Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6654071, 8 pages
https://doi.org/10.1155/2021/6654071

Research Article

Deep Visual Semantic Embedding with Text Data Augmentation and Word Embedding Initialization

Hai He [1] **and Haibo Yang** [2]

[1]*School of Big Data and Information Industry, Chongqing City Management College, Chongqing 401331, China*
[2]*Information Center, Chongqing Medical University, Chongqing 400016, China*

Correspondence should be addressed to Haibo Yang; flypigyang@163.com

Received 2 January 2021; Revised 18 April 2021; Accepted 20 May 2021; Published 28 May 2021

Academic Editor: Pier Luigi Mazzeo

Language and vision are the two most essential parts of human intelligence for interpreting the real world around us. How to make connections between language and vision is the key point in current research. Multimodality methods like visual semantic embedding have been widely studied recently, which unify images and corresponding texts into the same feature space. Inspired by the recent development of text data augmentation and a simple but powerful technique proposed called EDA (easy data augmentation), we can expand the information with given data using EDA to improve the performance of models. In this paper, we take advantage of the text data augmentation technique and word embedding initialization for multimodality retrieval. We utilize EDA for text data augmentation, word embedding initialization for text encoder based on recurrent neural networks, and minimizing the gap between the two spaces by triplet ranking loss with hard negative mining. On two Flickr-based datasets, we achieve the same recall with only 60% of the training dataset as the normal training with full available data. Experiment results show the improvement of our proposed model; and, on all datasets in this paper (Flickr8k, Flickr30k, and MS-COCO), our model performs better on image annotation and image retrieval tasks; the experiments also demonstrate that text data augmentation is more suitable for smaller datasets, while word embedding initialization is suitable for larger ones.

1. Introduction

Language and vision are the two most essential parts of human intelligence for interpreting the real world around us and communicating with each other. Intuitively, the union of these systems will be important in the research of human intelligence and artificial intelligence. With the rapid development of machine learning (ML), especially deep learning (DL) [1], we get breakthroughs on both separate and union levels of language and vision processing. But, in the union level of language and vision processing, how to compare language and vision in a union way is still a problem. Visual semantic embedding (VSE) is proposed for tackling the problem. In the research of visual semantic, datasets usually provide an image with its corresponding description, and the given description is like a single word/phrase or a sentence. This allows us to unify the image representation [2] and word representation/embedding [3] into the same feature space. The visual semantic embedding learns a representation that allows semantically associated paired image and text into the same space; that is, visual semantic embedding learns a common feature space that represents the underlying domain structure, and their embeddings of image and text are semantically meaningful. This allows us to compare the given images and texts in a union way and achieve multimodal retrieval. But, in real-world retrieval tasks, the labelled training set is always small compared with the whole data in the system and the growing speed is much slower than the whole system in this information boomed era. How to use the limited training data to obtain a robust and efficient model becomes the challenge in VSE tasks.

In natural language processing (NLP), we face the same problem on the limitation of training data, and higher-level

tasks like sentiment classification [4, 5], stance detection [6, 7], and answer generation [8] are all heavily dependent on the size and quality of training data to obtain a reasonable performance. To tackle this problem, many achievements have been made to dig the information of given training data or introduce other pretrained common-sense models. One popular study is to generate extra data by translating sentences into French and then back to English [9]; that is, we can translate the original sentence to any other language and translate it back to its original language using a pair of machine translation model. Other works have used predictive language models to replace synonym in the data to expand the size of original data [10] and apply data noising to smooth the expanded data [11]. Previous data augmentation methods were often time-consuming [9–11]. We use the newly proposed method called EDA (easy data augmentation) [12] and the classic word embedding model, Word2Vec [3], for tackling this problem.

The main contributions are summarized as follows:

(i) Introduce text data augmentation for visual semantic embedding methods, which helps the methods obtain more information from training data and achieve better performance on image annotation and image retrieval tasks. Experiment results also show that, with text data augmentation, the model can achieve the same performance with less training data; that is, the proposed method requires less training data, which is exactly what we are trying to achieve.

