Learning to Caption Images with Two-Stream Attention and Sentence Auto-Encoder

Arushi Goel¹, Basura Fernando², Thanh-Son Nguyen², and Hakan Bilen¹
¹School of Informatics, University of Edinburgh, ²A*STAR Artificial Intelligence Initiative, Singapore

goel.arushi@gmail.com, basuraf@gmail.com, nguyen.thanh.son@scei.a-star.edu.sg, h.bilen@ed.ac.uk

Abstract

Automatically generating natural language descriptions from an image is a challenging problem in artificial intelligence that requires a good understanding of the correlations between visual and textual cues. To bridge these two modalities, state-of-the-art methods commonly use a dynamic interface between image and text, called attention, that learns to identify related image parts to estimate the next word conditioned on the previous steps. While this mechanism is effective, it fails to find the right associations between visual and textual cues when they are noisy. In this paper we propose two novel approaches to address this issue – (i) a two stream attention mechanism that can automatically discover latent categories and relate them to image regions based on the previously generated words, (ii) a regularization technique that encapsulates the syntactic and semantic structure of captions and improves the optimization of the image captioning model. Our qualitative and quantitative results demonstrate remarkable improvements on the MSCOCO dataset setting and lead to new state-of-the-art performances for image captioning.

1. Introduction

Understanding the world around us via visual representations and communicating this extracted visual information via language is one of the fundamental skills of human intelligence. The goal of recreating a similar level of intellectual ability in artificial intelligence has motivated researchers from computer vision and natural language communities to introduce the problem of automatic image captioning. Image captioning, which is to describe the content of an image in a natural language, has been an active area of research and widely applied to image and video understanding in multiple domains. An ideal model for this challenging task must have two characteristics: understanding of the image content well and generating descriptive sentences which is coherent with the image content. Many image captioning methods [33] propose various encoder-decoder models to satisfy these needs where encoder extracts an embedding from the image and decoder generates the text based on the embedding. These two parts are typically built with a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN) respectively.

One of the challenging questions in the encoder-decoder architectures is how to design the interface that controls the information flow between a CNN and RNN. While early work [33] employs a static representation for the interface such that a CNN compresses an entire image into a fixed vector and a RNN decodes the representation into natural languages sentences, this strategy is shown to perform poorly when the target sentence is long [3] and the image is reasonably cluttered [35]. Inspired from [3], Xu et al. [35] propose a powerful dynamic interface, namely attention mechanism, that identifies the relevant parts of the image embedding to estimate the next word. RNN model then predicts a word based on the context vector associated with the relevant image regions and the previously generated words. The attentional interface is shown to obtain significant performance improvements over the static one and since then it has become a key component in all state-of-the-art image captioning models [2, 28, 35, 23]. Despite this interface is substantially effective and flexible, it comes with an important shortcoming. Nevertheless visual representations that are learned by CNNs [13, 30, 15] have
been rapidly improving the state-of-the-art recognition performance in various image recognition tasks in the past few years, they can still be inaccurate when applied to noisy images and perform poorly to describe their visual content. Such noisy representations can lead to inaccurate associations between words and image regions and potentially drive a language model to poor textual descriptions. To address these shortcomings, we propose two improvements that can be used in the standard encoder-decoder based image captioning framework (Fig 1 shows how our model corrects the baseline captions).

First we propose a novel and powerful attention mechanism that can more accurately attend to relevant image regions and better cope with ambiguities between words and image regions. In particular, we introduce a two-stream attention interface that is reminiscent of [4, 17]. It automatically identifies latent categories that capture high-level semantic concepts based on visual and textual cues, as illustrated in fig 2. The two-stream attention is modeled as a neural network where each stream specializes in orthogonal tasks: the first one soft-labels each image region with the latent categories and the second one finds the most relevant region for each category. Then their predictions are combined to obtain a context vector that is passed to a decoder.

