Bootstrap, Review, Decode: Using Out-of-Domain Textual Data to Improve Image Captioning

Wenhu Chen
RWTH Aachen
hustchenwenhu@gmail.com

Aurelien Lucchi
ETH Zurich
aurelien.lucchi@inf.ethz.ch

Thomas Hofmann
ETH Zurich
thomas.hofmann@inf.ethz.ch

Abstract

We propose a novel way of using out-of-domain textual data to enhance the performance of existing image captioning systems. We evaluate this learning approach on a newly designed model that uses – and improves upon – building blocks from state-of-the-art methods. This model starts from detecting visual concepts present in an image which are then fed to a reviewer-decoder architecture with an attention mechanism. Unlike previous approaches that encode visual concepts using word embeddings, we instead suggest using regional image features which capture more intrinsic information. The main benefit of this architecture is that it synthesizes meaningful thought vectors that capture salient image properties and then applies a soft attentive decoder to decode the thought vectors and generate image captions. We evaluate our model on both Microsoft COCO and Flickr30K datasets and demonstrate that this model combined with our bootstrap learning method can largely improve performance and help the model to generate more accurate and diverse captions.

1. Introduction

Giving the ability to a machine to describe an image has been a long standing goal in computer vision. It proved to be an extremely challenging problem for which interest has been renewed in recent years thanks to recent developments brought by deep learning techniques. Among these are two techniques brought from computer vision and natural language processing, namely convolutional [13] and recurrent neural network architectures especially Long Short-term Memory Networks [9].

Among the most popular benchmark datasets for image captioning are MS-COCO and Flickr30K [4, 27] whose recent release helped accelerating new developments in the field. In 2015, a few approaches [12, 22, 34, 33, 6] set a very high standards on both datasets by reaching a BLEU-4 score of over 27% on MS-COCO and over 19% on Flickr30k. Most of the recent improvements have since then built on these systems and tried to figure out new ways to adapt the network architecture or improve the visual representation, e.g. using complex attention mechanisms [36, 35, 11].

While all these developments are significant, the performance of existing state-of-the-art approaches is still largely hindered by the lack of image-caption approaches groundtruth data. The acquisition of this training data is a painstakingly process that requires long hours of manual labor [18] and there is thus a strong interest in developing methods that require less groundtruth data. In this paper, we depart from the standard training paradigm and instead propose a novel training method that exploits large amounts of unsupervised text data without requiring the corresponding image content.

Our model is inspired by the recent success of sequence-to-sequence models in machine translation [1, 19], and multimodal recurrent structure models [34, 12, 36]. These methods encrypt an input into a common vector space from which they can be decoded to a target space. The approach proposed in [34] was among the first to use a sequence-to-sequence model for image captioning. Their model extracts visual features from a low layer of a CNN and tries to decode words based on the use of an attention mechanism. Later on, a perhaps more intuitive approach was proposed in [36] which used a fully convolutional network (FCN [28]) and multi-instance learning (MIL [24]) to detect visual concepts from the images. These concepts are then mapped to a word vector space and serve as input to an attention mechanism of an LSTM-based decoder for caption generation. We follow their idea but we use salient image region features instead of semantic word embedding. Besides we add an input reviewer - as suggested in [35] - in order to perform a given number of review steps, then output thought vectors. We feed these thought vectors to the attention mechanism of the attentive decoder. The resulting model is depicted in Figure 1 and relies on four building blocks: (i) a convolutional layer, (ii) a response map localizer, (iii) an attentive LSTM reviewer and (iv) an attentive decoder. All these steps will be detailed in Section 3.

Besides a novel architecture based on the work of [34,
35, 36], our main contribution lies in the use of out-of-domain textual data - without visual data - to bootstrap our model. We propose a novel way to artificially generate the missing visual information in order to pretrain our model. Once the model is pre-trained, we start to feed pairwise in-domain visual textual training data to fine tune the model parameters. This bootstrap learning approach yields significant improvements in terms of CIDEr [32], BLEU-4 [25] and METEOR [2]. Since unpaired textual data is largely available and easy to retrieve in various forms, our approach provides a novel paradigm to boost the performance of existing image captioning systems. This is what we will demonstrate in this paper. Besides, we also make our code and pre-training dataset available on github \footnote{https://github.com/wenhuchen/ ETHZ-Bootstrapped-Captioning} to support further progress based on our approach.

2. Related Works

Visual Concept Detector. The problem of visual concept detection has been studied in the vision community for decades. Many challenges [29, 5, 18] are related to detecting objects from a given set of detectable concepts which are mainly restricted to visible entities such as "cars" or "pedestrians". Other concepts such as "sitting", "looking" or colors are typically ignored. These concepts are however important when describing the content of an image and ignoring them can thus severely hurt the performance of an image captioning system. This problem was partially addressed by the work of [6] who proposed a weakly supervised MIL algorithm, which is able to detect broader and more abstract concepts out of the images. A similar approach was proposed in [37] to learn weakly labeled concepts out of a set of images.