(ii) Introduce the pretrained word embedding to initialize the weight of the text encoder. When training the whole model jointly, the introduced pretrained word embedding can continually improve the text representation of the given corresponding description and reduce the training time of the whole model.

2. Related Work

2.1. Visual Semantic Embedding. Traditional image annotation tasks only take the feature provided by the image itself; intuitively, this kind of methods can be widely used because of the low requirement of data. But this limits the performance of these methods; to expand the features used by the methods, visual semantic embedding has been proposed. Visual semantic embedding can embed the features of images and texts into the same space; with the help of this embedding, one can obtain a better performance in image annotation tasks. Frome et al. proposed DeViSE (deep visual-semantic embedding) [13] to perform zero-shot image classification. It uses Word2Vec [3] to represent the label words of the given image. Karpathy et al. proposed DeFrag (deep fragment embedding) [14] which uses an R–CNN (region-convolutional neural network) [15] model to extract the image features. Karpathy et al. also proposed VSA (deep visual-semantic alignments) [16] which combines R–CNN [15] and BRNN (bidirectional recurrent neural network) [17] to extract the image features and text features, respectively. Kiros et al. proposed UVSE (unifying visual-semantic embedding) [18] which uses a VGG-19 [19] network to extract image features and an LSTM (long short-term memory) [20] network to extract text features of the corresponding image. Faghri et al. proposed an extension of UVSE called VSE++ (visual-semantic embedding++) [21] which implements hard negative mining in the original UVSE. The other methods include using GRU (Gated Recurrent Unit) [22], TextCNN [23], and other methods as the text extractor to boost the performance of given tasks or introduce multimodal hashing methods to support efficient multimedia retrieval [24, 25]. Our learning framework is an extension of VSE++ with text data augmentation and triplet ranking loss with hard negative mining.

2.2. Text Data Augmentation. Previous works have proposed many techniques for data augmentation in natural language processing (NLP) topic. One popular study is to generate extra data by translating sentences into French and then back to English [9]; that is, we can translate the original sentence to any other language and translate it back to its original language using a pair of machine translation model. Other works have used predictive language models to replace synonym in the data to expand the size of original data [10] and apply data noising to smooth the expanded data [11]. Although these techniques are useful, they are not widely used in practice because they have a high cost of computational resources to obtain reasonable performance. In this work, we utilize the simple yet powerful method called easy data augmentation (EDA) [12], which is used to expand the information with given data and was proved to be efficient in text classification tasks.

2.3. Hard Negative Mining. Hard negatives are the ones that are wrongly divided into positive samples, which result in the highest loss in the training procedure. To mitigate this problem, hard negative mining (HNM) has been proposed. Hard negative mining performs the following steps. The whole samples are firstly classified by a classifier, and the classified hard negative samples are put into the negative sample set; then the classifier is continuously trained with an updated negative sample set. HNM is a well-known procedure in the context of sliding-window detectors [26] in object detection and semantic segmentation domains [15]. HNM is well studied in computer vision tasks like human detection [27] and face recognition [28, 29]. Our work is an extension of VSE++ [21]. The main contribution is that we improve the performance by text data augmentation and triplet ranking loss with hard negative mining, which are discussed in the sections titled "Text Data Augmentation" and "Triplet Ranking Loss with Hard Negative Mining."

2.4. Word Embedding. Word embedding is an important basic topic in NLP. Since computers cannot directly process natural language, word embedding can transform natural language into computer processable values. The very basic word representation method is called one-hot

representation, but this method does not include the information of a word's context. To address this problem, many context-based word embedding methods have been brought out. One of the most impressive and powerful tools called Word2Vec is proposed [3]. After Word2Vec, Glove [30] has been proposed to improve the representation quality using global word statistic information combined with context information. In 2017, Facebook proposed an efficient method called FastText [31], which brings the production level word embedding method to academics. Proposed FastText achieved good quality of vector while having a relatively fast training time. Our model uses the original Word2Vec to initialize the embedding layer of the text encoder.

3. Learning Framework

3.1. Dual-Normalized Visual Semantic Embedding Learning. In this section, we will propose a dual-normalized visual semantic embedding framework using deep neural network, as shown in Figure 1.

