Soft Computing (2019) 23:4041–4050
https://doi.org/10.1007/s00500-018-3051-y

METHODOLOGIES AND APPLICATION

An angle-based method for measuring the semantic similarity between visual and textual features

Chenwei Tang[1] · Jiancheng Lv[1] · Yao Chen[1] · Jixiang Guo[1]

Published online: 6 February 2018

Abstract
The main challenge for most image–text tasks, such as zero-shot, is the way to measure the semantic similarity between visual and textual feature vectors. The common solution is to map the image feature vectors and text feature vectors into the Hilbert space and then rank the similarity by the inner product between feature vectors. In this paper, we learn the feature representation of images and their sentence descriptions by different deep neural networks to learn about the inner-modal correspondences between visual and language data. We then use a joint embedding structure based on angle calculation for measuring the semantic similarity between visual and textual features. In the proposed method, a constant factor b keeps the similarities of positive samples and negative samples at a certain distance. Since the proposed cosine similarity method involves both normalization and vectors computation, we also develop the learning algorithm on neural networks for expressing the semantic features of texts and images. We applied the angle-based method to the challenging Caltech-UCSD Birds and the Oxford-102 Flowers datasets. The experiments demonstrate good performances on both recognition and retrieval tasks.

Keywords Semantic similarity measurement · Joint embedding structure · Angle-based method · Image–text tasks · Deep neural network

1 Introduction

When visual tasks involve text information, such as zero-shot image classification (Lampert et al. 2014; Romera-Paredes and Torr 2015), images content caption (Fang et al. 2015; Karpathy and Li 2015; Mao et al. 2015), image–sentence retrieval (Gong et al. 2014b; Wang et al. 2015), visual question answering (VQA) (Antol et al. 2015; Gao et al. 2015), the main challenge becomes to how to measure the semantic similarity between visual data and textual data (Baioletti et al. 2012). Most papers choose the inner product, which allows the angles and magnitudes of the multimodal data's feature vectors in Hilbert Space (Kempf 1994) to be measured, as the embedding function when they solve the related problems (Gong et al. 2014a; Kulis et al. 2011; Wang et al. 2015; Chen et al. 2008, 2014).

With the development of deep neural networks, the feature representations of data in any shape of form (such as images, text, audio) have achieved encouraging results. However, after the feature vectors of heterogeneous data are mapped into the Hilbert space, there is no good way to measure the distance between those vectors. As for the inner product, the larger the value, more similar the semantic representations of two vectors. The inner product allows the rigorous introduction of intuitive geometrical notions such as the length of a vector or the angle between two vectors.

In the document similarity measurement, the weight of document feature is represented by a N-dimensional vectors (D) (Goyal et al. 2015; Shum et al. 2010). The content relevance $Simlarity(D1, D2)$ of two documents is commonly measured by the cosine of angle between vectors $D1$ and $D2$, instead of the inner product which is simpler in calculation. Since the length of the vector is affected by the size of the document, the effect of the magnitude should be eliminated for higher accuracy.

Just as the document similarity measurement, in the semantic similarity metric between visual data and textual data, the magnitude of the feature vector is also affected by other environmental factors. If we only use the angle between

Communicated by V. Loia.

✉ Jiancheng Lv
lvjiancheng@scu.edu.cn

[1] Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu 610065, People's Republic of China

image feature vectors and text feature vectors to measure the semantic similarity, the method will be more effective. Moreover, the cosine of an angle is also invariant to the rescaling of the inputs (Nair and Hinton 2015) and is very efficient to evaluate distance in high dimension, especially for sparse vectors.

Over recent years, cosine similarity has demonstrated state-of-the-art performance for face verification (Nguyen and Bai 2010), speaker unsupervised adaptation (Dehak et al. 2010) as well as pattern recognition and medical diagnosis (Ye 2011). However, the cosine similarity is always used for the metric learning between data of homogeneous nature. As for the heterogeneous data, there is no attempt to measure the semantic similarity by cosine methods.

In this paper, we explore the issues of zero-shot (Reed et al. 2016; Romera-Paredes and Torr 2015) object classification and text-based retrieval. The more complex tasks we target, the fewer annotations we have, the more relevant zero-shot learning is. Zero-shot means that the intersection of categories between the training set and test set categories is empty (Larochelle et al. 2008; Palatucci et al. 2009; Liao et al. 2015). Therefore, it is necessary to establish the mapping relationship between images and visual descriptions. Different from all the above methods in terms of distance measures, the proposed method is based on calculating the angle between visual and textual feature vectors and leads to an effective metric learning method. We summarize our contributions as follows:

- We propose an angle-based embedding method for measuring the semantic similarity between visual and textual feature vectors. The method demonstrates significant improvements over the Caltech-UCSD Birds dataset (Welinder et al. 2010) and the Oxford-102 Flowers dataset (Nilsback and Zisserman 2008) in the tasks of both zero-shot images retrieval and recognition.
- In order to keep the similarity of positive samples (a pair of sample data where text and image have the same class label) and negative samples (the image and text belong to different categories) at a certain distance, the constant factor b and the selection method on the value of b are proposed.
- Due to the structure of joint embedding and angle calculation, we develop the learning algorithm on neural networks for expressing the semantic features of texts and images more accurately. It contains the derivation formula of the cosine value between two high-dimensional vectors on each a vector.

The rest of this paper is organized as follows: Sect. 2 presents the background, and the details of our method are described in Sect. 3. In Sect. 4, several experiments are conducted and analyzed. Conclusion and future work are discussed in Sect. 5.

2 Background

Over the past several years, advances in the deep neural network (Lv et al. 2010) have driven rapid progress in visual and sequential tasks. In this section, we briefly describe several previous works that our methods are built upon.

