still valid indicators of whether a fragment is useful for the association optimization. We define p_t in equation (9) the importance measure of the t^{th} text fragments.

$$p_t = g_{j \in \text{Imgs}}(\sum_{i \in I_j} v_i {s_t}^T) \tag{9}$$

Here, $g_{j \in Imgs}$ is a *Reduction Function* (e.g. $\sum_j$) that applies to the image population. The idea is that the larger the score is, the more likely the text fragment receives diversified matching results over different image regions. If we can make these informative fragments contribute more to the association optimizer, then we have a better chance to achieve an accurate text-image matching.

B. Cascade Embedding Stages

By equation (9), we know which text fragments are more useful in the association. Now a model is needed to integrate the importance measure to the optimizer. We propose to connect two fragment embedding stages as the yellow path in Fig. 3. The text-image fragments are passed to the first stage to train the *Filter Embedding*. The first stage does not produce the ranking, but outputs the fragment importance measures for the texts. The measures are converted to text weights that are applied to the fragments at the second embedding stage. The second stage, *Match Embedding* is trained with the filtered text fragments with weighted contributions to the loss. Match embedding produces the improved alignment matrix that generates the final ranking results.

Filter embedding is trained to identify the fragment importance. The importance measures are converted to text weights using equation (10).

$$m_t = \frac{|T_k|}{\sum_{t' \in T_k} p_{t'}} p_t, \forall t \in T_k \tag{10}$$

Here, the weight is essentially the normalized fragment importance measure with respect to the number of the fragments in the belonging text. While favoring those informative words, the normalization keeps the total "energy" of the text samples the same (i.e. $\sum_{t \in T_k} m_t = |T_k|$) to prevent large swings of the training loss.

The text weights are applied to the original text fragments. The second stage is then trained with the weighted text fragments defined in equation (11). Activation f is *ReLU*.

$$s_t = f[W_s(m_t \begin{bmatrix} {w_t}^p \\ {w_t}^c \end{bmatrix}) + b_s) \tag{11}$$

In this configuration, the word vectors are multiplied by the weight values m_t so that the important fragments are given higher weights and vice versa. The idea is that the non-informative fragments contribute less to the loss function in equation (7), so that the parameters W_s and b_s are able to "focus" on the important word fragments that are not discriminated optimally during the first stage. The dot products related to the noisy text fragments are forced to be near zero by the weights. Thus no matter how the parameters interact with the noisy words, they do not affect the final text-image ranking much.

V. Image Fragment Enrichment

The image features extracted by RCNN are tuned for object recognition. For text-image datasets with descriptive sentences, the level of knowledge is sufficient since the sentences are directly describing the objects in the images. However, in picture news, the images may contain different levels of meanings that cannot be captured by the object features. For example, the identities of the persons appeared in the news picture may help differentiate events, and thus improve the association learning, but simply recognizing the person objects doesn't provide such information. Therefore we use image

fragments extracted by different CNN's to enrich the image understandings. Specifically, we extract the face features from the images.

Fig. 3 red path shows the workflow of extracting face fragments. The regions classified as "person" by RCNN are passed to the DPM face detector [27] to find the accurate face area. Then using VGG Face Descriptor [29], we extract the face features. The face features are then converted into the embedding space by equation (12).

$$v_l = W_f[\text{CNN}_F(R_l)] + b_f \tag{12}$$

The deep network $\text{CNN}_F(.)$ converts the pixels of the l^{th} detected face, R_l into a 4096-dimensional feature vector. Parameter W_f and b_f turns the face features into the image embedding v_l. The face fragments are placed along with the other image fragments to compute the alignment matrix. Because the number of faces detected in each image is small (usually less than 3), the face alignment score resembles equation (3). The final alignment score with face feature integrated is redefined in equation (13).

$$A_{j,k} = \sum_{t \in T_k} (\sum_{i \in I_j} v_i {s_t}^T + \max_{l \in F_j} v_l {s_t}^T) \tag{13}$$

where F_j is the set of faces detected in the j^{th} image.

VI. Evaluation

A. Datasets

Pascal1k with noises. Pascal1k [31] dataset contains 1000 images, each of which is annotated by 5 independent sentences. We append to the sentences random texts grabbed from news articles, and the 5 sentences of each image have the same random text added. With this setup, we know that the first sentence of each text sample always contains the most information. This setup serves as a synthetic baseline for text filtering.