Image Description Generation. Traditional methods for image captioning can be divided into two categories: (1) template-based methods such as [14] and [17], and (2) retrieval-based methods such as [15] and [16]. Template-based systems lack flexibility since the structure of the caption is fixed, the main task being to fill in the blanks of the predefined sentence. On the other hand, retrieval-based models heavily rely on the training data as new sentences can only be composed out of sentences coming from the training set. A recent breakthrough in image captioning came from the renewal of deep neural networks. Since then, a common theme has become utilizing both convolutional neural networks and recurrent neural networks for generating image descriptions. One of the early examples of this new paradigm is the work of [12] that utilizes a deep CNN to construct an image representation, which is then fed to a bidirectional Recurrent Neural Networks. Although this architecture is now a common practice in image captioning [23, 12, 21], there also exists different approaches such as the work of [6] that combines a maximum-entropy language model and a deep multimodal similarity model to generate image captions.

Another recent advance in the field of image captioning has been the use of attention models, initially proposed in [34] and quickly adopted by [11, 35] and others. These methods typically use spatially localized features computed from low layers in a CNN in order to represent fine-grained image context information while also relying on an attention mechanism that allows for salient features to dynamically come to the forefront as needed. Another related approach is [36] that extracts visual concepts (as in [6]) and uses an attentive LSTM to generate captions based on the embeddings of the detected visual concepts. The paper adopts two attention models to firstly synthesize the visual concepts and then to generate captions on the synthesized features.

Among these approaches, [36, 34, 35] are the closest to ours in spirit. Our model borrows many features from these existing systems. We make use of an input review module as suggested in [35] to encode semantic embedding into richer representation of factors. We then use a soft-attention mechanism [34] to generate attention weights for each factor, and we finally use beam search to generate a caption out of the decoder.

Leveraging External Training Data. Most image captioning approaches are trained using paired image-caption data. Given that producing such data is an expensive process, there has been some interest in the community to train models with fewer data. The approach developed in [22] allows the model to enlarge its word dictionary to describe the novel concepts using a few examples and without extensive retraining. However this approach still relies on paired data. The approach closest to ours is [8] that focuses on transferring knowledge from a model trained on external unpaired data through weight sharing or weight composition. Due to the architecture of our model, we can simply "fake" visual concepts from out-of-domain textual corpus and pre-train our model on the faked concept-caption pairwise data.

Evaluation Metrics. Automatic evaluation is critical to measure the image caption quality, however the evaluation has proven to be challenging. There exists different metrics to measure several desirable properties such as grammar, saliency and correctness. BLEU [25] is a precision-based machine translation evaluation metric, METEOR [2] performs well in corresponding to human subjects. CIDEr [32] has emerged as a new method to measure human consensus on generated captions.
3. Model

Our captioning system consists of the following steps. Given an input image we first use a Convolutional Neural Network to detect salient visual words in the image. The extracted visual features are then fed to a reviewer, which reviews all the information and learns thought vectors. These vectors along with the groundtruth captions are then used to train a soft attentive decoder similar to the one proposed in [34].

3.1. Visual Concept Detector

The first step in our approach is to detect salient visual concepts in an image. We follow the approach proposed in [6] and formulate this task as a multi-label classification problem. The set of output classes is defined by a dictionary $\mathcal{V}$ consisting of the 1000 most frequent words in the training captions, from which the most frequent 15 stop words were discarded. The set $\mathcal{V}$ covers around 92% of the word occurrences in the training data.

As pointed out in [6], a naive approach to image captioning is to encode full images as feature vectors and use a multilayer perceptron to perform classification. However, most concepts are region-specific and [6] demonstrated superior performance by applying a CNN to image regions and integrating the information using a weakly-supervised variant of the Multiple Instance Learning (MIL) framework originally introduced in [24]. We use this approach as the first step in our framework, we model the probability of a visual word $v_c \in \mathbb{R}^{\mathcal{V}}$ appearing in the image as a product of probabilities defined over a set of regions $\{b_j\}$. Formally, we define this probability as

$$p(v_c) = 1 - \prod_{b_j} (1 - \text{sigmoid}(W^t_{v_c} \phi(b_j) + u_w))$$  \hspace{1cm} (1)$$

where $j \in \{1, 2, \cdots, 14 \times 14\}$ indexes the image region from the response map $R_M(I) \in \mathbb{R}^{14 \times 14 \times A}$, $\phi(b_j) \in \mathbb{R}^A$ denotes the CNN features extracted over the region $b_j$, and $W^t_{v_c} \in \mathbb{R}^{\mathcal{V} \times A}$ and $u_w \in \mathbb{R}^{\mathcal{V}}$ are the parameters of the CNN learned from data. Our concept detector architecture is taken from [6], which is trained with maximum-likelihood on image-concept pairwise data extracted from MS-COCO.

We experiment with two approaches to encode the visual concepts, further details are given in the next section. Note that the visual concept detector is trained only on MS-COCO and we use the same model on Flickr30k.