Our proposed framework can be separated into two parts: one part is used to extract features from the input image, and the other part is used to extract features from the corresponding text; then we use triplet ranking loss to train this network. As shown in Figure 1, the left part is the image processing part; it is a CNN-based network (in our case, we use VGG-19 architecture to extract the image feature), followed by a fully connected layer and a normalization layer. While the right part is the text processing part, it is a typical RNN-based text representation framework (in our case, we use GRU architecture for extracting text feature) with data augmentation; that is, this part contains a text augmentation layer, word embedding layer, RNN layer, and a normalization layer. After extracting features from image and text, we use similarity (inner product of two given input features) of given image and text as paired feature and train the network using triplet ranking loss with hard negative mining (details in the section titled "Triplet Ranking Loss with Hard Negative Mining"). It is worth pointing out that we are training using image and text pairs; the usage of text data augmentation can efficiently increase the number of training samples.

3.2. Text Data Augmentation. To stay consistent with EDA [12], we tested some augmentation operations that are widely used in computer vision tasks; from the experiments, we found that, by adding data augmentation, we can train more robust models. The details of text augmentation are discussed in this part. Table 1 summarizes the notations used in this paper.

For a given sentence in the training set with length l, we implement the following augmentation operations:

(i) *Synonym Replacement (SR)*. Randomly choose n words from the sentence which are not stop words. Replace each of these words with one of its synonyms chosen from WordNet [31] randomly.

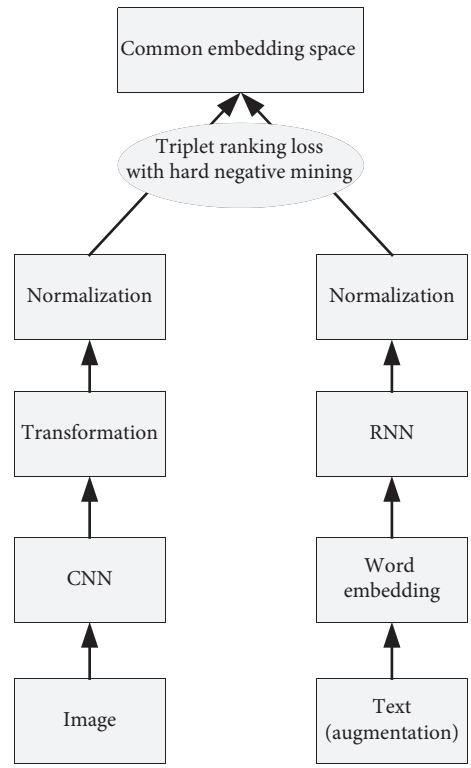


Figure 1: Proposed dual-normalized visual semantic embedding learning framework.

(ii) *Random Insertion (RI)*. Find a random synonym of a random word in the sentence which is not a stop word. Insert that synonym into a random position in the sentence. Do this n times.

(iii) *Random Swap (RS)*. Randomly choose two words in the sentence and swap their positions. Do this n times.

(iv) *Random Deletion (RD)*.Randomly remove each word in the sentence with probability p.

In the above operations, we set

$$n = \alpha l, \tag{1}$$

where α denotes the percent of words to be changed for SR, RI, and RS based on the sentence length l.

For RD operation, to be simple, we can set

$$p = \alpha. \tag{2}$$

3.3. Triplet Ranking Loss with Hard Negative Mining. Triplet loss [32] is a product of deep metric learning [33], which takes a triplet as input and lets the anchor closer to the positive one and lets the anchor away from the negative one.

In deep metric learning, the triplet loss is represented as

$$L_{\text{triplet}} = \left[S_{a,n} - S_{a,p} + \lambda\right]_+, \tag{3}$$

where $[x]_+ = \max(0, x)$, $S_{\text{a,p}}$ is the similarity of anchor x_{a} and positive input x_{p}, and $S_{\text{a,n}}$ is the similarity of anchor x_{a} and

Table 1: Notations used in this paper.