Reuters Picture News. We develop a crawler to download the thumbnail images along with their news articles from Reuters Picture News [32]. For each of the news categories (Scitech5k, Business5k and Politics5k), 5000 images with their associated articles longer than 70 words are collected. We also build a dataset of 15000 samples (Mixed15k) with news from all three categories. For face fragment evaluation, we construct a dataset of 1000 samples (People1k) with faces detected by RCNN [10] and DPM face detector [27].

B. Comparison Methods

For comparative study in image-text retrievals, we reproduce 4 baseline models.

Joint topics. We use K-means to cluster the RCNN [10] image regions into 1000 discrete visual words, and train an LDA [3] model with the joint corpus of both visual words and semantic words (similar to MixedLDA [8]). The LDA model is trained with 800 latent topics. Using LDA, the probability of each visual (semantic) word can be inferred from the latent topic distribution, which is inferred from the query bag of semantic (visual) words. We use the sum of the logarithm likelihoods of the visual (semantic) words as the alignment score.

DeViSE [9]. The work connects the modalities by minimizing the alignment loss between single words and images. It does not handle image or text as bag of fragments, but it can be treated as a special case for the fragment embedding. The word vectors in a paragraph are averaged (L2-normalized) to one word fragment, and the regions detected in the same image are summed up to one image fragment. Only the global loss in equation (4) is applied during the optimization.

DeFrag [16]. The approach improves the performance by breaking the text and image into fragments. The fragment embedding is optimized by a mixed objective (global loss + local loss + MIL). We implement DeFrag with theano [2] for our customized configurations, and use it as the building blocks for our cascade configuration described in Section IV.

DepTree edges [15]. This method is the simplified extension of DeFrag. It removes the local loss, and uses the alternative alignment calculation defined by Equation (3).

C. Experiment Setup

For embedding optimization, we use stochastic gradient descent with momentum of 0.9. For Pascal1k and the noisy version, the dimension of embedding space (i.e. v_i and s_t) is 700, the mini-batch contains 35 text-image pairs and the reduction function g_j for equation (9) is variance var_j. For the picture news datasets, we use 1000-dimensional embedding, mini-batch size of 100 and the sum reduction function $\sum_j$. For all datasets, 80% of the samples are used for training and the rest two populations of 10% samples are used for validation and testing respectively. Take Pascal1k for example, we use 800 samples for training, 100 for validation and 100 for testing. Both DeFrag and our model use MIL [6] for the local losses.

For retrieval tests, we follow the description in section III-B. The performance metrics are **R@K** and **Med** (Table I II). R@K is the percentage of the correct alignments that are ranked among the top K retrieval results (higher is better). Med is the medium rank of the correct samples (lower is better).

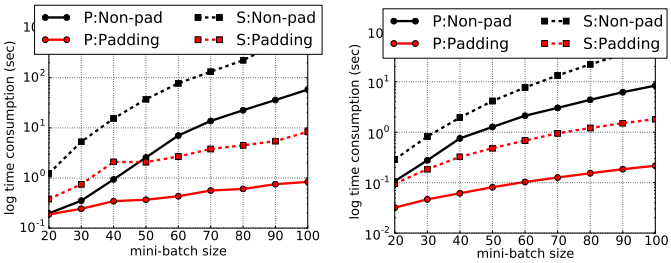
D. Improvement in Computation Speed

In this experiment, we evaluate the computation performance boost brought by padding the fragments to equal patches. The implementation without padding uses a *scan* node [2] to loop over patches of different sizes. For equal patches with padding fragments, a sum-pooling operation based on *images2neibs* [2] is used. We test the per-batch time consumptions for the optimization of loss equation (7).

Two platforms are tested with the datasets. On a laptop with Intel Core i5 4250U at 1.3GHz, the padding-based implementation (*:Padding) outperforms the loop-based implementation (*-Non-pad) on both Pascal1k dataset (P:*) and Scitech5k news dataset (S:*), and provides 10X to 100X speed-up (Fig. 4a). The sum-pooling operation is more suitable for vectorization than the scan node does. On our server with Intel Xeon W5580 at 3.2GHz and NVIDIA Tesla C2075 with 448 CUDA

TABLE I
TEXT-IMAGE RETRIEVAL RESULTS ON PASCAL1K

	Image Retrieval				Text Retrieval			
Implementation	**R@1**	**R@5**	**R@10**	**Med**	**R@1**	**R@5**	**R@10**	**Med**
Non-padding	27.2	62.2	82.2	3.0	25.0	67.0	80.0	2.0
Padding	25.6	66.6	84.0	2.0	27.0	60.0	74.0	3.0