3.2. Salient Regional Feature

A standard approach (see e.g. [36]) to feed information about the image context to the input reviewer is by taking
The semantic word embeddings [26] of the detected visual concepts. The resulting word vectors are more compact than one-hot encoded vectors and capture many useful semantic concepts such as gender, location, comparative degrees, …

In this paper, we also experiment with a different approach where we take visual features from the image sub-regions \( B^\tau \) that have the strongest connections with the set of detected visual concepts \( v_c^\tau \). For each of the top \( T \) concepts in \( \{v_c^\tau\}_{\tau=1}^T \), we compute image sub-regions \( \{B^\tau\}_{\tau=1}^T \) as

\[
B^\tau = \arg \max_{b_j} W_{v_c^\tau}^b \phi(b_j) \tag{2}
\]

In summary, we extract salient features \( \{v_c^\tau\}_{\tau=1}^T \) from an image either in a semantic manner or in a visual manner as follows:

\[
v_c = \begin{cases} 
E v_c^\tau & \text{semantic} \\
\phi(B^\tau) & \text{visual} 
\end{cases} \tag{3}
\]

where \( E \) is an embedding matrix that maps a one-hot encoded vector \( v_c \in \mathbb{R}^{|V|} \) to a more compact embedding space \( v_c \in \mathbb{R}^d \), and \( \phi(B^\tau) \in \mathbb{R}^A \) corresponds to the CNN features from image region \( B^\tau \). Note that \( v_c \) has a different dimension in the two cases, as semantic feature \( d = 300 \) while as visual feature \( d = A = 2048 \).

As demonstrated in Section 4, we found that using CNN regional features is advantageous over semantic word features. Our conjecture is that visual features are often more expressive since one image region can relate to multiple word choices.

3.3. Input Reviewer

As depicted in Figure 2, most of the detected visual concepts tend to capture very salient image concepts. However, some concepts are duplicated and others are incorrect such as “intersection” in the example provided in Figure 2. We address this problem by applying an input reviewer [35] to synthesize ”thought vectors” that capture globally consistent concepts and eliminate mistakes introduced by the visual concept detector. Note that unlike the approach described in [35] that takes serialized CNN features for the whole image as input, we instead use the features described in Section 3.2. Since these features already have a strong semantic meaning, we did not apply the “weight typing” and ’discriminative supervision” strategies suggested in [35].

Our input reviewer is composed of an attentive LSTM, which estimates an attention weight \( \beta_{c,\tau} \) for a given \( v_c \) and outputs its hidden state as though vectors \( f_t \in \mathbb{R}^F \). Formally,

\[
\beta_{c,\tau} = \frac{\exp(g_c(f_{t-1}, v_{c,\tau}, v^\tau_1))}{\sum_{\tau} \exp(g_c(f_{t-1}, v_{c,\tau}, v^\tau_1))}, \tag{4}
\]

where \( v_{c,\tau} \) is the overview context vector from the last step. We use an attention function \( g_c \) with parameters \( W^c \), defined as

\[
g_c(f_{t-1}, v_{c,\tau}, v^\tau_1) = W^c_a v^\tau_1 \tanh(v^\tau_1 + W^T_f f_{t-1} \beta_{c,\tau} v^\tau_1) \tag{5}
\]

where \( W^c_a \in \mathbb{R}^{d \times (d + F)} \) and \( W^T_f \in \mathbb{R}^d \) are parameters learned from data, we set \( F = 300 \) in our experiment.

In order to select which visual concepts to focus on, we could sample with regard to the attention weights \( \beta \), but a simpler approach described in [1] is to take a weighted sum over all inputs, i.e.

\[
v^\tau_t = \sum_{\tau} \beta_{c,\tau} v^\tau_1 \tag{6}
\]

As shown on Figure 3, our LSTM reviewer uses \( v^\tau_t \) and \( f_{t-1} \) as inputs to produce the next thought vector \( f_t \). Unlike the LSTM decoder presented in the next section, it does not
rely on the input symbols \( \{x_i\} \). The reviewer LSTM functions basically as a text synthesizer without any reliance on visual contexts, which explains why we can pre-train this part using only textual data (see Section 3.6).

![LSTM model for the Input Reviewer.](image)

Similarly to the reviewer, we use a weighted sum \( f'_t \) over all thought vectors to approximate sampling

\[
 f'_t = \sum_{i} \alpha_{i,t} f_t.
\]

Unlike the input reviewer whose initial state is set to zero, we introduce visual information in the decoder by initializing the LSTM memory \( c_0 \) and state \( h_0 \) with a linear transformation of CNN features, i.e.

\[
 h_0 = W_h \psi(I) \quad \text{and} \quad c_0 = W_m \psi(I),
\]

where \( \psi(I) \) denotes the CNN features of the image, \( W_h \in \mathbb{R}^{H \times |\psi(I)|} \) and \( W_m \in \mathbb{R}^{H \times |\psi(I)|} \) are parameters learned from data. Note that \( \psi(I) \) is different from \( \phi(B^r) \) in Equation 1 in that \( \psi(I) \) extracts full image features from the upper layer, while \( \phi(B^r) \) extracts sub-region features from the response map.