Second, inspired from sequence-to-sequence (seq2seq) machine translation methods [29, 24, 34, 12], we introduce a new regularization technique that forces the image encoder coupled with the attention block to generate a more robust context vector for the following RNN model. In particular, we design and train a novel seq2seq sentence auto-encoder model (“SAE”) that first reads in a whole sentence as input, generates a fixed dimensional vector, then the vector is further used to reconstruct the input sentence. SAE is trained to learn the structure of the input (sentence) space in an offline manner. Once it is trained, we freeze its parameters and incorporate only its decoder part (SAE-Dec) to our captioning model (“IC”) as an auxiliary decoder branch (see fig. 3). SAE-Dec is employed along with the original image captioning decoder (“IC-Dec”) to output the target sentence during training and removed in test time. We show that the proposed SAE-Dec regularizer improves the captioning performance for IC-Dec and do not bring any additional computation load in test time.

The remainder of the paper is structured as follows – section 2 discusses the related work, section 3.1 and section 3.2 introduce the proposed attentional interface and regularizer respectively, section 4 and 5 details the image captioning experiments on MSCOCO dataset [21] and discusses the results and section 6 concludes the paper.

2. Related Work

In the related work section, we first discuss the related attention mechanisms and then the use of knowledge transfer in image captioning models.

Attention mechanisms in image captioning. The pioneering work in neural machine translation [3, 25, 6] has shown that attention in encoder-decoder architectures can significantly boost the performance in sequential generation tasks. Visual attention is one of the biggest contributors to image captioning [11, 35, 2, 16]. Soft attention and hard attention variants for image captioning was introduced in [35]. Top down and bottom up self attention is effectively used in [2]. Attention over attention is used in recent work [16]. Interestingly, they use attention at both encoder and the decoder step of the captioning process. In comparison to these attention mechanisms, ours is significantly different due to three reasons. First, the traditional attention methods aims to find features or regions in an image that highly correlates with a word representation [2, 3]. In contrast, our two-stream attention uses latent class as semantic anchors to find relationship between word representations and image regions (features). Some image regions and word representations might be closer to one latent semantic class more than the others. The key difference is that both image regions and word representations predicts a scores for latent classes. We use these scores to obtain an soft attention weight to construct the context vector. Secondly, our attention mechanism uses two-stream approach to find most salient regions with respect to latent classes normalized over both latent classes and regions. Therefore, the neural structure and the attention mechanism is quite different from all prior work [35, 2, 16, 3].

Knowledge transfer in image captioning. It is well known that language consists of semantic and syntactic biases. We exploit these biases by first training a recurrent caption auto-encoder to capture those useful information using [29]. Our captioning auto-encoder is trained to reconstruct the input sentence hence decoder encapsulates structural, syntactic and semantic information of input captions. During captioning process we regularize the captioning RNN with the pretrained caption-decoder to exploit biases in the language domain. To the best of our knowledge,
no prior work has attempted such knowledge transfer in image captioning. Zhou et al. [38] encode external knowledge in the form of knowledge graphs using Concept-Net [22] to improve image captioning. The closest to ours is the work of [36] where they propose to generate scene graphs from both sentences and images and then encode the scene graphs to a common dictionary before decoding them back to sentences. However, generation of scene graphs from images itself is extremely challenging task. Besides obtaining an encoding for a graph is more challenging than obtaining a representation for sentences. Finally, we only propose to transfer syntactic and semantic information as a regularization technique during the image captioning process as an auxiliary loss. Our experiments suggest this simple idea leads to considerable improvements, specially in more structured measures such as CIDEr [31].

3. Method

We introduce our image captioning model with two-stream attention mechanism in section 3.1, a sequence-to-sequence sentence auto-encoder in section 3.2 and the integrated model in section 3.3.

3.1. Image Captioning with Two-Stream Attention

Our model takes a single raw image \( I \) and outputs a caption \( y \), as a sequence of \( y = (y_1, \ldots, y_S) \) where \( y_i \in \mathbb{R}^K \), \( K \) and \( S \) are the size of the vocabulary and length of the sentence (or caption) respectively.