Notation	Description
l	The length of sentence
n	Number of times doing augmentation operation
p	The probability to remove every word in the sentence
α	The percent of words to be changed in the sentence
L_{triplet}	The triplet loss
L_{model}	The loss of proposed model
$S_{\text{a,p}}$	The similarity of anchor x_{a} and positive input x_{p}
x_{a}	Anchor input
x_{p}	Positive input
x_{n}	Negative input
λ	The margin that let the negative pairs away from each other
$S_{\text{i,t}}$	The similarity of image i and text t
i	Paired image
t	Paired text
$\widehat{i}$	Not paired image
$\widehat{t}$	Not paired text

negative input x_{n}. λ is the margin that lets the negative pairs away from each other.

Therefore, we define our triplet ranking loss as

$$L_{\text{model}} = \left[S_{i,\widehat{t}} - S_{i,t} + \lambda\right]_+ + \left[S_{\widehat{i},t} - S_{i,t} + \lambda\right]_+, \quad (4)$$

where i and t are the paired image and text; that is, in the given data, there exists image i with text description t. Meanwhile $\widehat{i}$ and $\widehat{t}$ are the nonpaired image and text.

To emphasise the hard negative mining, we formulate our final loss as

$$L_{\text{model}} = \max\left[S_{i,\widehat{t}} - S_{i,t} + \lambda\right]_+ + \max\left[S_{\widehat{i},t} - S_{i,t} + \lambda\right]_+, \quad (5)$$

where max denotes the concern of the hardest negative sample; only select the one with the biggest loss (i.e., hardest negative sample) as the loss of the model.

4. Experiments

Like the experiment settings in VSE++ [21], we tested image annotation task and image retrieval task in our experiments. In image annotation task, the input of this task is an image; our model needs to find out which text description is the right description for this image; and, in image retrieval task, the input becomes a description text, and find out which image best matches the corresponding text. To evaluate the results, we use Recall@K (R@K, higher is better) and Median rank (Med r, lower is better) as evaluation metrics.

4.1. Dataset. To evaluate the performance of the proposed framework, we test the methods on Flickr8k [34], Flickr30k [35], and MS-COCO [36]. The Flickr-based datasets contain 8,000 and 31,000 images, respectively, and we use 6,000 images for training, 1,000 images for validation, and 1,000 images for testing. While the MS-COCO dataset contains 329,000 images, we use 5,000 images for validation, 5,000 images for testing, and the rest of the images for training.

The dataset split method is the same as the method mentioned in [16]; the details can be seen in Table 2.

4.2. Model Settings. We set the batch size as 128 on all the datasets, using Adam [37] as the model optimizer, and set the initial learning rate as 2e-4; to better control the training learning rate, we set the learning rate as 10% for every 10 epochs (we set 30 epochs in total). For the image features extractor, we use VGG-19 architecture to extract the image features at 4096 dimensions; and, for the text features extractor, we use GRU architecture for extracting text features at 300 dimensions. To be more precise, we compare the performances with and without text data augmentation, as well as whether to initiate the embedding layer with Word2Vec. We implement the loss function of our proposed learning framework by triplet ranking loss with hard negative mining and set the margin as a fixed value of 0.2 in equation (5). We set the dimension of unified feature space as 1024 and use cosine similarity to measure the distance of image and text pairs. For the text data augmentation part, we follow the recommended usage parameters in [12] and set n at 4 and p at 0.1 in equations (1) and (2), respectively.

4.3. Performance with Text Data Augmentation. We perform experiments on Flickr8k, Flickr30k, and MS-COCO datasets. The results are depicted in Tables 3–5, respectively.

4.4. Experimental Results on Flickr8k Dataset. Table 3 shows the experimental results with and without text data augmentation on the Flickr8k dataset. "Aug" in the second column indicates the result with text data augmentation. We utilize the VSE++ [21] of our implementation as the baseline and compare the results with the models mentioned in the section titled "Dual-Normalized Visual Semantic Embedding Learning." From Table 3, we obtain 28.2% improvement on image annotation task and 20.8% on image retrieval task over baseline model, in Recall@1 metric. Also, we obtain

Table 2: Statistics and split of datasets.

Dataset	Image count		
	Train	Validation	Test
Flickr8k	6,000	1,000	1,000
Flickr30k	29,000	1,000	1,000
MS-COCO	319,000	5,000	5,000

Table 3: Experimental results with text data augmentation on Flickr8k.