2.1 GoogLeNet

The deep convolutional networks (Nilsback and Zisserman 2008; Donahue et al. 2013; Szegedy et al. 2014) have been successfully used for visual recognition and many other tasks on large-scale benchmarks such as ImageNet (Deng et al. 2009). We use GoogLeNet, a top-performing entry of the ILSVRC-2014 classification task, for images feature extracting. GoogLeNet is based on very deep ConvNets (22 weight layers) and small convolutional filters(apart from 3×3, they also use 1×1 and 5×5 convolutions). However, in this work, the remaining challenges for the setting we study are both fine-grained and zero-shot (Reed et al. 2016). Although the target objects of the pictures are single and centered in the CUB and Flowers dataset, domain knowledge is still required to distinguish the various classes which are visually similar.

2.2 Char-CNN-RNN

Recently, recurrent networks like LSTM (Graves 1997) and convolutional recurrent components (Zhang et al. 2015) have yielded highly discriminative and generalizable text representations learned automatically from words or even characters. Similar to Reed et al. (2016), we use the character-level convolution recurrent network (Char-CNN-RNN) for extracting visual descriptions features.

In Zhang et al. (2015), character-level convolutional neural networks(ConvNets) are used for text classification. Different from image-based CNN, the text-based CNN uses temporal (1D) convolution and temporal (1D) max-pooling. After each convolution layer, rectified linear activation unit (ReLU) which is defined as $relu(x) = max(0; x)$ is used. The whole network is constructed by using convolution, pooling and threshold activation function layers. According to the fact that the character-level convolutional network lacks a strong temporal dependency along the input text sequence, Reed et al. (2016) proposed to stack a recurrent network on the top of a mid-level temporal CNN hidden layer. Benefit from both RNN and CNN, the low-level temporal features can be easily extracted. The features can be learned efficiently with fast convolutional networks, and temporal structure can still be exploited by the recurrent network. This

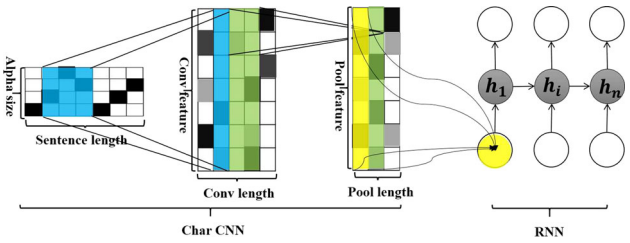

Fig. 1 Character-level convolutional recurrent net model

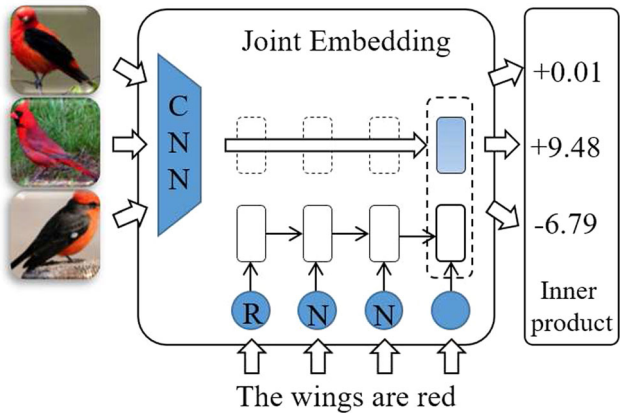

Fig. 2 Deep structured joint embedding with inner product as the compatibility function

can be viewed as modeling temporal structure at the abstract or conceptual level, not strictly delineated by word boundaries.

As Fig. 1 shows, the size of the input matrix is fixed. The RNN is stacked on the last max-pooling layer, and each column of the matrix is a moment of the RNN input. And the final encoded feature is the average hidden unit activation h_i over the number of matrix columns n, that is $\varphi(d) = \frac{1}{n}\sum_{i=1}^{n} h_i$.

2.3 DSJE with inner product

The approach of deep symmetric joint embedding (DSJE) in Reed et al. (2016) can leverage images and visual descriptions by learning a compatibility function. As Fig. 2 shows, instead of using the bilinear compatibility function, the common way is using the inner product of features generated by deep neural encoders as the matching score.

Same to Reed et al. (2016) and Akata et al. (2015a, b), given data $S = \{(i_n, d_n, l_n), n = 1, \ldots N\}$ containing images content $i_n \in I$, visual descriptions $d_n \in D$ and class labels $l_n \in L$, the goal of the deep structured joint embedding (DSJE) is to learn functions $f_i : I \to L$ and $f_d : D \to L$ by minimizing an empirical risk of the form:

$$\min_{f_i, f_d \in F} \frac{1}{N}\sum_{n=1}^{N} \Delta\left(l_n, f_i(i_n)\right) + \Delta\left(l_n, f_d(d_n)\right), \tag{1}$$

where $\Delta : L \times L \to \Re$ measures the loss incurred while incurred predicting $f_i(i)$ or $f_d(d)$ when the true label is l, and if $a = b$, $\Delta(a, b) = 0$, otherwise, $\Delta(a, b) = 1$.

Only a single model is needed for the image recognition or retrieval. The zero-shot images classifier $f_i(i; w1)$ and zero-shot text-based retriever $f_d(d; w2)$ derive a prediction by maximizing the compatibility F $F : I \times D \to \Re$ where I is the image space and D is the visual description space, over DSJE as follows:

$$f_i(i; w1) = \arg\max_{l \in L} E_{d \sim D(l)}\left[F(i, d; w1)\right], \tag{2}$$

$$f_d(d; w2) = \arg\max_{l \in L} E_{i \sim I(l)}\left[F(i, d; w2)\right]. \tag{3}$$

the class label of the visual description or image, which makes the compatibility function largest, is the result of zero-shot recognition or text-based retrieval.