We build our image captioning model on the standard encoder-decoder architecture [35] that uses a CNN as encoder and a Long Short-Term Memory (LSTM) network [14] as decoder. CNN takes an image and extracts \( R \) feature vectors \( v = \{v_1, \ldots, v_R\} \) using the previous output sequence as context where \( v_i \in \mathbb{R}^D \), \( D \) is the vision feature dimensionality. Note that each feature vector \( v_i \) is extracted from a different part of the corresponding image.

The LSTM generates one word at each time step by taking as input a context vector, the previous hidden state and the previously generated word. We depict an overview of our captioning model in fig 3(a). Our LSTM implementation closely follows the formulation in [31]. At a high-level, its dynamics can be written as

\[
\mathbf{h}^t = \text{LSTM}(E\mathbf{y}^{t-1}, \mathbf{c}^t, \mathbf{h}^{t-1}; \Theta_L)
\]  

where \( E \in \mathbb{R}^{m \times K} \) is a word embedding vector, \( m \) is the embedding dimensionality, \( h^t \in \mathbb{R}^n \) is the hidden state of the LSTM at time \( t \), \( n \) is the LSTM dimensionality, \( \Theta_L \) are the parameters of the LSTM. The vector \( \mathbf{c}^t \in \mathbb{R}^D \) is the context vector which is a linear combination of the feature vectors \( v \). \( \mathbf{c}^t \) is generated by the attention module which is explained next.

Here we introduce a novel attention mechanism that outputs \( \mathbf{c}^t \) for the given previous hidden state \( \mathbf{h}^{t-1} \):

\[
\mathbf{c}^t = f_{\text{att}}(v, \mathbf{h}^{t-1})
\]

where \( f_{\text{att}} \) is the attention function. We first review a commonly used such function, known as “soft” attention, which is proposed in [35]. It computes the context vector \( \mathbf{c}^t \) by feeding \( v \) and \( \mathbf{h}^{t-1} \) through a single layer neural network followed by a softmax function to weigh each of the \( R \) feature vectors with a relevance score:

\[
\alpha_i^t = \frac{\exp(z_i^t)}{\sum_{j=1}^R \exp(z_j^t)}
\]

where \( W_v \in \mathbb{R}^{D \times D} \) and \( W_h \in \mathbb{R}^{D \times n} \), \( w_h \in \mathbb{R}^p \) are the parameters to be learnt by the neural network and \( p \) is the dimensionality of \( w_h \). \( \alpha_i^t \) is the relevance weight for the feature vector \( v_i \). Now the context vector \( \mathbf{c}^t \) can be computed as:

\[
\mathbf{c}^t = \sum_{i=1}^R \alpha_i^t v_i
\]

Next we describe our attention mechanism as shown in Fig 3(b). Let’s assume that there exists \( C \) latent categories which capture certain semantic correlations between visual and textual features (e.g. running dog). As these latent categories are not known in prior, we propose to simultaneously identify the categories and the associations between these categories and image regions. To this end, we design a two-stream attention network that takes the previous hidden state of the LSTM \( \mathbf{h}^{t-1} \) and image features \( (v) \), and outputs a corresponding score \( \alpha_i^t \) for each \( v_i \). The two-stream network contains localization and classification streams.

Classification stream. This stream soft-assigns a latent category label to each image region for given the previous hidden state \( \mathbf{h}^{t-1} \) and \( R \) image feature vectors \( v \) by first...
mapping each \(v_i\) to a \(C\)-dimensional vector of latent category scores:
\[
q_i^t = W_{sc}v_i + W_{hc}h^{t-1}_i
\]
where parameters \(W_{sc} \in \mathbb{R}^{C \times D}\) and \(W_{hc} \in \mathbb{R}^{C \times n}\) are jointly learned along with other network parameters during training. \(q_i^t \in \mathbb{R}^C\) is latter passed through a softmax operator:
\[
\hat{q}_{ij}^t = \frac{\exp(q_{ij}^t)}{\sum_{k=1}^C \exp(q_{ik}^t)}
\]