Model	Feature (s)	Image count				Image retrieval			
		R@1	R@5	R@10	Med r	R@1	R@5	R@10	Med r
DeViSE [13]	Word2Vec	4.8	16.5	27.3	28	5.9	20.1	29.6	29
DeFrag [14]	R–CNN	12.6	32.9	44.0	14	9.7	29.6	42.5	15
VSA [16]	R–CNN + BRNN	16.5	40.6	54.2	7.6	11.8	32.1	44.7	12.4
UVSE [18]	ConvNet + LSTM	13.5	36.2	45.7	13	10.4	31	43.7	14
UVSE (VGG) [19]	VGG + LSTM	18.0	40.9	55.0	8	12.5	37.0	51.5	10
VSE++ [21]	VGG + GRU + HNM	16.3	37.7	52.5	9	12	33.3	48.1	11
Ours	Aug	20.9	44.1	58.8	7	14.5	39	51.2	10
Ours	Aug + Word2Vec	21.5	49.1	62.3	6	15.1	38.9	53.1	9

Table 4: Experimental results with text data augmentation on Flickr30k.

Model	Feature (s)	Image count				Image retrieval			
		R@1	R@5	R@10	Med r	R@1	R@5	R@10	Med r
DeViSE	Word2Vec	4.5	18.1	29.2	26	6.7	21.9	32.7	25
DeFrag	R–CNN	16.4	40.2	54.7	8	10.3	31.4	44.5	13
VSA	R–CNN + BRNN	22.2	48.2	61.4	4.8	15.2	37.7	50.5	9.2
UVSE	ConvNet + LSTM	14.8	39.2	50.9	10	11.8	34.0	46.3	13
UVSE (VGG)	VGG + LSTM	23.0	50.7	62.9	5	16.8	42.0	56.5	8
VSE++	VGG + GRU + HNM	29.0	54.4	66.5	4	20.3	48	59.9	6
Ours	Aug	30.6	57.9	68.5	4	21.4	49.3	61.4	6
Ours	Aug + Word2Vec	33.4	59.2	69.6	3	23.3	49.9	61.7	6

Table 5: Experimental results with text data augmentation on MS-COCO.

Model	Feature (s)	Image count				Image retrieval			
		R@1	R@5	R@10	Med r	R@1	R@5	R@10	Med r
VSA	R–CNN + BRNN	38.4	69.9	80.5	1	27.4	60.2	74.8	3
VSE++	VGG + GRU + HNM	43.6	74.8	84.6	2	33.7	68.8	81.0	3
Ours	Aug	45.1	75.8	85.3	2	33.8	67.4	80.2	3

a lower Median rank on both image annotation and image retrieval tasks. This shows the improvement of our proposed learning framework.

4.5. Experimental Results on Flickr30k Dataset. Table 4 shows the experimental results on the Flickr30k dataset. We also obtain 5.5% improvement on image annotation task and 5.4% on image retrieval task in Recall@1 metric. We notice that the results of VSE++ on the Flickr30k dataset (in Table 4) are better than those of UVSE (VGG), while they are slightly worse than the ones on the Flickr8k dataset (in Table 3). Consider the data scale of the two Flickr-based datasets (8,000 to 31,000), we can know that the VSE++ model overfits on the Flickr8k dataset. This manifests that the text data augmentation helps alleviate the overfitting problem on small datasets (the larger the dataset is, the less likely it is to overfit). Furthermore, when comparing the improved performance on the two datasets, our model gains more on Flickr8k, the smaller one, which demonstrates that the text data augmentation is more suitable for smaller datasets.

