Video Captioning Using Hierarchical LSTM and Text-Based Sliding Window

Huanhou Xiao and Jinglun Shi(✉)

South China University of Technology, Guangzhou 510641, China
x.huanhou@mail.scut.edu.cn, shijl@scut.edu.cn

Abstract. Automatically describing video content with natural language has been attracting a lot of attention in multimedia community. However, most existing methods only use the word-level cross entropy loss to train the model, while ignoring the relationship between visual content and sentence semantics. In addition, during the decoding stage, the resulting models are used to predict one word at a time, and by feeding the generated word back as input at the next time step. Nevertheless, the other generated words are not fully exploited. As a result, the model is easy to "run off" if the last generated word is ambiguous. To tackle these issues, we propose a novel framework consisting of hierarchical long short term memory and text-based sliding window (HLSTM-TSW), which not only optimizes the model at word level, but also enhances the semantic relationship between the visual content and the entire sentence during training. Moreover, a sliding window is used to focus on k previously generated words when predicting the next word, so that our model can make use of more useful information to further improve the accuracy of forecast. Experiments on the benchmark dataset YouTube2Text demonstrate that our method which only uses single feature achieves superior or even better results than the state-of-the-art baselines for video captioning.

Keywords: Multimedia · Sentence semantics · Long short term memory · Sliding window · Video captioning

1 Introduction

With the rapid development of Internet technology, huge amounts of videos are uploaded online every day, which need to be quickly retrieved and understood. Driven by this challenge, automatically generating video caption has recently received increased interest and become an important task in computer vision. Moreover, video captioning provides the potential to bridge the semantics connection between video and language. A wide range of applications can benefit from it such as multimedia recommendation [1], assist the visually impaired [2], and human-robot interaction [3].

Before exploring the video captioning, previous work predominantly focused on describing images with natural language. Owing to the rapid development of deep learning, significant improvements have been made in image captioning. Then, researchers have extended these approaches to video. However, compared to describing images, video captioning is more challenging as the diverse information of objects, actions, and scenes.

B. Li et al. (Eds.): IoTaaS 2018, LNICST 271, pp. 57–66, 2019.
https://doi.org/10.1007/978-3-030-14657-3_6

Recently, the Long Short Term Memory (LSTM) [4] based encoder-decoder framework has been explored to generate descriptions for videos. LSTM is able to learn when to forget previous hidden states and when to update hidden states. Therefore, it can naturally deal with sequences of frames and learn long-range temporal patterns. In order to make a soft-selection over visual signals during sentence generation, attention mechanism is proposed to compute a categorical distribution of visual features, which further improve the quality of the descriptions.

Although previous encoder-decoder approaches have shown promising improvements, most of them ignore the semantic relationship between the video content and the complete sentence during training, which may cause the resulting model to generate incorrect semantics such as objects or verbs. In addition, they are trained to predict the next word given the previous ground truth word as input, while the other generated words are not holistically exploited. Therefore, the model is easy to "run off" if the last generated word is ambiguous during testing.

To tackle the above issues, we propose a Hierarchical Long Short Term Memory Model with Text-based Sliding Window (HLSTM-TSW), which utilizes an extra loss to bridge the video content and the entire sentence, as shown in Fig. 1. As a result, the relationship between visual content and sentence semantics can be explored during training. Simultaneously, a sliding window is proposed to make use of k previously generated words when predicting the next word, so that our model is able to exploit more useful information in the decoding stage. The popular video captioning dataset, Microsoft Research Video Description Corpus (YouTube2Text) [5] is used in our experiments, which demonstrates the effectiveness of the proposed method.