**Localization stream.** This stream performs localization by finding the most discriminative image regions for a given category:
\[
r_i^t = W_{sl}v_i + W_{hl}h^{t-1}_i
\]
where parameters \(W_{sl} \in \mathbb{R}^{C \times D}\) and \(W_{hl} \in \mathbb{R}^{C \times n}\) are jointly learned along with other network parameters during training. \(r_i^t \in \mathbb{R}^R\) is later passed through a softmax operator:
\[
\hat{r}_{ij}^t = \frac{\exp(r_{ij}^t)}{\sum_{k=1}^R \exp(r_{ik}^t)}.
\]
Note that similar two-stream architectures can be found in \cite{jang2015recurrent, lee2015learning} where the goal is to learn detecting instances of certain object categories with image-level labels. Our attention differs in two aspects: (1) By contrast, our categories are latent and automatically discovered during training. (2) Our localization is conditioned on the previously generated words via \(h^{t-1}_i\) such that localized image regions vary at different time steps.

From the above equations, we compute \(\alpha_i^t\) by element-wise multiplying \(\hat{q}_{ij}^t\) and \(\hat{r}_{ij}^t\) and then summing over the latent classes to get the attention weights.
\[
\alpha_i^t = \sum_{j=1}^C \hat{q}_{ij}^t \cdot \hat{r}_{ij}^t.
\]
Given this \(\alpha_i^t\), we can compute the context vector as in eq. (5).

### 3.2. Sentence Auto-Encoder (SAE)

Encoder-decoder methods \cite{kim2018dual} \cite{li2018unsupervised} \cite{rao2014learning} are widely used for translating one language to another. In this framework, an encoder RNN encodes the input sequence, a sequence of vectors \(x = (x_1, ..., x_s)\) into a vector \(c\) and a decoder RNN generates a sequence of words \(y = (y_1, ..., y_T)\) in the target language, where \(x_i\) and \(y_i\) are the size of the vocabulary and length of the input and output sentence respectively. When the input and target output sentences are the same, these models function as autoencoders, by encoding an entire sentence into a vector in a low dimensional space \((e.g. 512)\), and then using it to reconstruct the input sentence. The encoder-decoder have been employed for unsupervised training in text classification \cite{mikolov2010recurrent} and machine translation \cite{bahdanau2014neural} in the past few years. In this paper, our focus is to show that auto-encoder can learn the structure of the caption space and its decoder can be further added as a regularizer to a multi-modal network for image captioning.

Our model for sentence reconstruction is a basic encoder-decoder neural network architecture \cite{rao2014learning}. The encoder is a Gated Recurrent Unit (GRU) \cite{cho2014properties} which takes as input the word at each time-step, \(x^t\) and the previous hidden state \(h^{t-1}_E\) to update \(h^t_E\):
\[
h^t_E = \text{GRU}_{E}(x^t, h^{t-1}_E; \Theta_E)
\]
where \(\text{GRU}_E\) denotes the GRU encoder, \(\Theta_E\) are the learnable parameters. We call the last hidden state of the encoder GRU as “summary vector”, as it encodes an entire sentence. We use this vector to initialize the first hidden state \(h^0_D\) of the decoder GRU. The decoder GRU predicts a word \(y^t\) at each time-step \(t\) using the previous generated word and the hidden state at that time step, \(h^t_D\):
\[
h^t_D = \text{GRU}_{D}(E_D y^{t-1}, h^{t-1}_D; \Theta_D)
\]
where \(y \in \mathbb{R}^K, E_D \in \mathbb{R}^{m \times K}\) is a word embedding matrix, \(m\) is the embedding dimensionality, \(\Theta_D\) are the learnable parameters. We train this auto-encoder model to learn reconstructing the input sequence by using the cross-entropy loss between the generated word from \(\text{GRU}_D\) and the correct word at each time step.