(a) Laptop (b) Server

Fig. 4. Computation Speed Comparisons

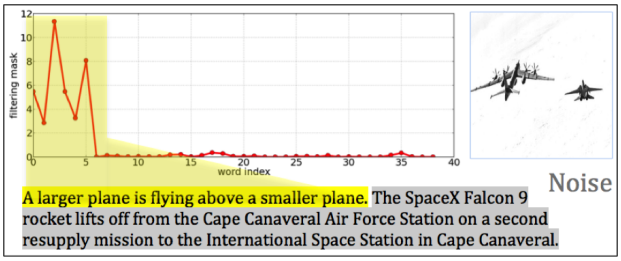


Fig. 5. Weight output from filter embedding on Pascal1k with noises

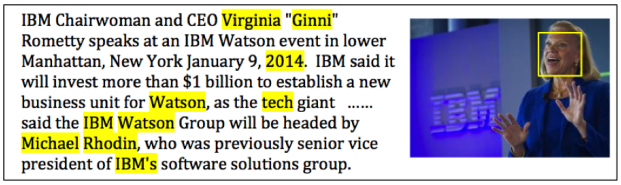


Fig. 6. Top 10 fragments with the highest dot products to the detected face

cores at 1.15GHz, our fragment padding also accelerates the optimization process (Fig. 4b). At 100 batch size, the per-batch runtime on Scitech5k dataset is around 2 seconds with fragment padding, while the loop-based implementation consumes more than 50 seconds. The pooling operation can better utilize the GPU resources. Although fragment padding produces slightly larger visual-semantic matrix, the removal of scan nodes provides substantial improvement in training speed.

We also test the retrieval performance for padding-based implementation on Pascal1k dataset [31]. As shown in Table I, fragment padding (Padding) does not degrade the accuracy. It achieves equivalent performance as compared to the scan-based (Non-padding) approach, while significantly improves the computation speed.

E. Results of Text-Image Retrievals

In this section, we evaluate the accuracy of comparison models on both synthetic dataset and picture news (Table II).

We first perform retrieval tests on Pascal1k with noises. Fig. 5 shows the output weights of the filtering embedding obtained from an example piece of text. It is observed that the filter is able to capture the informative text fragments, i.e. the original description, and suppress the noises that we appended. The method correctly identifies the first part of the text as the most important, and assigns it high weights. The retrieval performances for both texts and images are improved significantly compared to the baseline methods. The R@1 measures are about 40% better than the second best model, DeFrag. This validates our assumption of using the filter embedding to extract the useful part of the texts.

Secondly, evaluations are done on the real picture news of different categories. Generally, Joint topics do not perform well because clustering regions into words causes loss of visual information. It has relatively better results on People1k as the images usually contain less types of objects. Also, DeViSE does not associate the text-image pairs as good as those fragment-based approaches, because using only the whole picture and averaged word vectors loses the details of the images and texts. The DepTree edges method uses the simplified alignment formulation and only considers the global objective [15]. This assumes that each text fragment aligns to one image region. When many of the text fragments align to none or multiple regions, this assumption reduces accuracy. The ranking results (R@K) are worse than its more complete peer [16] with both local and global objectives.

The proposed method of text fragment filtering achieves substantial performance boost on the picture news mapping. On Scitech5k dataset, fragment filtering outperforms the second best method by around 10% in the ranking metrics. For Business5k dataset, our method produces better image ranking than those of the comparison methods, and competitive text ranking with DeFrag. On Politics5k dataset, fragment filtering still generates around 10% R@K improvement over the best baseline results. For Mixed15k, all approaches perform worse than they do on the smaller datasets, because Mixed15k is larger and more difficult. Our method adapts to the mixed news categories and produces the best ranking results among the comparison methods.