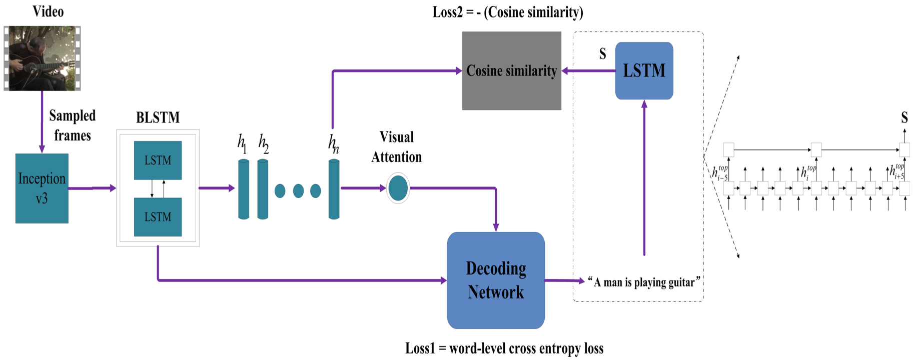

Fig. 1. The overall framework of our proposed HLSTM-TSW. Loss1 that represents the word-level cross entropy loss and Loss2 that represents the semantic relationship between video content and entire sentence are utilized together to optimize the captioning model.

2 Related Work

Early works for captioning task mainly focus on rule based systems, which detect the visual attributes (subjects, verbs, and objects) firstly, and then generate description using the template-based approach. For example, early work in [6] predicts phrases with a bilinear model and generates sentence using simple syntax statistics. However, the expansibility and richness of the natural language generated by these methods are limited by the language template.

With the rapid development of deep learning, the encoder-decoder framework has been widely applied to image captioning and video captioning. Recent works make a combination of convolutional neural network (CNN) [7] and recurrent neural network (RNN) [8] to translate the visual input to the textual output. In the case of image captioning, Vinyals et al. [9] utilize the LSTM to generate sentences with CNN features extracted from the image. Xu et al. [10] use an attention mechanism to obtain correspondences between the feature vectors and image regions. The authors of [11] propose a deep multimodal similarity model to project image features and sentences into a joint embedding space.

In video captioning, Venugopalan et al. [12] transfer knowledge from image caption models via adopting the image CNN as the encoder and LSTM as the decoder. Pan et al. [13] use the mean-pooling caption model with joint visual and sentence embedding. However, they ignore the temporal structures of video. To address this issue, Yao et al. [14] incorporate the local C3D features and a global temporal attention mechanism to select the most relevant temporal segments. Venugopalan et al. [15] present a sequence to sequence video captioning model which incorporates a stacked LSTM to read the CNN outputs firstly and then generates a sequence of words. Pan et al. [16] propose a hierarchical recurrent video encoder to exploit multiple time-scale abstraction of the temporal information.

In order to generate high-quality description for a target video, Chen et al. [17] combine the multi-modalities such as visual and audio contents to predict video topics as guidance to further improve the video captioning performance. A hierarchical structure that contains a sentence generator and a paragraph generator for language processing is introduced in h-RNN [18]. In addition, Gan et al. [19] use the Semantic Compositional Network (SCN) which extends each weight matrix of the LSTM to an ensemble of tag-dependent weight matrices to generate captions. More recently, the authors in [20] propose a multi-model stochastic RNNs network (MS-RNN) which models the uncertainty observed in the data using latent stochastic variables to improve the performance of video captioning. Song et al. [21] design an adjusted temporal attention mechanism to avoid focusing on non-visual words during caption generation. In [22], a novel encoder-decoder-reconstruction network is proposed to utilize both the forward and backward flows for video captioning.

Though the video captioning approaches mentioned above have achieved excellent results, the semantic relationship between the video content and the complete sentence is not fully exploited. Inspired by [13], in this paper, we design an extra loss to bridge the video content and sentence. Moreover, our proposed HLSTM-TSW contains a sliding window with window length of k, which enables it to focus on k previously generated words during the decoding stage.

3 Proposed Method

In this section, we introduce our approach for video captioning, as shown in Fig. 1. Firstly, the encoding stage with visual attention mechanism is presented. Then, we propose a textual attention in decoding network to calculate the contribution of words contained in the sliding window. Finally, we introduce our mixed-loss model, which simultaneously considers the context relationship between previous words and future words and the semantic relationship between visual content and entire sentence.