4.6. Experimental Results on MS-COCO Dataset. Table 5 shows the experimental results on the MS-COCO dataset. The result lines of VSA and VSE++ are directly obtained from their corresponding public papers [16, 21]. Compared with the VSA model, our model achieved 17.4% and 23.4% improvement in image annotation and image retrieval tasks, respectively (in Recall@1 metric). It can be seen that, except for the Median rank metric (Med r), our model exceeds the VSA model in almost all metrics. This may indicate that the VSA model applies special training skills, or it is only a

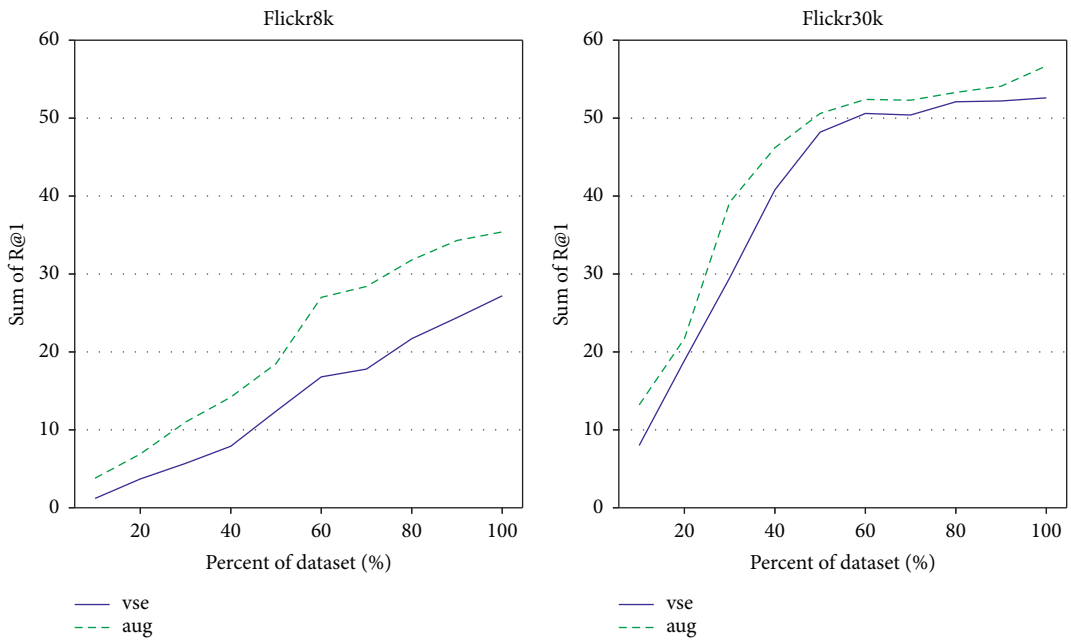
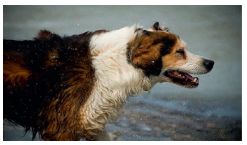


Figure 2: Performance for percent of the dataset used for training.

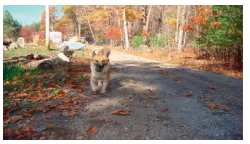
Table 6: Experimental results with word embedding initialization.

Dataset	Feature (s)	Image count				Image retrieval			
		R@1	R@5	R@10	Med r	R@1	R@5	R@10	Med r
Flickr8k	Aug	20.9	44.1	58.8	7	14.5	39.0	51.2	10
	Aug + Word2Vec	21.5	49.1	62.3	6	15.1	38.9	53.1	9
Flickr30k	Aug	30.6	57.9	68.5	4	21.4	49.3	61.4	6
	Aug + Word2Vec	33.4	59.2	69.6	3	23.3	49.9	61.7	6

GT: The kid is on a float in the snow.
SR: The kid is snow a float in the on.
RI: The kid is on a float in the snow.
RS: The kid is on vitamin a float in the snow.
RD: The kid is on a float in the snow.



GT: A brown dog shaking off water.
SR: A brown dog shaking off water.
RI: A brown hit dog shaking off water.
RS: A brown dog stirring off water.
RD: A brown dog shaking off water.



GT: A dog with a ball in its mouth running down a road covered in leaves.
SR: A dog with a ball in its mouth running down a road covered in leaves.
RI: A wiener with a ball in its mouth running downa road covered in leaves.
RS: A dog with a ball in its mouth running down a road covered in leaves.
RD: A dog with a ball in its mouth running down a road covered in leaves.

GT: A man wakeboards on choppy water.
SR: A man wakeboards on jerky water.
RI: A man wakeboards on choppy water.
RS: A man wakeboards on choppy water.
RD: A man wakeboards on choppy water.