On People1k dataset, fragment filtering (F1) generates better ranking results over the first three methods. By integrating the deep face representations (F2), we outperform the best baseline approach DeFrag by 50% in the R@10 score. The face fragments provide another layer of context matching. The example in Fig. 6 highlights the child words of the dependent

TABLE II
TEXT-IMAGE RETRIEVAL RESULTS ON NOISY DATASETS

	Image Retrieval				Text Retrieval			
Model	**R@1**	**R@5**	**R@10**	**Med**	**R@1**	**R@5**	**R@10**	**Med**
Pascal1k with noises								
Joint topics	3.0	15.6	23.6	36.0	4.0	11.0	15.0	85.5
DeViSE	6.2	17.8	31.0	22.0	6.0	7.0	14.0	70.5
DepTree edges	4.8	20.0	36.2	16.5	6.0	20.0	23.0	38.0
DeFrag	12.6	42.2	63.6	6.0	11.0	31.0	43.0	14.5
Fragment filtering	17.6	51.2	68.8	4.0	16.0	43.0	55.0	8.0
Scitech5k								
Joint topics	4.6	12.0	15.8	119.5	4.2	8.8	12.2	183.0
DeViSE	3.2	14.0	24.0	38.5	5.0	18.4	30.4	30.0
DepTree edges	9.0	22.0	32.0	26.5	7.6	26.4	36.2	23.0
DeFrag	12.0	29.8	39.4	18.0	11.2	30.6	41.6	15.0
Fragment filtering	14.0	31.8	42.8	15.0	13.4	32.8	46.2	12.0
Business5k								
Joint topics	4.8	10.8	17.0	88.5	2.2	6.2	8.0	177.5
DeViSE	4.4	17.2	28.0	30.5	6.2	22.6	32.6	23.0
DepTree edges	6.4	22.6	33.4	23.5	7.2	26.8	37.0	17.0
DeFrag	11.6	31.2	41.2	14.0	12.8	36.2	48.4	10.5
Fragment filtering	11.2	33.0	45.8	12.5	13.0	36.2	47.2	12.0
Politics5k								
Joint topics	1.2	7.2	10.2	159.0	1.4	5.6	8.0	209.5
DeViSE	2.0	9.4	19.2	50.5	4.2	11.8	22.0	43.0
DepTree edges	4.2	15.4	21.4	46.0	7.0	19.8	31.6	35.0
DeFrag	8.0	22.2	31.0	25.0	1.8	22.8	33.2	22.0
Fragment filtering	9.0	25.6	35.2	19.5	8.2	26.8	36.2	19.5
Mixed15k								
Joint topics	1.7	4.7	6.6	426.0	1.4	2.7	3.7	579.5
DeViSE	2.0	8.7	14.1	88.5	2.5	10.4	17.3	69.5
DepTree edges	4.4	14.9	21.0	59.5	4.0	13.4	22.7	50.0
DeFrag	6.2	19.7	28.8	34.0	2.4	20.4	31.9	29.0
Fragment filtering	8.8	24.9	34.0	31.0	3.6	24.9	34.7	25.0
People1k								
Joint topics	13.7	25.5	31.4	27.0	10.8	21.6	29.4	27.5
DeViSE	2.9	13.7	25.5	23.0	4.9	23.5	41.2	17.5
DepTree edges	7.8	27.5	37.3	18.0	5.9	24.5	38.2	13.0
DeFrag	12.7	30.4	35.3	16.5	5.9	28.4	37.3	15.0
Fragment filtering (F1)	14.7	33.3	44.1	11.5	16.7	33.3	38.2	16.5
Face fragments (F2)	22.5	46.1	58.8	5.0	15.7	45.1	54.9	5.5
F1 + F2	31.4	46.1	59.8	6.0	22.5	47.0	58.8	5.5

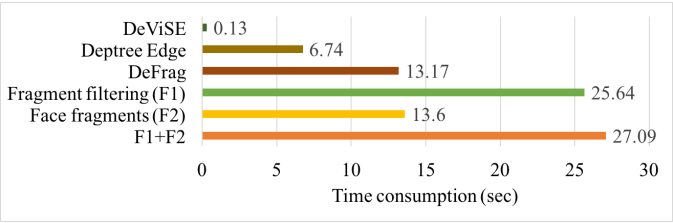


Fig. 7. Training time for 800 samples on People1k

word pairs whose fragments produce the highest dot products $v_l {s_t}^T$ with the detected face. The face of IBM's CEO is strongly correlated with the company, her and her colleague's name, and "Watson". Some images that are previously not distinguishable can now be better identified by the person's facial characteristics. Finally, the combination of both text fragment filtering and image fragment enrichment (F1 + F2) obtains more accurate rankings compared to the two individual enhancements. It reaches 31.4 of R@1 for image retrieval.