3.1 Encoding Network

Given a video **v** with N sampled frames, the visual features and the textual features can be represented as $v = \{v_1, v_2, \ldots, v_i, \ldots v_N\}$ and $w = \{w_1, w_2, \ldots, w_i, \ldots w_T\}$, where $v_i \in R^{D_v \times 1}$, $w_i \in R^{D_w \times 1}$, and T is the length of the sentence. Specifically, D_v and D_w are the dimension of frame-level features and the dimension of vocabulary respectively. We use a bi-directional LSTM (Bi-LSTM) which can capture both forward and backward temporal relationships to encode the visual features. The activation vectors are obtained as:

$$h_t = h_t^{(f)} + h_t^{(b)} \tag{1}$$

where $h_t^{(f)}$ and $h_t^{(b)}$ are the forward and backward hidden activation vectors.

The attention mechanism is realized by using attention weights to the hidden activation vectors throughout the input sequence, so the output context vector at time step t can be represented as:

$$a_t = \sum_{i=1}^{N} \alpha_{t,i} h_i \tag{2}$$

and

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{k=1}^{N} \exp(e_{t,k})} \tag{3}$$

$$e_{t,i} = w^T \tanh(W_a h_i + V_a h_{t-1} + b_a) \tag{4}$$

where w, W_a, V_a, b_a are learned parameters, and h_{t-1} is the hidden state of the decoder LSTM at $(t-1)$-th time step.

3.2 Decoding Network

In our decoding network, we use hierarchical LSTM to generate the description, as described in Fig. 2. During the sentence generation process, we use a sliding window to focus on k nearest generated words when predicting the next word. Following it, a

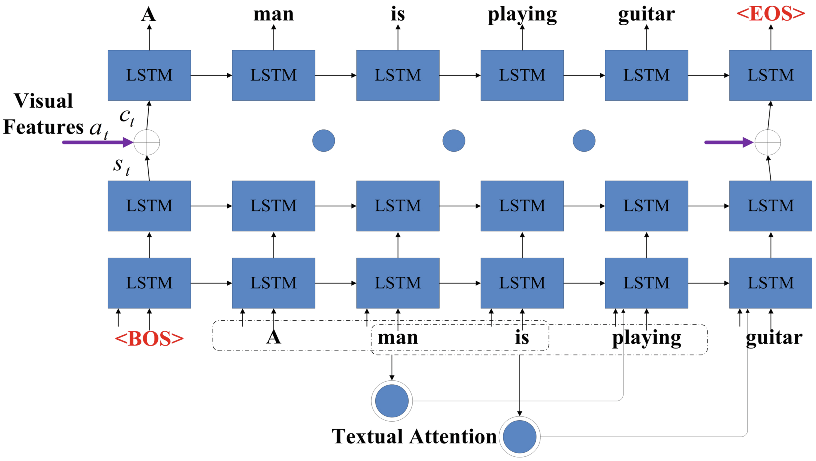

Fig. 2. The schematic diagram of our decoding network. When predicting the next word, a sliding window is utilized to focus on k nearest generated words, and the textual attention calculates their corresponding contributions.

textual attention is used to calculate the corresponding contributions of these k words. The output of it is:

$$q_t = \sum_{i=t-k}^{t-1} \beta_{t,i} w_i \tag{5}$$

and

$$\beta_{t,i} = \frac{\exp(u_{t,i})}{\sum_{m=1}^{k} \exp(u_{t,m})} \tag{6}$$

$$u_{t,i} = w_c^T \tanh(W_c w_i + V_c a_t + b_c) \tag{7}$$

where w_c, W_c, V_c, b_c are learned parameters.

Once the above operations are completed, the concatenation of w_{t-1} and q_t will be utilized as input to the bottom LSTM. Therefore, our model can focus on k previously generated words instead of only the last generated word. In addition, a visual adjusted gate is designed to avoid the problem that imposing visual attention on non-visual words, which is introduced in [21]. It can be computed as:

$$g_t = sigmoid(W_g r_t) \tag{8}$$

where W_g is learned parameter, r_t is the output of the bottom LSTM. Suppose the output of the middle LSTM is s_t, Then the input of the top LSTM is:

$$c_t = \mathrm{g}_t a_t + (1 - \mathrm{g}_t)\mathrm{s}_t \tag{9}$$

3.3 Mixed-Loss Model

According to the above analysis, at time step t, our model utilizes $\mathbf{v}$ and the previous words $w_{<t}$ to predict a word w_t with the maximal probability $P(w_t|w_{<t}, \mathbf{v})$, until we reach the end of the sentence. So the word-level cross entropy loss can be defined as:

$$loss1 = -\sum_{t=1}^{T} \log P(w_t|w_{<t}, \mathbf{v}; \theta) \tag{10}$$

where θ is the model parameter set.