The inner product of features generated by deep neural encoders is used as the compatibility function. The $\theta(i)$ and $\varphi(d)$ denote the learnable functions for images and text, respectively. The inner product is written as:

$$F(i, d) = \theta(i)^T \varphi(d). \tag{4}$$

In order to train the model, the compatibility function between a visual description and its matching images should be maximized, and the matching score of mismatching pairs should be minimized, inversely. In this work, we also train the model by this approach.

3 Method

As shown in Fig. 3, our approach measures the semantic similarity between visual and textual feature vectors by calculating the angle between feature vectors, where the image feature vectors encoded by GoogLeNet and text feature vectors encoded by the hybrid character-level convolutional recurrent (char-CNN-RNN) neural network. Although the feature extractors are different, the dimensions of visual feature vectors $\theta(i)$ and textual feature vectors $\varphi(d)$ should be the same.

During training, the input to our model is a batch of images (the class labels for each picture are different) and their corresponding visual descriptions (only one sentence for each image). We first extract the feature representation in the same dimension of images and descriptions, respectively, by GoogLeNet and char-CNN-RNN. We then treat these feature vectors as input data for the compatibility function that learns the semantic similarity between vectors.

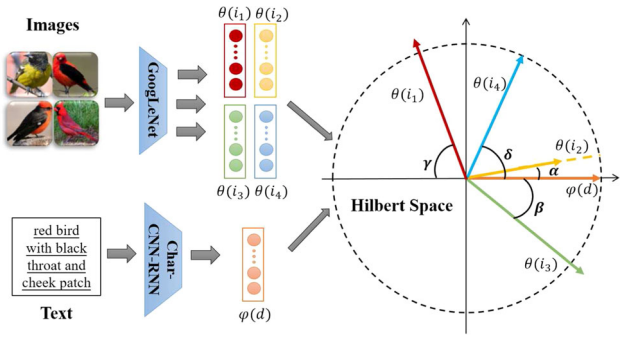

Fig. 3 Diagram for evaluating the semantic similarity of images and visual descriptions. Images are embedded with the GoogLeNet (left upper). Texts are embedded into the feature vectors of the same dimension as the image feature vectors with the char-CNN-RNN (lower left). Pairwise similarities are computed with cosine (matching score shown in angle size) in the Hilbert Space (for ease of understanding, abstract the high-dimensional space into two-dimensional planes)

After the multimodal embedding, we can get a confusion matrix (Visa et al. 2011) about positive and negative samples. The data on the diagonal from the top left to the lower right of the confusion matrix are all positive samples (the score between a corresponding image and sentence pair), and the others are all negative samples. In the asymmetric model, by minimizing the cost function (Eq. 1), each value on the diagonal of the confusion matrix can be maximized compared to the corresponding row and column.

3.1 Motivation

Different from Akata et al. (2015a, b) and Reed et al. (2016), where the semantic similarity metrics are based on the inner product, we use the cosine of the angle between visual and textual feature vectors as the compatibility function $F : I \times D \rightarrow \Re$:

$$F(i,d) = \frac{\theta(i)^T \varphi(d)}{\|\theta(i)\| \cdot \|\varphi(d)\|}, \tag{5}$$

where $\theta(i)$ denotes the image feature vector and $\varphi(d)$ denotes the visual description feature vector. Our goal is to maximize the compatibility between an image and its matching visual descriptions and minimize compatibility between mismatching pairs.

Both the inner product and cosine pay attention to the difference between two vectors in the direction instead of position. With the change of the length of the vectors, the inner product changes, too, while the cosine distance remains the same. Since the cosine distance is not sensitive to the magnitude of the vectors, it can address the problem of unified data metrics.

In general, there are three main reasons that calculating angle is more suitable for similarity measurement:

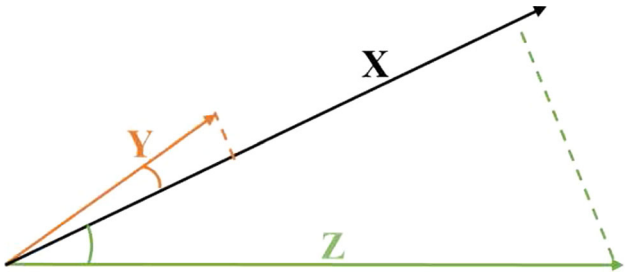

Fig. 4 Illustration of the relationship among vectors X, Y *and* Z. In a mathematical sense, X is more similar to Y because that the vector Y is closer to X in direction than Z. Therefore, we hope that $F(X,Y) > F(X,Z)$, while $Inner\_Product(X,Y) < Inner\_Product(X,Z)$, $Cosine(X,Y) > Cosine(X,Z)$

- Images and visual descriptions belong to heterogeneous data. Besides, feature vectors of images and text are extracted from two different neural networks, which means that the measurement standards are not uniform. Therefore, when calculating the angle between images and texts feature vectors, the influence of the magnitude should be eliminated.
- Compared to the inner product with no upper and lower limit, the cosine value of angle ranges from -1 to 1. Because of the boundless of cosine, the selection of the distance between compatibility of matching pairs and score of mismatching pairs can be simplified and effective. In this work, we propose a constant factor b to control the disparity, which will be detailed in following Sect. 3.2.
- It may happen that feature vector Y is closer to X on direction than Z, while Y is much shorter on the norm than Z. If we use inner product as the compatibility, $Inner\,P(X,Y) < Inner\,P(X,Z)$. While for cosine similarity, $CS(X,Y) > CS(X,Z)$, empirically the closer in the direction the vectors are similar, that is $F(X,Y) > F(X,Z)$, as shown in Fig. 4.