Finally, Fig. 7 shows the times for training 800 samples on People1k data. DeViSE is fast because it does not handle fragments. So for a sample pair with 10 regions and 10 words, the size of the visual-semantic matrix is only 1/100 of the other methods'. DeFrag is slower than Deptree edge since the former optimizes both local and global costs. Fragment filtering sequentially connect the embedding stages, so the training complexity is about two times as that of DeFrag. Adding the face fragments only add a small overhead to the counterparts, because the number of faces in an image is usually small. The retrieval is accomplished by doing the forward pass with the network, so the time consumption is proportional to the training.

F. Limitations

From the text side, the fragments rely on Stanford CoreNLP [25] to extract the dependency edges. This process is computationally expensive and may lose semantic information. BRNN [15] has used Recurrent Neural Networks to extract long-term concepts expressed by each word. Since our enhancements work in fragment level, it can be adapted to the BRNN fragments without much difficulty. Also, the proposed work

only does reduction to the text fragments, but sometimes it is helpful to create richer text fragments (addition). For example, when seeing "Trump" and "Hilary" in the text, bringing up a new fragment such as "election" could improve the association learning. Therefore, inference-based model such as ITRS [30] can be integrated into the configuration. Finally, the improvement brought by the face descriptor is subject to the face detection, a more elastic approach is needed when the dataset lacks of facial information.

VII. Conclusion

This paper addresses the problem of associating images with noisy texts. We first modify the implementation by padding empty fragments to generate visual-semantic matrix with equally sized patches, which accelerate the speed of computation. Second, an embedding cascade configuration is designed to suppress the noisy part of the texts, so that in the second match embedding stage the optimization can be more effective in distinguishing the correct alignments. Third, we integrate face CNN to the image fragment generation in order to interpret richer information from the images. We show the improvements of our methods over the existing works on both synthetic dataset and real datasets of picture news.

References

[1] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003.

[2] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a cpu and gpu math expression compiler. In *Proceedings of the Python for scientific computing conference (SciPy)*, volume 4, page 3. Austin, TX, 2010.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[4] A. E. Cano Basave, Y. He, and R. Xu. Automatic labelling of topic models learned from twitter by summarisation. In *Association for Computational Linguistics (ACL)*, 2014.

[5] J. Z. Chang, R. T.-H. Tsai, and J. S. Chang. Wikisense: Supersense tagging of wikipedia named entities based wordnet. In *PACLIC*, pages 72–81, 2009.

[6] Y. Chen, J. Bi, and J. Z. Wang. Miles: Multiple-instance learning via embedded instance selection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):1931–1947, 2006.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[8] Y. Feng and M. Lapata. Automatic caption generation for news images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(4):797–812, 2013.

[9] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, pages 2121–2129, 2013.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014.

[11] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[12] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, pages 853–899, 2013.

[13] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics, 2012.

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[15] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

[16] A. Karpathy, A. Joulin, and F. F. F. Li. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in neural information processing systems*, pages 1889–1897, 2014.

[17] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[18] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4437–4446, 2015.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[20] R. Lebret, P. O. Pinheiro, and R. Collobert. Phrase-based image captioning. *arXiv preprint arXiv:1502.03671*, 2015.

[21] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.

[22] L. Li, B. Roth, and C. Sporleder. Topic models for word sense disambiguation and token-based idiom detection. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1138–1147. Association for Computational Linguistics, 2010.

[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[24] L. Ma, Z. Lu, L. Shang, and H. Li. Multimodal convolutional neural networks for matching image and sentence. *arXiv preprint arXiv:1504.06063*, 2015.

[25] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014.

[26] J. Mao, W. Xu, Y. Yang, J. Wang, and A. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*, 2014.

[27] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *Computer Vision–ECCV 2014*, pages 720–735. Springer, 2014.

[28] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[29] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. *Proceedings of the British Machine Vision*, 1(3):6, 2015.

[30] Q. Qiu, Q. Wu, M. Bishop, R. E. Pino, and R. W. Linderman. A parallel neuromorphic text recognition system and its implementation on a heterogeneous high-performance computing cluster. *Computers, IEEE Transactions on*, 62(5):886–899, 2013.

[31] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier. Collecting image annotations using amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 139–147. Association for Computational Linguistics, 2010.

[32] Reuters picture news, 2015. pictures.reuters.com.

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[34] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *arXiv preprint arXiv:1411.4555*, 2014.

[35] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015.

[36] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, pages 487–495, 2014.