To explore the semantic relationship between the visual content and the entire sentence, the last hidden activation vector h_n that represents the visual information of the video content and the sentence vector $\boldsymbol{S}$ that represents the semantic information of the entire sentence are utilized to calculate the cosine similarity, as shown in Fig. 1. In particular, $\boldsymbol{S}$ is the final output of another LSTM whose inputs are the corresponding hidden activation vectors of the top LSTM of decoding network. It is worth noting that the input at t = 1 will only flow through T/5 steps to the final output rather than T steps, which prevents the loss of information during long-distance transmissions, especially for short sentences. The cosine similarity between h_n and $\boldsymbol{S}$ can be computed as:

$$\cos(h_n, S) = \frac{h_n \bullet S}{||h_n||\,||S||} \tag{11}$$

Aiming to pull the corresponding video-sentence pairs closer in the mapping space, we define our loss2 as follow:

$$loss2 = -\cos(h_n, S) \tag{12}$$

and the final loss of our model is:

$$loss = loss1 + \delta loss2 \tag{13}$$

where δ is the tradeoff parameter.

4 Experiments

4.1 Dataset

The YouTube2Text dataset consists of 1,970 short video clips collected from YouTube, which is well suited for training and evaluating an automatic video captioning model. This dataset contains about 80,000 clip-description pairs and each clip has multiple sentence descriptions. Following [14] and [15], we split 1200 videos for training, 100 videos for validation, and 670 videos for testing.

4.2 Data Preprocessing

We convert all descriptions to lower cases, and then utilize the WordPunct function from NLTK[1] toolbox to tokenize sentences and remove punctuations. Therefore, it yields a vocabulary of 13374 in size for the dataset. In our experiments, we use the one-hot vector (1-of-N decoding, where N is the vocabulary size) to represent each word, and use the inceptionv3 [23] to extract frame-level features. In addition, we uniformly sample 60 frames from each clip.

4.3 Training Details

In our experiments, with an initial learning rate 10^{-5} to avoid the gradient explosion, we set all the LSTM unit size and the word embedding size as 512, empirically. In addition, we train our model with mini-batch 64 using ADAM optimizer [24], and the length of sentence T is set as 20. For sentence with fewer than 20 words, we pad the remaining inputs with zeros. Moreover, beam search with beam width of 5 is used to generate descriptions during testing process. To regularize the training and avoid overfitting, we apply dropout with rate of 0.5 on the outputs of LSTMs.

4.4 Metrics

We evaluate our model on the following widely-used metrics: BLEU [25], METEOR [26] and CIDEr [27], and use the Microsoft COCO evaluation server [28] to obtain our experimental results reported. BLEU is defined as the geometric mean of n-gram precision scores multiplied by a brevity penalty for short sentences. CIDEr measures the consensus between the candidate descriptions and the reference sentences. METEOR is defined as the harmonic mean of precision and recall of unigram matches between sentences.

4.5 Results and Analysis

In this subsection, we firstly explore the effect of the tradeoff parameter δ. We adjust it from 0.1 to 0.9 at intervals of 0.2. The performance curves with a different tradeoff parameter are shown in Fig. 3. We normalized METEOR and BLEU scores using the following function:

$$Q_{norm} = \frac{Q - \min(Q)}{\min(Q)} \tag{14}$$

where Q and Q_{norm} are the original and normalized performance values, respectively.

[1] [Online]. Available: https://www.nltk.org/index.html.