The strengths and weakness of cosine from different aspects are analyzed in Sect. 4.

3.2 Learning algorithm

Compared to cosine similarity, the inner product makes the concept more precise and calculation more simple. Furthermore, the inner product not only considers the direction relations between tow vectors, but also takes into account their respective magnitude. These factors should be the reasons for earlier research using the inner product as embedding functions. Through the previous analysis and later experimental result, calculating the semantic similarity after vector normalization is more accurate for two heterogeneous data in high-dimensional space. What is more, we derive the corresponding learning algorithm as follows.

The loss function $\Delta(a, b) = \{0, 1\}$ in Eq. (1) is discontinuous and non-differentiable. According to the irregular structured SVM formulation (Tsochantaridis et al. 2005), we can use a continuous and convex function as the surrogate objective function:

$$J = \frac{1}{N}\sum_{n=1}^{N}\{C_i(i_n, d_n, l_n) + C_d(i_n, d_n, l_n)\}. \tag{6}$$

For each mini-batch, a confusion matrix (Visa et al. 2011) is used to sample the training data. Each column of the confusion matrix represents the compatibility (the cosine value) between one image feature vector and all batch descriptions feature vectors and vice versa. Therefore, only the diagonal elements of the confusion matrix are the cosine of matching pairs.

The goal is to maximize the value of each diagonal element $F(i_n, d_n)$ with respect to the other cosine values both on the corresponding rows and columns. In order to increase the inter-class variety, a constant factor b which can keep a certain gap between matching scores and mismatching scores is proposed (Xie et al. 2016; Wei et al. 2015). The misclassification loss $C_i(i_n, d_n, l_n)$ takes the form:

$$C_i(i_n, d_n, l_n) = \max_{l \in L}\left\{0, \mathbb{E}_{d\sim D(l)}\left[F(i_n, d) - F(i_n, d_n)\right] + b\right\}. \tag{7}$$

And $C_d(i_n, d_n, l_n)$ means that the losses between a text feature and all image features:

$$C_d(i_n, d_n, l_n) = \max_{l \in L}\left\{0, \mathbb{E}_{i\sim I(l)}\left[F(i, d_n) - F(i_n, d_n)\right] + b\right\}. \tag{8}$$

In the equations, i_n refers to a view of a sample image from each class, and d_n refers to one of its ten corresponding visual descriptions. d and i refer to the mismatching visual descriptions and images, respectively. b refers to the constant factor.

Figure 5 gives an intuitive illustration of how the constant factor b can help with cross-view matching (Wang et al. 2015). On the left, the distance between a circle (image representation) and the triangle (visual description information) of the same color as the circle should be closest. Similarly, for an image feature vector (red straight line on the x-axis) on the right, the matching textual feature vector (another red line) should have the smaller angle than other lines of different colors (mismatching textual feature vectors). As a general view, the constant factor b balances the distance between cosine values of matching pairs (F_1) and mismatching pairs (F_2, F_3) on the left of Fig. 5. On the right, it is more intuitive that the angles between mismatching pairs (β) are larger than matching pairs' angle (α), and the gaps are controlled in the

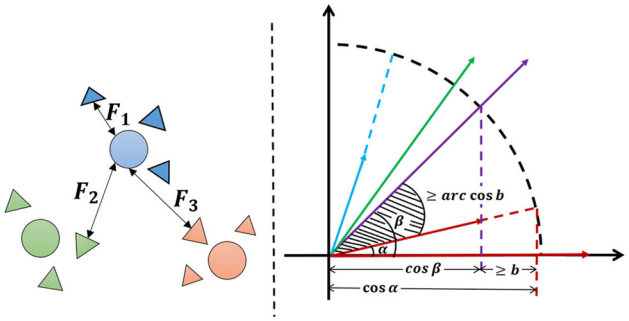

Fig. 5 Illustration of the constant factor b for visual and textual features pairs. Circles represent images, and triangles represent descriptions on the left. The red straight line on the x-axis represents image feature vector, and other color lines represent text feature vectors. The same color indicates matching images and texts (color figure online)

range of larger than $arccos(b)$. The experiment on the effect of the size of constant factor b is conducted in Sect. 4.

To train the model, a surrogate objective related to Eq. (6) J is minimized. Since the original GoogLeNet has made great progress in image encoder, the deep convolution network will not be optimized. The resulting gradients are back-propagated only through f_d by using SGD with RMSprop to learn a discriminative visual description feature extractor. The d denotes the variable of description feature vector. Therefore, the gradient as follows:

$$\begin{aligned}\delta &= \frac{\partial J}{\partial d} \\ &= \frac{1}{N}\sum_{n=1}^{N}\left\{\frac{\partial C_i(i_n, d_n, l_n)}{\partial d} + \frac{\partial C_d(i_n, d_n, l_n)}{\partial d}\right\} \\ &= \frac{1}{N}\sum_{n=1}^{N}\{\delta_1 + \delta_2\}.\end{aligned} \tag{9}$$