A stochastic variation of the vanilla sentence auto-encoders is denoising autoencoders \cite{vincent2010stacked} which are trained to both encode the input but also undo the effect of a corruption process stochastically applied to the input of the autoencoder. This not only make the encoder robust to partial

---

**Figure 4:** Overview of our proposed Image Captioning Decoder with the Sentence Auto-Encoder (SAE) - Decoder. The top half of the model shows the pre-training of our SAE. Once that model is pre-trained, we remove the SAE-decoder and add it as an auxiliary branch to our two-stream attention image captioning decoder (the bottom half). This model achieves remarkable performances both qualitatively and quantitatively.
corruption of the input but also makes the decoder invariant to the resulting noisy encoder output. To inject stochasticity, we drop each word in the input sentence with a probability of 50% to reduce the contribution of a single word on the semantics of a sentence. This helps the decoder to reconstruct input sequence even when the summary vector has less semantic information from the input. Note that we train the auto-encoder models separately from the image captioning model. We explain the training details of the model in the next section.

3.3. Learning to Caption

In this section, we explain how the decoder of the SAE is incorporated to the training of the image captioning model, as shown in fig. 4. In train time, our model takes an image as input and generate two captions by using two separate decoders. The first caption is obtained by following pipeline as explained in section 3.2: a pretrained CNN takes an image, encodes it into a feature vector and then the two-stream attention LSTM generates a sentence for the given feature vector. We denote its output as $y_t$. For generating the second caption, we use the decoder of the SAE (GRU$D$) which takes its input not from the SAE encoder (GRU$E$) but from the hidden state vector of the two-stream attention LSTM $h^*$ at time step $t$. The idea is that GRU$D$ regulates the LSTM to use the learned structure of the sentence space by the autoencoder model. We denote this caption as $y_G$. In the training phase, the loss is computed from both the output sequences and our model is optimized as follows:

$$\arg\min_\Omega \lambda L(y^*, y_t) + (1 - \lambda) L(y^*, y_G) \quad (9)$$

where $L = -\sum_{t=1}^T \log p(y_t^* | y^{1:t-1})$ is the cross-entropy loss for both to measure the word-wise difference between the generated sequences and the original input. $\lambda$ denotes a loss balancing hyperparameter. $\Omega$ are the parameters of our model that we explain in more details below.

During training of this combined objective function, we have two settings: (1) we set the parameters of the SAE decoder $\Theta_D$ to the weights of the pre-trained SAE decoder and freeze them while optimizing eq. (9) for the LSTM parameters only, i.e. $\Omega = \{\Theta_L\}$; (2) we set the parameters of the SAE decoder $\Theta_D$ to the weights of the pre-trained SAE decoder but now fine-tune them along with the LSTM parameters, i.e. $\Omega = \{\Theta_L, \Theta_D\}$. Results and experimental details are discussed in section 4.3 and section 5.

Noted that another intuitive way of exploiting the information from the pre-trained SAE model is to bring the representations from the captioning decoder closer to the encodings of the SAE encoder by minimizing the Euclidean distance between the hidden state from the SAE encoder and the hidden state from the captioning decoder at each time-step. However, in our preliminary experiments, we found this setting too restrictive on the learned hidden state of the LSTM. In the next sections, we discuss our experiments and results on various settings of our captioning model.

4. Experiments

4.1. Datasets

Microsoft COCO Dataset. Our models are evaluated on the standard MSCOCO 2014 image captioning dataset [21]. For fair comparisons, we use the same data splits for training, validation and testing as in [18] which have been used extensively in prior works. This split has 113,287 images for training with 5 captions for each image and 5k images for validation and testing respectively also with 5 captions for each image. We perform evaluation on all relevant metrics for generated sentence evaluation - CIDEr [31], Bleu [26], METEOR [10] and, ROUGE-L [20].

4.2. Implementation Details

For training our image captioning model, we compute the image features based on the Bottom-Up architecture proposed by [2], where the model is trained using a Faster-RCNN model [27] on the Visual-Genome Dataset [19] with object and attribute information. These features have $R$ regions and $D$ dimensions, where $R$ and $D$ is 36 and 2048 respectively as proposed in [2]. These 36 $\times$ 2048 image features remain consistent throughout our experiments.