### 3.5. Model Learning

The output state \( h_t \) of the decoder LSTM contains all the useful information for predicting next word \( x_t \). We follow the implementation of [34] and add a two-layer perceptron with dropout mechanism on top of the decoder LSTM to predict the distribution for all words in the vocabulary. We calculate the cross-entropy loss based on the proposed distribution \( p(x_i) \) and groundtruth word \( y_i \).

We train our model using maximum likelihood with a regularization term on the attention weights \( \alpha \) and \( \beta \) of the input reviewer and attentive decoder. Formally, we write

\[
 Loss = -\min_{\theta} \sum_i \log p(y_i) + \lambda (g(\alpha) + g(\beta))
\]

\[
 g(\alpha) = \sum_{i} (1 - \sum_{i} \alpha_{i,t})^2,
\]

where \( y_i \) is the groundtruth word, \( \theta \) refers to all model parameters and \( \lambda > 0 \) is a balancing factor between the cross-entropy loss and a penalty on the attention weights. We use the penalty function \( g \) described in [34] to ensure every concept and thought vector receives enough attention.

### 3.6. Bootstrap Learning

Most existing approaches to image captioning rely on pairs of images and corresponding captions for learning their model parameters. Such training data is typically expensive to produce and usually requires using crowd-sourcing techniques. The MS-COCO dataset was for instance annotated using Amazon’s Mechanical Turk, a process that required 70K worker hours [18]. In contrast, unpaired text and image data is abundant and cheap to obtain but can not be used as is with current image captioning.
We here suggest a novel approach to exploit text data without corresponding images to bootstrap our model. Since images are used as inputs to the visual concept detector to generate visual concepts or salient regional features, we need to "fake" these inputs during the bootstrap training phase.

Fake Semantic Embeddings In the case where the salient features \( \{ v^\tau \}^{\tau=1}_{\tau=T} \) described in Section 3.2 are based on semantic embeddings, we can directly fake these concepts based on the groundtruth sentences. This process is illustrated in Figure 6. We experiment with two methods named "Truth Generator" and "Noisy Generator". The "Truth Generator" approach takes sample words from sentences longer than 15 words or zero-pad shorter sentences to generate 15 concepts. The "Noisy Generator" mixes words sampled from the groundtruth sentences with randomly sampled words to form 15 concepts. Besides, we also experimented with out-of-domain text data with different sizes, i.e. 600K and 1.1M captions, which roughly corresponds to the number of training captions in MS-COCO.

Fake Regional Features The case of using salient regional features is more difficult to handle since our additional training data only consists of textual data without corresponding images. We propose to address this problem by relying on the strong correlation between visual embeddings and regional features. Specifically we construct a new regional feature for a given concept \( v^\tau \) by aggregating all the regional features of this concept in the training data, i.e.

\[
\bar{\phi}_{v^\tau} = \mathbb{E}_{B^{\tau^\prime}=\tau^\prime} [\phi(B^{\tau^\prime})]
\]

(13)

where \( \bar{\phi} \in \mathbb{R}^{2048} \). We visualized these “faked” regional features using t-SNE [20] and the results shown in Figure 5 demonstrate that the aggregated regional features capture similar properties to the ones of the semantic embeddings.

Pre-training This bootstrap learning strategy results in a two-step training procedure. The first step is to pre-train our model on unpaired textual data, which teaches the model to produce human-like captions based on out-of-domain language samples. Note that nearly all the parameters can be pre-trained, except the ones related to the visual features (i.e. parameters related to \( v'_t, v_\tau, h_0, c_0 \)). In the second phase of training, we adapt the model to in-domain samples for which pairwise text and visual data is available. As can be seen from Figure 7, after only one epoch of adaptation, the performance already reaches a promising stage.

4. Experiments

We first describe the datasets used in our experiments as well as the experimental methodology and we then present and discuss the evaluation results.

4.1. Data

We evaluate the performance of our model on the popular MS-COCO [18] and Flickr30K [27] datasets. The MS-COCO dataset contains 123,287 images for training and validation and 40,775 images for testing, while Flickr30K provides 31,783 images for training and testing. For MS-COCO, we use the standard split described by Karpathy 2 for which 5000 images were used for both validation and testing and the rest for training. Similarly, we follow the split of [11] using 1K images for both validation and test and the rest for training. We report the performance of our model and competing methods in terms of six standard metrics used for image captioning as described in [4]. During the pre-training phase, we use both the 2008-2010 News-CommonCrawl and Europarl corpus 3 as out-of-domain training data. Combined, these two datasets consists of

\[ \text{Pre-training} \]

https://github.com/karpathy/neuraltalk2

http://www.statmt.org/wmt11/translation-task.html#download
around 3M sentences, from which we removed sentences shorter than 7 words or longer than 30 words. We also filter out sentences with unseen words in the MS-COCO dataset. After filtering, we create two separate datasets of size 600K and 1.1M, which are then both tokenized and lowercased, and used for the pre-training phase. We train the model with a batch size of 256 and validate on an out-of-domain held-out set. The training is ended when the validation score converges or the maximum number of epochs is reached. After pre-training, we then use the trained parameters to initialize the in-domain training stage.