Figure 3: Examples of easy data augmentation.

wrong record. Compared with the VSE++ model, we only get a 4.1% improvement in image annotation task (in Recall@1 metric), and the effect in image retrieval task is the same as that of the VSE++ model. This shows that the model improvement effect of text data augmentation on large datasets such as MS-COCO is limited because its text data is rich enough. According to the section titled "Performance with Word Embedding Initialization," we can choose to use word embedding initialization to further improve the learning effect of the model.

4.6.1. Performance on Different Training Set Sizes. To introduce the performance details of text data augmentation, we run both full dataset and the following training set fraction (%): {10, 20, 30, 40, 50, 60, 70, 80, 90}. Figure 2 shows the performance across the two Flickr-based datasets. The x-axis shows the percent of the whole training dataset during training, and the y-axis indicates the sum of two Recall@1 on image annotation and image retrieval tasks. The blue solid line and the green-dotted one describe the results without and with the text data augmentation, respectively. The best sum of Recall@1 without text data augmentation, 28.3% and 49.3% on Flickr8k dataset and Flickr30k dataset, respectively, is achieved using 100% of the training data. Meanwhile, with only about 60% of the available training data with text data augmentation, we surpass those two numbers. We can also infer that text data augmentation is especially helpful with smaller datasets, for the performance with text data augmentation on Flickr8k gains much more margin than the one on Flickr30k.

4.6.2. Performance with Word Embedding Initialization. Table 6 shows the performance with word embedding initialization of our learning framework. "Aug" and "Word2Vec" in the second column denote the one with text data augmentation and using word embedding method, specifically, with Word2Vec as the GRU text feature extractor, separately. We also gain an average 3.5% and 8.9% improvement in Recall@1 metric on Flickr8k and Flickr30k datasets, respectively. By comparing the numbers on the two datasets, we may infer that word embedding initialization is more suitable for larger datasets.

4.6.3. Examples of Easy Data Augmentation. Figure 3 shows the easy data augmentation in a random selected training sample. These examples illustrate that the easy data augmentation can expand the text information and provide more diverse samples.

5. Conclusion

In this paper, we introduce text data augmentation and word embedding initialization to the visual semantic embedding learning framework based on recurrent neural networks, which can unify the representation spaces of image and text into the same feature space. In the image aspect, we apply the most widely used and effective convolutional neural networks. In the text aspect, we apply the recurrent neural networks which are good at processing sequential data and utilize the word embedding models to initialize the text features extractor in the recurrent neural networks. The performance is compared with that of the model with or without text data augmentation. On the loss function part, we choose the triplet ranking loss with hard negative mining. Compared with the other models on Flickr8k, Flickr30k, and MS-COCO datasets, the experiments demonstrate that our proposed visual semantic embedding learning framework performs better in tasks such as image annotation and image retrieval. Besides, we also analyse the influences on the model of the percentage of the training set used in model learning and the word embedding initialization. The above experiments also prove that text data augmentation is more suitable for small datasets and word embedding initialization is more suitable for larger ones.

Data Availability

The experimental datasets used in this work are publicly available, and the bundled data and code of this work are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the development and application of "IOT +" maker general prototype platform project of science and technology research program of Chongqing Education Commission of China (No. KJQN201803310), the intelligent detection and location system of noncooperative targets based on hyperspectral video images project of science and technology research program of Chongqing Education Commission of China (No. KJQN201803308), and project of Research Innovation Team of Chongqing City Management College (No. KYTD202006).

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] D. Beymer and T. Poggio, "Image representations for visual learning," *Science*, vol. 272, no. 5270, pp. 1905–1909, 1996.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, https://arxiv.org/abs/1301.3781.

[4] F. Hu, L. Li, Z.-L. Zhang, J.-Y. Wang, and X.-F. Xu, "Emphasizing essential words for sentiment classification based on recurrent neural networks," *Journal of Computer Science and Technology*, vol. 32, no. 4, pp. 785–795, 2017.