In consideration of the premises, the gradient contains the derivation formula of the cosine value between two high-dimensional vectors on each a vector. The images and descriptions feature vectors are defined in the form of elements: $i_n(i_1, i_2, \ldots, i_k, \ldots, i_m)$ and $d(d_1, d_2, \ldots, d_k, \ldots, d_m)$. The subscript $\{1, 2, \ldots, k, \ldots m\}$ denotes the dimensional sequence of vector i and d, where m equals to the dimension of feature vectors extracted from neural networks. Then the magnitude formulas of vectors i_n and d_n are: $|i| = \sqrt{\sum_{k=1}^{m}(i_k)^2}$ and $|d| = \sqrt{\sum_{k=1}^{m}(d_k)^2}$. Then, the derivation formula of the cosine value to the elements d_k of visual description feature vector is as follows:

$$
\begin{aligned}
\delta'_{d_k} &= \frac{\partial \frac{i^T d}{|i||d|}}{\partial d_k} \\
&= \frac{\partial \frac{\sum_{k=1}^{m} i_k d_k}{\sqrt{\sum_{k=1}^{m} i_k^2}\sqrt{\sum_{k=1}^{m} d_k^2}}}{\partial d_k} \\
&= \frac{1}{\sqrt{\sum_{k=1}^{m} i_k^2}} \frac{\partial \frac{\sum_{k=1}^{m} i_k d_k}{\sqrt{\sum_{k=1}^{m} d_k^2}}}{\partial d_k} \\
&= \frac{1}{|k|}\left(\frac{i_k}{\sqrt{\sum_{k=1}^{m} d_k^2}} - \frac{\left(\sum_{k=1}^{m} i_k d_k\right) d_k}{\left(\sum_{k=1}^{m} d_k^2\right)^{\frac{3}{2}}}\right) \\
&= \frac{1}{|k|}\left(\frac{i_k}{|d|} - \frac{\left(i^T d\right) * d_k}{|d|^3}\right). \qquad (10)
\end{aligned}
$$

Therefore, the calculation result on vector d is shown as follows:

$$
\begin{aligned}
\delta'_d &= \frac{1}{|i|}\left(\frac{i}{|d|} - \frac{\left(i^T d\right) d}{|d|^3}\right) \\
&= \frac{i}{|i||d|} - \frac{\left(i^T d\right) d}{|i||d|^3}. \qquad (11)
\end{aligned}
$$

The result can be done in the same manner, so Eqs. (12) and (13) can be worked out as follows:

$$
\delta_1 = \frac{i}{|i||d|} - \frac{\left(i^T d\right) d}{|i||d|^3}, \qquad (12)
$$

$$
\delta_2 = \left(\frac{i}{|i||d|} - \frac{\left(i^T d\right) d}{|i||d|^3}\right) - \left(\frac{i_n}{|i_n||d|} - \frac{\left(i_n^T d\right) d}{|i_n||d|^3}\right), \qquad (13)
$$

where i is a variable of image feature vector, i_n is the label image feature vector which is also known as the image of matching class for the description feature vector d, and the $|i|$ and $|d|$ are the magnitudes of image feature vector i and text feature vector d, respectively. δ is just the gradient between total cost function and the last layer of text encoder, which will be back-propagated through all layers of the character-level convolutional recurrent neural network.

Algorithm 1 summarizes the DSJE training procedure. After extracting visual and textual feature vectors (lines 5–6), the compatibility values between matching pairs and mismatching pairs are computed (line 7–8). J indicates the cost function (line 9). When matching pairs' compatibility is less than the mismatching pairs, $J > 0$ and network does not convergence. It is worth noting that we only compute the gradient of text encoder (line 10) and update $w2$ (line 11).

Algorithm 1 DSJE learning algorithm using SGD with RMSprop

1: **Input:** minibatch images i, matching visual description d, learning rate α, iteration E
2: **Initial:** weights of image and text network $w1$, $w2$
3: **repeat**
4: **for** $n = 1$ **to** $minibatch\_size$ **do**
5: $\theta(i) \leftarrow f_i(i, w1)$
6: $\varphi(d) \leftarrow f_d(d, w2)$
7: $sco\_mismat \leftarrow F(\theta(i), \varphi(d))$
8: $sco\_mat \leftarrow F(\theta(i), \varphi(d))$
9: $J \leftarrow C(\Delta, sco\_mismat, sco\_mat)$
10: $\delta \leftarrow \partial J / \partial(\varphi(d))$
11: $w2 \leftarrow w2 - \alpha\delta$
12: **end for**
13: **until** $trainingepochs$ equals to E

4 Experiment

Dataset We use the Caltech-UCSD Birds 200-2011 (CUB) and the Oxford-102 flowers (Oxford Flower) datasets in our experiments. The CUB dataset contains 11,788 bird images of 200 different categories, and the Oxford-102 Flowers dataset contains 8189 flower images of 102 different categories. Following Reed et al. (2016) and Akata et al. (2015b), we split these into class-disjoint training and test sets. The images in CUB are split into 100 training, 50 validation and 50 test categories. Once hyper-parameters have been cross-validated, the training + validation (150 categories) classes can be taken as the training set. The Oxford Flowers has 82 training + validation and 20 test classes. During mini-batch selection for training, we randomly pick a view of the image (10 views containing middle, upper left, upper right, lower left and lower right crops for the original and flipped images) and one of the ten descriptions.

Data Preprocessing For images features, we extracted 1024-dimensional pooling units from GoogLeNet (Szegedy et al. 2014) with batch normalization (Ioffe and Szegedy 2015) implemented in Torch by Reed et al. (2016). For text features, we also extracted 1024-dimensional hidden units from the hybrid of the char-CNN-RNN described in Reed et al. (2016). The alphabet's length equals to 70 consisting of all lowercase characters and punctuation, and the char-CNN input size is set to 201. Longer text inputs are cut off at this point and shorter ones are zero padded. SGD with RMSprop is used to update parameters with adaptive learning rate 0.0007 (Lv et al. 2007) and mini-batch size 40.

Task We mainly evaluate the proposed method through the task of zero-shot images recognition and text-based images retrieval on two public benchmarks. We also conduct the experiments both on the effect of the compatibility function in DSJE, the dimension of feature vectors and the value of constant factor b.