4.2. Experimental Setting

Our implementation uses Theano [3] and Caffe [10], it is based on the code of [6] 4 and [34] 5. We use GloVe [26] 6 to create the semantic word vectors, the 300-dimensional word embedding pre-trained on Wiki+Gigaword is used for our experiments. We use full image features extracted from the CNN architecture like [30, 7] for decoder LSTM initialization. In our experiments, we set the batch size to 256, vocabulary size to 9.6K, reviewer LSTM hidden size to 300 and decoder LSTM hidden layer size to 1000. Our models are trained on a Tesla K20Xm graphics card with 6G Bytes of memory. We use rmsprop [31] with a learning rate of $10^{-4}$ to optimize the model parameters. Training on MS-COCO takes around 1 days to reach the best performance. We do model selection by evaluating the model on the validation set after every epoch, with the maximum training epoch set to 20. We keep the model with the best BLEU-4 score and test its performance on the test set. We here only report the model performance on the test set. At test time, we do beam search with beam size of 4 to decode words until the end of sentence symbol is reached.

4.3. Evaluation Results

We evaluate our models using six standard metrics and compare our results with other competing methods. The results are shown in Table 1 where “Ours-x” indicates the performance of different variants of our model. The “Baseline” model takes visual concept embeddings as inputs to the attentive decoder without using any bootstrapping or visual feature for initialization. The “Rev” variant adds an input reviewer in front of the attentive decoder to synthesize salient features from the images. The models with “Fc7” and “Pool5” separately uses fc7 layer from VGG [30] and Pool5 layer from ResNet152 [7] for the decoder initialization. The latter brings significant improvements across all the metrics. Models with “Sm” use semantic embeddings as input to the reviewer, while “Rf” use regional features as input. Models with “Bsl” use our pre-train method to bootstrap the model (“large” corresponds to using the 1.1M corpus while “small” is for the 660K corpus, and “noisy” means applying the Noisy Generator). Finally, “Ens” means using an ensemble strategy to combine the results of 5 identical models “Ours-Pools5-Rev-Rf-Bsl” which were trained independently with different initial parameters.

Our model without bootstrap learning (Ours-Fc7-Rev-Sm) gets similar performance to ERD+VGG [35]. When bootstrapped with out-of-domain data, our model outperforms its rival consistently across different metrics. We have also observed that the improvements on Flickr30K is more significant than on MS-COCO, which might partly be due to the smaller amount of training data for Flickr30K. When bootstrapped with out-of-domain data and combined with the reviewer module and ResNet152 features, our ensemble model improves significantly across several metrics and achieves state-of-art performance on both datasets. This clearly demonstrates that bootstrap learning can not only increase n-grams precision but also adapts to human consensus by generating captions that are more diverse.

Bootstrap Learning. We show the evolution of the BLEU-4 score on the validation set in Figure 7. We can see that with bootstrap learning, the model can start from a very good score and keeps increasing afterwards, finally the bootstrapped model outperforms the non-bootstrapped model by a large margin. We also experimented with the “Truth Generator” and “Noisy Generator” variants described in Section 3.6 with varying size of the corpus. The results are shown in Table 1. We observe that adding noise improves the performance in terms of most metrics, which indicates that a model trained with additional noise is more robust, thus produce more accurate captions. Besides, we see that simply enlarging the size of the training corpus (model with “large” in the title) does not help achieve significantly better scores, which might be due to the fact that the additional data is taken from the same source as the smaller one.

Sample Results. We visualize some examples of the captions produced by our model in Figure 8. We would like to make two observations from these examples: (1) using a pre-training phase on additional out-of-domain text data yields a model that can produce a wider variety of captions and (2) the regional features captures more adequate visual concepts.

Results on MS-COCO testing server For fairer comparison, we also submitted our best run results to the MS-COCO testing server and evaluate its performance on official test set. Table 2 shows the performance Leaderboard on official testing image set with 5 reference captions (c5) and 40 reference captions (c40). Please note that we applied

---

4 https://github.com/kelvinxu/arctic-captions
5 https://github.com/s-gupta/visual-concepts
6 https://github.com/stanfordnlp/GloVe
### Table 1: Performance in terms of BLEU-1,2,3,4, METEOR and CIDEr compared to other state-of-the-art methods on MS-COCO and Flickr30K dataset. For the competing methods, we report the performance results cited in the corresponding references. The numbers in red denotes the best known results, the numbers in blue denotes the second best known results, (-) indicates unknown scores. Note that all the scores are reported in percentage.