4.3. Experimental Setup

Sentence Auto-Encoder. The Sentence Auto-encoder is trained offline on the MSCOCO 2014 captioning dataset [21] with the same splits as discussed in section 4.1. For the architecture, we have a single layer GRU for both the encoder and the decoder. The word embeddings are learned with the network using an embedding layer and the dimension of both the hidden state and the word embeddings is 1024. During training, the decoder is trained with teacher-forcing with a probability of 0.5. For inference, the decoder decodes till it reaches the end of caption token. The learning rate for this network is 2e-3 and it is trained using the ADAM optimizer.

Image Captioning Decoder with Sentence Auto-Encoder. In the image captioning model, the hidden state of the LSTM is fixed to 1024, the word embeddings are trained with the model with a dimension of 1024. While training the captioning model together with the SAE-decoder, we jointly learn an affine embedding layer (dimension 1024) by combining the embeddings from the image captioning decoder and the SAE-decoder. The SAE-decoder is initialized with the hidden state of the image captioning decoder. We experimentally evaluate our
models by initializing with the last hidden state and the first hidden state. Results for all the settings are discussed in section 5. During inference, we use beam search to generate captions from the captioning decoder using a beam size of 5. For training the overall objective function as given in Equation 9, the value of $\lambda$ is initialized by 0.7 and grows by a rate of 1.1 every 5 epochs until it reaches a value of 0.9. We use the ADAM optimizer with a learning rate of 2e-3. Our codes are implemented using PyTorch [1] and will be made publicly available.

5. Image Captioning Results

Two-Stream Attention. Table 1 depicts the results for the two-stream attention model that is described in section 3.1. Soft-attention is used as a baseline and corresponds to the attention mechanism in [35] with the difference that image regions are obtained from the method of [2]. We replace this attention mechanism with our proposed one and evaluate its performance for different number of latent categories ($C$). The models with latent categories 64, 128 and 256 outperform the baseline, setting $C$ to either of 128 and 256 result in significant improvements. The higher CIDEr score for 128 shows the model's capability to generate more descriptive human-like sentences. Figure 6 shows a qualitative comparison over the baseline and our attention with 128 and 256 latent categories and illustrates that our models better localize related regions (e.g. scissors and bottle) for given word context. We also observe that the model with 256 categories successfully localizes more subtle semantics such as “holding”.

As we increase the latent categories from 128 to 256, we predict more words as is evident from the higher Bleu scores, but relatively shorter sentences (lower CIDEr by a margin of 0.41%). We also observe that setting $C$ too high, 512 degrades the performance, possibly, due to overfitting to the train images. The results from our two-stream attention network helps to find the “best” latent class dimension for our Two-Stream Image Captioning Decoder with SAE-decoder shown in Figure 4.

Sentence Auto-Encoder. Figure 5 reports the reconstruction loss of Vanilla-SAE and Denoising-SAE on the test set. Given the inputs words are not corrupted, the vanilla model outperforms the denoising one in both metrics. This is expected as the denoising model is only trained with corrupted input sequences. The loss for both the Vanilla and Denoising SAE start from a relatively high value of approximately 0.8 and 0.4 respectively, and converge to a significantly low error of 0.1 and 0.2. For a better analysis, we also compute the Bleu-4 metrics on our decoded caption against the 5 ground-truth captions. As reported in table 2, both models obtain significantly high Bleu-4 scores. This indicates that an entire caption can be compressed in a low dimensional vector (1024) and can be successfully reconstructed. In the next discussions, we show that they are effective to be used as a regularizer in our image captioning model.