[5] A. Abdi, S. M. Shamsuddin, S. Hasan, and J. Piran, "Deep learning-based sentiment classification of evaluative text based on Multi-feature fusion," *Information Processing and Management*, vol. 56, no. 4, pp. 1245–1259, 2019.

[6] N. Yu, D. Pan, M. Zhang, and G. Fu, "Stance detection in Chinese microblogs with neural networks," in *Natural Language Understanding and Intelligent Applications*, pp. 893–900, Springer, Cham, Switzerland, 2016.

[7] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi, and B.-W. On, "Fake news stance detection using deep learning architecture (CNN-LSTM)," *IEEE Access*, vol. 8, pp. 156695–156706, 2020.

[8] H. Jayakumar, M. S. Krishnakumar, V. V. V. Peddagopu, and R. Sridhar, "RNN based question answer generation and ranking for financial documents using financial NER," *Sādhanā*, vol. 45, no. 1, pp. 1–10, 2020.

[9] A. W. Yu, D. Dohan, M. T. Luong et al., "Qanet: Combining Local Convolution with Global Self-Attention for Reading Comprehension," 2018, https://arxiv.org/abs/1804.09541.

[10] S. Kobayashi, "June. Contextual augmentation: data augmentation by words with paradigmatic relations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 2, pp. 452–457, (Short Papers), New Orlean, LI, USA, June 2018.

[11] Z. Xie, S. I. Wang, J. Li et al., "Data noising as smoothing in neural network language models," 2017, https://arxiv.org/abs/1703.02573.

[12] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," 2019, https://arxiv.org/abs/1901.11196.

[13] A. Frome, G. S. Corrado, J. Shlens et al., "Devise: a deep visual-semantic embedding model," in *Advances in Neural Information Processing Systems*, pp. 2121–2129, 2013.

[14] A. Karpathy, A. Joulin, and L. F. Fei-Fei, "Deep fragment embeddings for bidirectional image sentence mapping," *Advances in Neural Information Processing Systems*, vol. 27, pp. 1889–1897, 2014.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, June 2014.

[16] 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*, pp. 3128–3137, Boston, MA, USA, June 2015.

[17] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[18] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," 2014, https://arxiv.org/abs/1411.2539.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, "Vse++: Improving visual-semantic embeddings with hard negatives," 2017, https://arxiv.org/abs/1707.05612.

[22] K. Cho, B. Van Merriënboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, https://arxiv.org/abs/1406.1078.

[23] Y. Kim, "October. Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, Doha, Qatar, October 2014.

[24] L. Zhu, X. Lu, Z. Cheng, J. Li, and H. Zhang, "Flexible multi-modal hashing for scalable multimedia retrieval," *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 2, pp. 1–20, 2020.

[25] C. Zheng, L. Zhu, X. Lu, J. Li, Z. Cheng, and H. Zhang, "Fast discrete collaborative multi-modal hashing for large-scale multimedia retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 11, pp. 2171–2184, 2019.

[26] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[27] N. Dalal and B. Triggs, "June. Histograms of oriented gradients for human detection,"vol. 1, pp. 886–893, in *Proceedings of the 2005 IEEE Computer Society Conference On Computer Vision And Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893, IEEE, San Diego, CA, USA, June 2005.

[28] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: a unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, Boston, MA, USA, June 2015.

[29] C. Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, "Sampling matters in deep embedding learning," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2840–2848, Honolulu, HI, USA, June 2017.

[30] J. Pennington, R. Socher, and C. D. Manning, "Glove: global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, October 2014.

[31] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[32] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proceedings of the International Workshop on Similarity-Based Pattern Recognition*, pp. 84–92, Springer, Copenhagan, Denmark, October 2015.

[33] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, "Multi-similarity loss with general pair weighting for deep metric learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5022–5030, Long Beach, CA, USA, June 2019.

[34] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: data, models and evaluation metrics," *Journal of Artificial Intelligence Research*, vol. 47, pp. 853–899, 2013.

[35] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: new similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 67–78, 2014.

[36] T.-Y. Lin, M. Maire, S. Belongie et al., "Microsoft COCO: common objects in context," in *European Conference on Computer Vision*, pp. 740–755, Springer, Zurich, Switzerland, September 2014.

[37] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.