| Dataset       | Model                           | MS-COCO   |   | Flickr30K   |   |
|---------------|---------------------------------|-----------|---|-------------|---|
|               |                                 | B-1 | B-2 | B-3 | B-4 | METEOR | CIDEr | B-1 | B-2 | B-3 | B-4 |
|               |                                 |     |     |     |     |        |       |     |     |     |     |
| NeuralTalk2 [12] |                                | 62.5 | 45.0 | 32.1 | 23.0 | 19.5  | 66.0  | 57.3 | 36.9 | 24.0 | 15.7 |
| Soft Attention [34]     |                                | 70.7 | 49.2 | 34.4 | 24.3 | 23.9  | -     | 66.7 | 43.4 | 28.8 | 19.1 |
| Hard Attention [34]     |                                | 71.8 | 50.4 | 35.7 | 25.0 | 23.0  | -     | 66.9 | 43.9 | 29.6 | 19.9 |
| MSR [6]                 |                                | -   | -   | 25.7 | 23.6 | -     | -     | -   | -   | -   | -   |
| Google NIC [33]         |                                | 66.6 | 46.1 | 32.9 | 24.6 | -     | -     | 66.3 | 42.3 | 27.7 | 18.3 |
| TWS [22]                |                                | 68.5 | 51.2 | 37.6 | 27.9 | 22.9  | 81.9  | -   | -   | -   | -   |
| ATT-FCN(Ens) [36]       |                                | 70.9 | 53.7 | 40.3 | 30.4 | 24.3  | -     | 64.7 | 46.0 | 32.4 | 23.0 |
| ATT-FCN(Sin) [36]       |                                | 70.4 | 53.1 | 39.4 | 29.3 | 23.9  | -     | 62.8 | 43.7 | 30.1 | 20.7 |
| ERD+VGG [35]            |                                | -   | -   | -   | 29.0 | 24.0  | 89.5  | -   | -   | -   | -   |
| Ours-Baseline           |                                | 68.2 | 50.7 | 37.1 | 26.7 | 23.4  | 84.2  | 61.8 | 41.9 | 28.2 | 18.8 |
| Ours-Fc7-Sm             |                                | 68.6 | 50.7 | 37.3 | 27.7 | 23.6  | 85.5  | 61.9 | 43.0 | 29.4 | 19.6 |
| Ours-Fc7-Rev-Sm         |                                | 70.2 | 53.3 | 39.3 | 28.8 | 23.4  | 87.8  | 62.5 | 43.0 | 29.1 | 19.7 |
| Ours-Fc7-Rev-Sm-Bsl(small) |                             | 70.1 | 53.9 | 39.9 | 29.5 | 23.8  | 90.4  | 63.5 | 44.3 | 30.5 | 20.8 |
| Ours-Pool5-Rev-Sm-Bsl(small) |                        | 72.2 | 54.6 | 40.4 | 29.8 | 24.3  | 92.7  | 66.1 | 47.2 | 33.1 | 23.0 |
| Ours-Pool5-Rev-Sm-Bsl(large) |                       | 72.3 | 54.7 | 40.5 | 30.0 | 24.5  | 93.4  | 66.5 | 47.3 | 33.1 | 22.7 |
| Ours-Pool5-Rev-Sm-Bsl(noisy) |                      | 72.6 | 55.0 | 40.8 | 30.2 | 24.7  | 94.0  | 66.6 | 47.3 | 33.2 | 22.9 |
| Ours-Fc7-Rev-Rf         |                                | 70.6 | 53.6 | 39.5 | 29.0 | 23.6  | 87.6  | 61.8 | 42.9 | 29.4 | 20.0 |
| Ours-Fc7-Rev-Rf-Bsl     |                                | 71.4 | 54.6 | 40.6 | 30.1 | 24.3  | 91.3  | 64.2 | 45.5 | 31.7 | 21.9 |
| Ours-Pool5-Rev-Rf-Bsl   |                                | 72.5 | 55.1 | 41.0 | 30.6 | 24.8  | 95.0  | 66.4 | 47.3 | 33.3 | 23.0 |
| Ours-Pool5-Rev-Rf-Bsl-Ens |                              | 73.4 | 56.5 | 42.5 | 32.0 | 25.2  | 98.2  | 67.2 | 48.2 | 34.0 | 23.8 |

Table 1: Performance in terms of BLEU-1,2,3,4, METEOR and CIDEr compared to other state-of-the-art methods on MS-COCO and Flickr30K dataset. For the competing methods, we report the performance results cited in the corresponding references. The numbers in red denotes the best known results, the numbers in blue denotes the second best known results, (-) indicates unknown scores. Note that all the scores are reported in percentage.

### Table 2: Leaderboard of the published state-of-the-art image captioning models on the online COCO testing server (http://mscoco.org/dataset/#captions-leaderboard), where B@N, M, R, and C are short for BLEU@N, METEOR, and CIDEr scores. All values are reported as percentage (%).

| Dataset       | Model                                | B@1 | B@2 | B@3 | B@4 | CIDEr | METEOR |
|---------------|--------------------------------------|-----|-----|-----|-----|-------|--------|
|               |                                      |     |     |     |     |       |        |
|               | ATT-LSTM-EXT (Ours)                   | 73.4| 91.0| 56.3| 82.3| 42.3  | 71.4   |
|               | ATT [36]                             | 73.1| 90.0| 56.5| 81.5| 42.4  | 70.9   |
|               | Google [33]                          | 71.3| 89.5| 54.2| 80.2| 40.7  | 69.4   |
|               | kimiyoung [35]                       | 72.0| 90.0| 55.0| 81.2| 41.4  | 70.5   |

Table 2: Leaderboard of the published state-of-the-art image captioning models on the online COCO testing server (http://mscoco.org/dataset/#captions-leaderboard), where B@N, M, R, and C are short for BLEU@N, METEOR, and CIDEr scores. All values are reported as percentage (%).