4.1 Zero-shot recognition and retrieval

For tasks of zero-shot image recognition and text-based image retrieval, we first extract description encodings from all test captions and average them per class. For the recognition task (Eq. 2), given a test image, we extract image feature vector. Then we compute the compatibility between the image feature vector and description feature vectors of all test class. The label of the visual description with maximum the compatibility maximum is the recognition result of this image. Therefore, the recognition accuracy can be obtained by calculating the proportion of the test image whose label is same to the text categories with the highest score.

For retrieval task, we compute the compatibility between all test images feature vectors and the averaged text embedding (Eq. 3). We report AP@50 (Reed et al. 2016), i.e., the percent of top-50 scoring images whose class matches that of the text query, averaged over the 50 test classes, and AP@1, i.e., the accuracy of the highest scoring images whose class matches that of the text query.

Here, we show the results of both zero-shot image recognition and text-based retrieval on the CUB dataset and compare to the previous published results in Reed et al. (2016). For the full generality and robustness to types and large vocabulary, only the char-CNN-RNN language model is compared in these tasks.

Besides the fine-grained visual descriptions in nature language, the attributes (manually encoded vectors describing shared characteristics) are also the current best side information to visual features (Lampert et al. 2014). In this task, we will also compare our results with the state-of-the-art results based on attributes.

Table 1 summarizes our results on CUB. In both the recognition (first one column) and retrieval (last two columns), cosine as the compatibility function in our method outperforms the inner product in Reed et al. (2016). Especially for the 100 classes training set, cosine performs much better than inner product consistently for both recognition and retrieval tasks.

Table 1 Zero-shot recognition and retrieval on CUB. "100" and "150" refer to the training set of training(100) categories and the training + validation (150) categories

Compatibility	Recognition (%)	Retrieval (%)	
	Top-1 ACC	AP@1	AP@50
InnerProd (100)	50.1	56.0	42.5
Cosine (100)	58.8	70.0	47.5
InnerProd (150)	54.8	60.0	45.6
Attributes (150)	50.9	–	50.0
Cosine (150)	60.2	80.0	48.3

The "InnerProd" and "Cosine" refer to using the inner product and cosine as the compatibility function of DSJE, respectively. The "Attributes" refers to using vectors embedded by attributes as the text feature vectors

Table 2 Zero-shot % recognition accuracy and retrieval average precision on the Oxford Flowers

Compatibility	Recognition (%)	Retrieval (%)	
	Top-1 ACC	AP@1	AP@50
InnerProd	63.7	70.0	57.3
Cosine	68.7	85.0	60.2

In the recognition setting, there are notable improvements. For the 100 classes training set (first two laws), cosine (58.8% Top-1 ACC, 70.0% AP@1 and 47.5% AP@50) outperforms the inner product (50.1% Top-1 ACC, 56.0% AP@1 and 42.5% AP@50) for zero-shot images recognition and text-based retrieval. Notably for both recognition and retrieval, using cosine (58.8% Top-1 ACC, 70.0% AP@1 and 47.5% AP@50) as the compatibility trained on the 100 classes (secondly law) even works better than the inner product (54.8% Top-1 ACC, 56.0% AP@1 and 45.6% AP@50) on the training + validation (150) categories. Besides, using cosine on the 150 classes for retrieval (80.0% AP@1) is higher than the inner product (60.0% AP@1) with 20% accuracy. Although the cosine is 1.7% lower than the attributes in the retrieval task, the cosine performs better than the attributes about 10% in the classification task.

To demonstrate that our results generalize beyond the case of bird images, we report the same set of experiments on Oxford-102 Flowers dataset. The experimental setting here is as the same as the previous experiments about recognition and retrieval on CUB. Table 2 summarizes our results on the Oxford Flowers.

For the zero-shot recognition (first column), using the cosine as the compatibility function achieves 68.7% Top-1 ACC compared to 63.7% with the inner product. For the text-based retrieval (last two columns), notably for AP@1, cosine (85.0%) obtains better result than the inner product (70.0%).

To demonstrate that our results generalize beyond the case of both bird images and flowers and avoid the influence of others factors, we adopt the quantitative analysis method. The experimental setting here is as the same as in Reed et al. (2016), except that the compatibility function and threshold value are different. According to the results shown in Tables 1 and 2, on both CUB and Oxford Flowers dataset, using cosine as the compatibility function is significantly better than the inner product for both zero-shot image recognition and text-based retrieval tasks.

4.2 Effect of the different hyper-parameters

Since the cosine value is neatly bounded in $[-1, 1]$, the value of constant factor b can be confirm basically. We investigate the effect of the value of constant factor c on semantic similarity measurement. The experimental setting here is as the same as in Sect. 4, except that there are no experimental results of inner product as the compatibility function.

The same value of the constant factor b is used for both image classification and text-based retrieval tasks on CUB and Oxford Flowers. As Table 3 shows, the optimal values b is chosen by a cross-validation method in which different values of $b = \{0.0009, 0.03, 0.1\}$ are tried. The factor $b = 3$ is used in our experiment, which has shown to outperform the other value for object recognition and retrieval tasks.

Figure 6a, b, c shows the method that choosing 0.03 as the constant factor b. We sampled randomly 140 matching pairs and 140 mismatching pairs from the validation set on both CUB and Flowers. Then, we calculated the cosine value, that is the matching score, for each pair and drawn the cosine value. The blue points denote score of matching pairs, and the orange points denote that of the mismatching pairs. All figures show that the values of blue points are basically higher than those of the orange points, and the gap between the average values keeps in a certain range which is larger than 0.03.