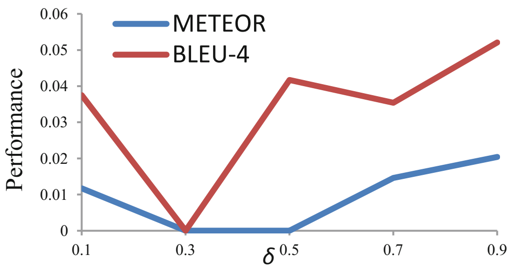

Fig. 3. The effect of δ on YouTube2Text dataset.

Table 1. Caption performance of HLSTM-TSW and other state-of-the-art methods on YouTube2Text dataset in terms of BLEU-4, METEOR, and CIDEr scores (%). HLSTM (single) represents that it was trained by cross entropy loss only, and HLSTM (mixed) represents that it was trained using mixed loss. The symbol "–" indicates such metric is unreported.

Model	BLEU-4	METEOR	CIDEr
S2VT [15]	–	29.8	–
SA [14]	41.9	29.6	51.7
h-RNN [18]	49.9	32.6	–
HRNE-SA [16]	46.7	33.9	–
hLSTMat [21]	53.0	33.6	73.8
MS-RNN [20]	**53.3**	33.8	74.8
RecNet [22]	52.3	34.1	80.3
HLSTM-TSW (single)	50.2	34.5	80.0
HLSTM-TSW (mixed)	50.5	**35.0**	**82.8**

From Fig. 3 we can see that our captioning model achieves the best performance when $\delta = 0.9$, which proves that enhancing the semantic relationship between the visual content and the entire sentence is conducive to boost the captioning model.

Then, we compare our HLSTM-TSW approach with other state-of-the-art methods, including the baseline sequence to sequence model (S2VT, MS-RNN), and the attention-based LSTM Model (SA, h-RNN, HRNE-SA, hLSTMat, RecNet).

Table 1 shows the quantitative results of the comparison. We can observe that our HLSTM-TSW performs best on METEOR and CIDEr metrics, verifying the effectiveness of our proposed method. In addition, HLSTM (mixed) performs better than HLSTM (single) on all metrics, which demonstrates that exploring the semantic relationship between video content and entire description benefits the captioning model.

Besides, some representative captions are presented in Fig. 4. Six videos are used for demonstration and two frames are extracted from each video. We notice that the sentences generated from our model are able to describe the salient contents of videos, such as woman-applying-makeup, man-shooting-gun, and monkey-pulling-dog's tail, which proves the superiority and reliability of our approach. In some of the cases, our model correctly identifies parts of the sentences, but fails to find the correct object. For example, for the top video in the right column, the generated caption is “a man is playing a piano keyboard” while the reference is “a boy is playing a keyboard”. This is due to the reason that our training data does not provide training samples to distinguish “man” and “boy”. Therefore, existing datasets for video captioning still require further refinement.

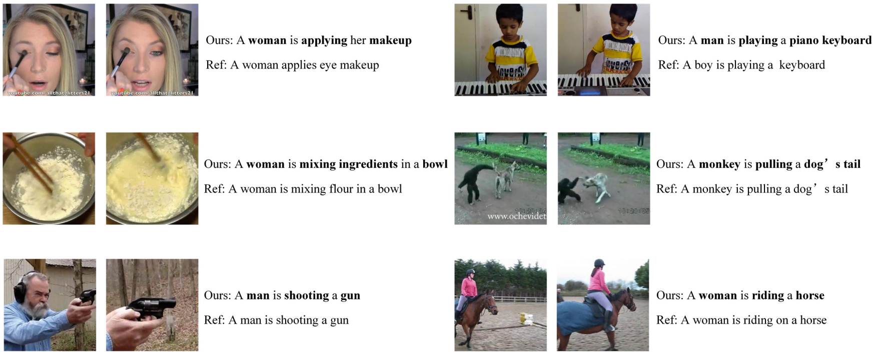
Ours: A **woman** is **applying** her **makeup**
Ref: A woman applies eye makeup
Ours: A **man** is **playing** a **piano keyboard**
Ref: A boy is playing a keyboard
Ours: A **woman** is **mixing ingredients** in a **bowl**
Ref: A woman is mixing flour in a bowl
Ours: A **monkey** is **pulling** a **dog’ s tail**
Ref: A monkey is pulling a dog’ s tail
Ours: A **man** is **shooting** a **gun**
Ref: A man is shooting a gun
Ours: A **woman** is **riding** a **horse**
Ref: A woman is riding on a horse

Fig. 4. Example results of YouTube2Text dataset.