In the same setting as the best model reported in Table 1, our model outperforms the other competing methods over most of the metrics and ranks among the top10 on the Leaderboard.

### 5. Conclusion

In this paper, we proposed a novel training method that exploits external text data without requiring corresponding images. This yields significant improvements in terms of the ability of the language model to generate more accurate captions. We also introduced a new model that borrows some of its key components from existing approaches and suggested new improvements such as using salient region features instead of traditional semantic word embeddings. Our new model together with the suggested pre-training method achieves state-of-the-art performance. Given the wide availability of text data, our pre-training method has the potential of largely improving the generality of most existing image captioning systems, especially for domains with little paired training data.
Figure 7: Qualitative analysis of the impact of bootstrap learning on training time. The y axis represents the BLEU-4 score on the validation set and the x axis denotes the number of epochs.

Figure 8: Qualitative analysis of the impact of the pre-training procedure as well as the use of visual regional features.

6. Acknowledgments

We thank Prof. Juergen Rossmann and Dr. Schristian Schlette from RWTH Aachen University for their support as well as the Aachen Super Computing Center for providing GPU computing.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014. 1, 4
[2] S. Banerjee and A. Lavie. 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, volume 29, pages 65–72, 2005. 2
[3] J. Bergstra, F. Bastien, O. Breuleux, P. Lamblin, R. Pascanu, O. Delalleau, G. Desjardins, D. Warde-Farley, I. Goodfellow, A. Bergeron, et al. Theano: Deep learning on gpus with python. In NIPS 2011, BigLearning Workshop, Granada, Spain, Citeseer, 2011. 7
[4] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325, 2015. 1, 6
[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. International Journal of Computer Vision, 88(2):303–338, June 2010. 2
[6] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. 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. 1, 2, 3, 7, 8
[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385, 2015. 7
[8] L. A. Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, and T. Darrell. Deep compositional captioning: Describing novel object categories without paired training data. arXiv preprint arXiv:1511.05284, 2015. 2
[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997. 1
[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093, 2014. 7
[11] J. Jin, K. Fu, R. Cui, F. Sha, and C. Zhang. Aligning where to see and what to tell: image caption with region-based attention and scene factorization. arXiv preprint arXiv:1506.06272, 2015. 1, 2, 6
[12] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3128–3137, 2015. 1, 2, 8
[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012. 1
[14] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Babytalk: Understanding and generating simple image descriptions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(12):2891–2903, 2013. 2
7. Model Details

7.1. Input Reviewer

The input reviewer uses an LSTM to generate thought vectors. Formally, a thought vector \( f_t \) is computed as

\[
\begin{align*}
    f_t^e &= \sigma(W_r^e[f_{t-1}, v_t^e] + b_c^e) \\
    a_t^e &= \sigma(W_o^e[f_{t-1}, v_t^e] + b_c^o) \\
    i_t^e &= \sigma(W_i^e[f_{t-1}, v_t^e] + b_c^i) \\
    \tilde{f}_t &= \tanh(W_h^e[f_{t-1}, v_t] + b_h^e) \\
    c_t^e &= f_t^e c_{t-1} + i_t^e \tilde{f}_t \\
    f_t &= a_t^e \tanh(c_t^e),
\end{align*}
\]

where \( f_t^e, a_t^e, i_t^e, \tilde{f}_t, c_t^e, f_t \in \mathbb{R}^F \) are the forget/output/input gates and cell input/hidden/output state. These gates and states are controlled by the last thought vector \( f_{t-1} \) and overview feature vector \( v_t \). \( W_r^e, W_o^e, W_i^e, W_h^e \in \mathbb{R}^{F \times (F+D)}, b_c^e, b_c^o, b_c^i, b_h^e \in \mathbb{R}^F \) are the LSTM parameters learned from the dataset. We use the LSTM cell output states \( \{f_t\}_{t=1}^T \) directly as thought vectors. We set the LSTM state size \( F = 300 \) in our experiments.

Semantic Feature. When semantic features are used as input to the reviewer, we set \( d = 300 \), which is the same as the GloVe embedding size. As described, the reviewer contains around \( F \times (F + d) \times 4 = 0.72M \) parameters.

Regional Feature. When regional features are used as input to the reviewer, we set \( d = 2048 \), which corresponds to the dimension of the convolutional feature map of visual concept detector. The reviewer contains around \( F \times (F + d) \times 4 = 2.8M \) parameters in total.