Table 3 Effect of enforced margin b

Dataset	b	Recog (%)	Retri (%)	
		Top-1 ACC	AP@1	AP@50
CUB (100)	0.009	53.5	62.0	44.4
	0.03	58.8	70.0	47.5
	0.1	55.5	62.0	43.6
CUB (150)	0.009	56.6	76.0	47.5
	0.03	60.2	80.0	48.3
	0.1	58.4	64.0	45.8
Flowers	0.009	65.5	85.0	56.4
	0.03	68.7	70.0	60.2
	0.1	67.8	65.0	59.3

Table 4 Effect of the feature vector dimensions

Dataset	d	Recog (%)	Retri (%)	
		Top-1 ACC	AP@1	AP@50
CUB (150)	128	57.1	72.0	52.6
	512	57.5	70.0	52.0
	1024	60.2	80.0	48.3
	2048	56.3	68.0	50.1
Flowers	128	63.1	70.0	59.6
	512	62.4	70.0	59.5
	1024	68.7	85.0	60.2
	2048	64.9	75.0	61.4

The d denotes the dimensions of image and visual description vectors before joint embedding

In order to measure the semantic similarity between two heterogeneous data of images and visual descriptions, the feature vectors should be mapped into the same dimension space. So we investigate the effect of the feature vector dimensions. Obviously having larger or smaller dimension is worse, but with this experiment we can see which dimensional size is best at which task. The experimental setting of char-CNN-RNN is as the same as before except the cell number of the final layer hidden units of RNN. As for the images feature extractor (pre-trained GoogLeNet), a full connection layer without activation function is added. For testing, the protocol is the same as in Tables 1 and 2.

We show the performance of several feature vector dimensions in Table 4. For CUB dataset, $d = 1024$ is competitive in zero-shot classification and when the AP1 is calculated in zero-shot retrieval. But when computing the AP50 in retrieval task, the 128 wins. As for the Flowers dataset, $d = 1024$ is also lost in the AP50 in retrieval task, but the best effect on AP@50 is achieved by 2048.

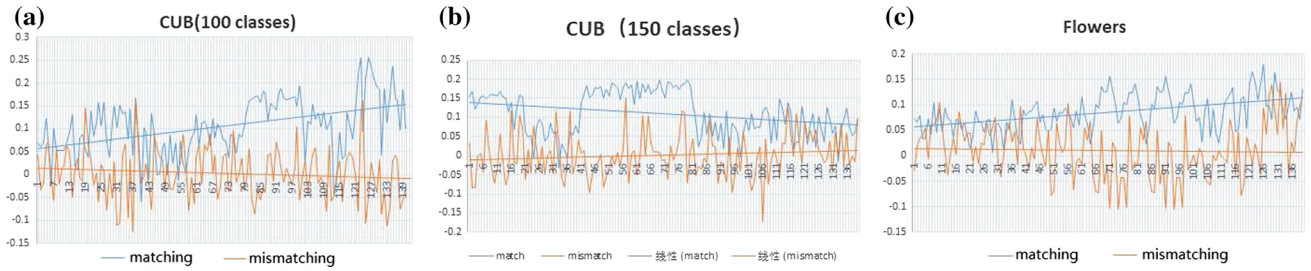

Fig. 6 Distribution of matching and mismatching cosine value. We computed the cosine value of randomized sampling-based 140 matching pairs (blue points) and 140 mismatching pairs (orange points). Results are reported on CUB (the training 100 classes and the training + validation (150) classes) and Flowers. **a** Score on CUB (100 classes), **b** score on CUB (150 classes), **c** score on Flowers (color figure online)

Fig. 7 Qualitative results on CUB and Flowers

4.3 Qualitative results

Finally, Fig. 7 shows several examples of text-based images retrieval results using a single text description on both CUB and Flowers. For every query description, the top ten retrieval images in matching score are displayed. It is worth noting that although most categories of the images are not consistent with the label of the query text, they match basically the description of the retrieved sentences. More importantly, the average precision of cosine similarity is higher than that of the inner product, especially for the Top 1.

5 Conclusion

We propose an angle-based method and develop the learning algorithm for measuring the semantic similarity between visual and textual features. With using the constant factor, the proposed method outperforms the inner product for zero-shot image recognition and text-based retrieval tasks on CUB and Oxford Flowers datasets.

We will further improve the quality of the visual description or enlarge the corpus collected from *Wikipedia* articles. Moreover, the joint embedding structure with the compatibility function of cosine has a wide range of applications such as Visual Question Answering, which we plan to explore in future work.

Acknowledgements This work is supported by National Key R&D Program of China under contract No. 2017YFB1002201 and supported by National Natural Science Fund for Distinguished Young Scholar (Grant No. 61625204) and partially supported by the State Key Program of National Science Foundation of China (Grant Nos. 61432012 and 61432014).

Compliance with ethical standards

Conflict of interest We declare that we have no conflict of interest.

References

Akata Z, Perronnin F, Harchaoui Z, Schmid C (2015a) Label-embedding for image classification. IEEE Trans Softw Eng 38(7):1425–1438

Akata Z, Reed S, Walter D, Lee H (2015b) Evaluation of output embeddings for fine-grained image classification. In: IEEE Computer Vision and Pattern Recognition, pp 2927–2936

Antol S, Agrawal A, Lu J, Mitchell M, Batra D, Zitnick CL, Parikh D (2015) VQA: visual question answering. In: IEEE International Conference on Computer Vision, pp 2425–2433

Baioletti M, Coletti G, Petturiti D (2012) Weighted attribute combinations based similarity measures. Springer, Berlin, pp 211–220

Chen CH, Lin CJ, Lin CT (2008) An efficient quantum neuro-fuzzy classifier based on fuzzy entropy and compensatory operation. Soft Comput 12(6):567–583

Chen D, Lv JC, Yi Z (2014) A local non-negative pursuit method for intrinsic manifold structure preservation. In: The 28th AAAI Conference on Artificial Intelligence (AAAI), vol 3, pp 1745–1751