Image Captioning Decoder with SAE-Decoder. Table 3 reports results for our full image captioning model with SAE decoder as shown in fig. 4. As discussed in section 5.1, SAE decoder (parameters defined by $\Theta_D$) is initialized with the hidden state of the image captioning decoder. During training, we test different settings of how the SAE decoder would perform.
Table 3: Image captioning performance on the Karpathy test split of the MSCOCO 2014 caption dataset [21]. FT denotes fine-tuning ΘD. Our two-stream attention model (without SAE) performs better than the baselines in almost every metric. Our final image captioning decoder with the SAE decoder significantly improves over the baselines by 2-5% across all the metrics. These results are evaluated over the captions generated only from the image captioning decoder (by removing the auxiliary SAE decoder).

Figure 6: Illustrations of attention weights over the 36 image regions for soft attention, two-stream attention with 128 and 256 latent categories.

is trained with the image captioning decoder: (1) Vanilla vs Denoising SAE, (2) \( h^{\text{first}} \) vs \( h^{\text{last}} \), whether the SAE decoder is initialized with the first or last hidden state of the LSTM decoder, (3) Frozen vs Finetune, the parameters of GRU_D (ΘD) either get updated or not when trained with the image captioning model (in both cases the parameters are initialized with the weights of the pre-trained Vanilla or Denoising SAE decoder).
In this paper, we have introduced two novel methods for image captioning that can be used with the encoder-decoder models and improve them substantially. The first method improves over the standard soft attention mechanism by using latent categories that are automatically learned and associated with each image part with a two-stream network. The second one is a decoder network that is pretrained in an autoencoder framework to learn the structure of the caption space and is plugged into the image captioning model during training to regulate its training. We show that an image captioning model contains the proposed attention mechanism and trained with the regularizer network achieves remarkable improvement over the previous image captioning methods. As future work, we plan to further investigate potential use of label space structure learning for other challenging vision tasks.
References

[1] Pytorch. https://pytorch.org/

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6077–6086, 2018.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

[4] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2846–2854, 2016.

[5] Xinpeng Chen, Lin Ma, Wenhao Jiang, Jian Yao, and Wei Liu. Regularizing runs for caption generation by reconstructing the past with the present. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7995–8003, 2018.

[6] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 103–111, 2014.

[7] Kyunghyun Cho, Bart Van Merrienboer, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078, 2014.

[8] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, 2014.

[9] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In Advances in neural information processing systems, pages 3079–3087, 2015.

[10] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In Proceedings of the ninth workshop on statistical machine translation, pages 376–380, 2014.

[11] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1473–1482, 2015.

[12] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pages 1243–1252. JMLR. org, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4700–4708, 2017.

[16] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In The IEEE International Conference on Computer Vision (ICCV), October 2019.

[17] Vadim Kantorov, Maxime Oquab, Minsu Cho, and Ivan Laptev. Contextlocnet: Context-aware deep network models for weakly supervised localization. In European Conference on Computer Vision, pages 350–365. Springer, 2016.

[18] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3128–3137, 2015.

[19] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kananidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. International Journal of Computer Vision, 123(1):32–73, 2017.

[20] Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, page 605. Association for Computational Linguistics, 2004.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer, 2014.

[22] Hugo Liu and Push Singh. Conceptneta practical commonsense reasoning tool-kit. BT technology journal, 22(4):211–226, 2004.

[23] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 375–383, 2017.

[24] Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. arXiv preprint arXiv:1511.06114, 2015.

[25] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025, 2015.

[26] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics, pages 311–318. Association for Computational Linguistics, 2002.

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.
Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7008–7024, 2017.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Advances in neural information processing systems, pages 3104–3112, 2014.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1–9, 2015.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4566–4575, 2015.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning, pages 1096–1103. ACM, 2008.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3156–3164, 2015.

Sam Wiseman and Alexander M Rush. Sequence-to-sequence learning as beam-search optimization. arXiv preprint arXiv:1606.02960, 2016.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In International conference on machine learning, pages 2048–2057, 2015.

Xu Yang, Kailhua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 10685–10694, 2019.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. arXiv preprint arXiv:1409.2329, 2014.

Yimin Zhou, Yiwei Sun, and Vasant Honavar. Improving image captioning by leveraging knowledge graphs. In 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 283–293. IEEE, 2019.