7.2. Decoder

Our decoder is also based on an LSTM architecture, but unlike the reviewer in last section, it also involves the embedding of previous word \( x_{i-1} \) as inputs. Formally, the word distribution \( p(w_i) \) is computed as

\[
\begin{align*}
    f_t^d &= \sigma(W_p^d[h_{i-1}, x_{i-1}, f_t^e] + b_c^p) \\
    a_t^d &= \sigma(W_o^d[h_{i-1}, x_{i-1}, f_t^e] + b_c^o) \\
    i_t^d &= \sigma(W_i^d[h_{i-1}, x_{i-1}, f_t^e] + b_c^i) \\
    \tilde{h}_t &= \tanh(W_h^d[h_{i-1}, x_{i-1}, f_t^e] + b_h^d) \\
    c_t^d &= f_t^d c_{t-1} + i_t^d \tilde{h}_t \\
    h_t &= a_t^d \tanh(c_t^d) \\
    p(w_i) &= \text{softmax}(W_p^1 \tanh(W_p h_i + b_p^1) + b_p^2) \\
    x_i &= Aw_i,
\end{align*}
\]

where \( f_t^d, a_t^d, i_t^d, \tilde{h}_t, c_t^d, h_t \in \mathbb{R}^H \) are the forget/output/input gates and cell input/hidden/output states. These gates and states are controlled by past cell output state \( h_{i-1} \), overview thought vector \( f_t^e \) as well as input symbol \( x_i \), which is more complicated than reviewer. \( W_p^d, W_p^o, W_p^i, W_h^d \in \mathbb{R}^{H \times (H+D)}, b_c^d, b_c^o, b_c^i, b_h^d \in \mathbb{R}^H \) are the decoder LSTM parameters. \( A \) is the word embedding matrix, it transforms the one-hot vector \( w_i \) into an embedding presentation \( x_i \). \( W_p^1 \in \mathbb{R}^{E \times H}, W_p^2 \in \mathbb{R}^{V \times E}, b_p^1 \in \mathbb{R}^{E}, b_p^2 \in \mathbb{R}^V \) are multiple-layer perceptron parameters, which is used to estimate word distribution. We set \( H = 1000, E = 300, V = 9600 \) in our experiments, note that \( E \) is the embedding size and \( V \) is the vocabulary size. The number of parameters is around \( H \times (H + F + E) \times 4 + E \times V + H \times E = 9.5M \).

Pre-trainable Parameters. In the case of semantic features, all the parameter except initialization matrix can be bootstrapped, the dimension of pre-trainable parameters is around \( 0.72M + 9.5M = 10.2M \). In the case of regional features, things are similar so the dimension of all pre-trainable parameters is \( 2.8M + 9.5M = 12.3M \).
Un-pretrainable Parameters The feature transformer $W_h, W_m \in \mathbb{R}^{H \times |\psi(I)|}$ cannot be bootstrapped in our architecture, its high dimension renders it infeasible to be "faked". In case of "Fc7", the total amount of these un-pretrainable parameters is 8.2M, while in "Pool5" case, the total amount is 4.1M.

8. Visualization of Concept Attention & Captions

Figure 9: Additional examples of concept attention
Figure 10: Additional examples of concept attention
| Caption                                                                 | Image                                                                 |
|-----------------------------------------------------------------------|----------------------------------------------------------------------|
| a large plane sitting on top of a runway                              | ![Image of a large plane sitting on top of a runway]                 |
| a red and white airplane parked in front of a building                | ![Image of a red and white airplane parked in front of a building]   |
| a large jetliner sitting in front of a tall building                  | ![Image of a large jetliner sitting in front of a tall building]     |
| a bunch of green bananas on a tree                                   | ![Image of a bunch of green bananas on a tree]                      |
| a large tree filled with lots of green leaves                         | ![Image of a large tree filled with lots of green leaves]            |
| a lot of plants there tops green and stalks are brown                 | ![Image of a lot of plants there tops green and stalks are brown]    |
| a black and white photo of a box                                      | ![Image of a black and white photo of a box]                        |
| a close up of a parking meter on a street                            | ![Image of a close up of a parking meter on a street]               |
| a clock mounted on a stove top oven                                   | ![Image of a clock mounted on a stove top oven]                     |
| a close up of a plate of food on a table                              | ![Image of a close up of a plate of food on a table]                |
| a plate of food on a table with a fork                                | ![Image of a plate of food on a table with a fork]                  |
| the restaurant presents a gourmet breakfast of eggs and toast         | ![Image of the restaurant presents a gourmet breakfast of eggs and toast] |
| a living room with a couch and a table                                | ![Image of a living room with a couch and a table]                  |
| a woman sitting on a couch in a living room                           | ![Image of a woman sitting on a couch in a living room]             |
| a child standing in a room with various paintings and a bed           | ![Image of a child standing in a room with various paintings and a bed] |
| a keyboard and a mouse on a table                                     | ![Image of a keyboard and a mouse on a table]                       |
| a laptop computer sitting on top of a wooden table                    | ![Image of a laptop computer sitting on top of a wooden table]       |
| there are keyboard keys on a wooden table                             | ![Image of there are keyboard keys on a wooden table]               |

Figure 11: Additional examples of captions on the MS-COCO dataset. yellow: without bootstrapping, green: with bootstrapping, orange: groundtruth.
Figure 12: Additional examples of captions on the MS-COCO dataset. yellow: without bootstrapping, green: with bootstrapping, orange: groundtruth.
Figure 13: Additional examples of captions on the Flickr30K dataset. yellow: without bootstrapping, green: with bootstrapping, orange: groundtruth.