Dehak N, Dehak R, Glass J, Reynolds D, Kenny P (2010) Cosine similarity scoring without score normalization techniques. In: Proceedings of Odyssey 2010—The Speaker and Language Recognition Workshop

Deng J, Dong W, Socher R, Li LJ, Li K, Li FF (2009) Imagenet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, pp 248–255

Donahue J, Jia Y, Vinyals O, Hoffman J, Zhang N, Tzeng E, Darrell T (2013) Decaf: a deep convolutional activation feature for generic visual recognition. Comput Sci 50(1):815–830

Fang H, Gupta S, Iandola F, Srivastava RK (2015) From captions to visual concepts and back. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 1473–1482

Gao H, Mao J, Zhou J, Huang Z, Wang L, Xu W (2015) Are you talking to a machine? Dataset and methods for multilingual image question answering. Computer science, pp 2296–2304

Gong Y, Ke Q, Isard M, Lazebnik S (2014a) A multi-view embedding space for modeling internet images, tags, and their semantics. Int J Comput Vis 106(2):210–233

Gong Y, Wang L, Hodosh M, Hockenmaier J, Lazebnik S (2014b) Improving image-sentence embeddings using large weakly annotated photo collections. Springer, Berlin

Goyal MM, Agrawal N, Sarma MK, Kalita NJ (2015) Comparison clustering using cosine and fuzzy set based similarity measures of text documents. Computer science

Graves A (1997) Long short-term memory. Neural Comput 9(8):1735–1780

Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. International

Conference on International Conference on Machine Learning, pp 448–456
Karpathy A, Li FF (2015) Deep visual-semantic alignments for generating image descriptions. Eprint Arxiv, pp 3128–3137
Kempf A (1994) Hilbert space representation of the minimal length uncertainty relation. Phys Rev D Part Fields 52(2):1108–1118
Kulis B, Saenko K, Darrell T (2011) What you saw is not what you get: domain adaptation using asymmetric kernel transforms. In: Computer Vision and Pattern Recognition, pp 1785–1792
Lampert CH, Nickisch H, Harmeling S (2014) Attribute-based classification for zero-shot visual object categorization. IEEE Trans Pattern Anal Mach Intell 36(3):453–465
Larochelle H, Erhan D, Bengio Y (2008) Zero-data learning of new tasks. In: Proceedings of the National Conference on Artificial Intelligence. vol 2, pp 46–651
Liao SH, Hsieh JG, Chang JY, Lin CT (2015) Training neural networks via simplified hybrid algorithm mixing Nelder–Mead and particle swarm optimization methods. Soft Comput 19(3):679–689
Lv JC, Yi Z, Tan KK (2007) Global convergence of GHA learning algorithm with nonzero-approaching learning rates. IEEE Trans Neural Netw TNN 18(6):1557–1571
Lv JC, Yi Z, Zhou J (2010) Subspace learning of neural networks, vol 42. CRC Press, Boca Raton
Mao J, Xu W, Yang Y, Wang J, Huang Z, Yuille A (2015) Deep captioning with multimodal recurrent neural networks (m-rnn). Eprint Arxiv
Nair V, Hinton GE (2015) Rectified linear units improve restricted boltzmann machines. In: Proceedings of the ICML, pp 807–814
Nguyen HV, Bai L (2010) Cosine similarity metric learning for face verification. Springer, Berlin, pp 709–720
Nilsback ME, Zisserman A (2008) Automated flower classification over a large number of classes. Computer Vision, Graphics & Image Processing, 2008. ICVGIP '08. Sixth Indian Conference on, pp 722–729
Palatucci M, Pomerleau D, Hinton GE, Mitchell TM (2009) Zero-shot learning with semantic output codes. In: Advances in neural information processing systems. International Conference on Neural Information Processing Systems, pp 1410–1418
Reed S, Akata Z, Lee H, Schiele B (2016) Learning deep representations of fine-grained visual descriptions. Computer Vision and Pattern Recognition, pp 49–58
Romera-Paredes B, Torr PHS (2015) An embarrassingly simple approach to zero-shot learning. In: International Conference on Machine Learning, pp 2152–2161
Shum S, Dehak N, Dehak R, Glass JR (2010) Unsupervised speaker adaptation based on the cosine similarity for textindependent speaker verification. In: Proceedings of Odyssey
Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2014) Going deeper with convolutions. In: Computer vision and pattern recognition, pp 1–9
Tsochantaridis I, Joachims T, Hofmann T, Altun Y (2005) Large margin methods for structured and interdependent output variables. J Mach Learn Res 6(2):1453–1484
Visa S, Ramsay B, Ralescu AL, Knaap EVD (2011) Confusion matrix-based feature selection. In: Midwest Artificial Intelligence and Cognitive Science Conference 2011, Cincinnati, Ohio, USA, April, pp 120–127
Wang L, Li Y, Lazebnik S (2015) Learning deep structure-preserving image–text embeddings. Computer Science
Wei J, Lv JC, Yi Z (2015) Robust classifier using distance-based representation with square weights. Soft Comput 19(2):507–515
Welinder P, Branson S, Mita T, Wah C, Schroff F, Belongie S, Perona P (2010) Caltech-UCSD birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology
Xie C, Lv J, Li X (2016) Finding a good initial configuration of parameters for restricted Boltzmann machine pre-training. Soft Computing, pp 1–9
Ye J (2011) Cosine similarity measures for intuitionistic fuzzy sets and their applications. Math Comput Model 53(1):91–97
Zhang X, Zhao J, Lecun Y (2015) Character-level convolutional networks for text classification. In: NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems. vol 1, pp 649–657