5 Conclusion

In this paper, we propose a novel framework HLSTM-TSW to make use of the semantic relationship between video content and the entire description. In our hierarchical structure, an extra loss is utilized to map the video-sentence pairs closer in the embedding space. Moreover, the combination of the text-based sliding window and the textual attention mechanism enables the model to exploit k previously generated words instead of only the last generated word in next-word generation. Experimental results on YouTube2Text dataset show that our HLSTM-TSW achieves superior performance compared with the current start-of-the-art models. In the future work, we will combine the reinforcement learning algorithms to further improve our caption model.

References

1. Sun, L., Wang, X., Wang, Z., Zhao, H., Zhu, W.: Social-aware video recommendation for online social groups. IEEE Trans. Multimedia **19**(3), 609–618 (2017)
2. Wu, S., Wieland, J., Farivar, O., Schiller, J.: Automatic Alt-text: computer-generated image descriptions for blind users on a social network service. In: CSCW, pp. 1180–1192 (2017)
3. Das, A., et al.: Visual dialog. In: CVPR (2017)
4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
5. Chen, D.L., Dolan, W.B.: Collecting highly parallel data for paraphrase evaluation. In: ACL (2011)

6. Lebret, R., Pinheiro, P.O., Collobert, R.: Phrase-based image captioning. arXiv preprint arXiv:1502.03671 (2015)
7. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
8. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NIPS (2014)
9. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: a neural image caption generator. In: CVPR (2015)
10. Xu, K., et al.: Show, attend and tell: neural image caption generation with visual attention. In: ICML (2015)
11. Fang, H., et al.: From captions to visual concepts and back. In: CVPR (2015)
12. Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R., Saenko, K.: Translating videos to natural language using deep recurrent neural networks. In: NAACL HLT (2015)
13. Pan, Y., Mei, T., Yao, T., Li, H., Rui, Y.: Jointly modeling embedding and translation to bridge video and language. arXiv preprint arXiv:1505.01861 (2015)
14. Yao, L., et al.: Describing videos by exploiting temporal structure. In: ICCV (2015)
15. Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., Saenko, K.: Sequence to sequence - video to text. In: ICCV (2015)
16. Pan, P., Xu, Z., Yang, Y., Wu, F., Zhuang, Y.: Hierarchical recurrent neural encoder for video representation with application to captioning. arXiv preprint arXiv:1511.03476 (2015)
17. Chen, S., Chen, J., Jin, Q.: Generating video descriptions with topic guidance. In: ICMR (2017)
18. Yu, H., Wang, J., Huang, Z., Yang, Y., Xu, W.: Video paragraph captioning using hierarchical recurrent neural networks. In: CVPR (2016)
19. Gan, Z., et al.: Semantic compositional networks for visual captioning. In: CVPR (2017)
20. Song, J., Guo, Y., Gao, L., Li, X., Hanjalic, A., Shen, H.T.: From deterministic to generative: multi-modal stochastic RNNs for video captioning. arXiv preprint arXiv:1708.02478 (2017)
21. Song, J., Guo, Z., Gao, L., Liu, W., Zhang, D., Shen, H.T.: Hierarchical LSTM with adjusted temporal attention for video captioning. arXiv preprint arXiv:1706.01231 (2017)
22. Wang, B., Ma, L., Zhang, W., Liu, W.: Reconstruction network for video captioning. arXiv preprint arXiv:1803.11438 (2018)
23. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR (2016)
24. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
25. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: ACL (2002)
26. Lavie, A., Agarwal, A.: METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pp. 65–72 (2005)
27. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: CIDEr: consensus-based image description evaluation. In: CVPR (2015)
28. Chen, X., et al.: Microsoft COCO captions: data collection and evaluation server. arXiv preprint arXiv:1504.00325 (2015)