Detailed Image Captioning

Junjiao Tian
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
Report Number: CMU-RI-TR-19-34
junjiaot@andrew.cmu.edu

June 9, 2019

Abstract

While researchers have made great improvement on generating syntactically correct sentences by learning from large image-sentence paired datasets, generating semantically rich and controllable content has remained a major challenge. In image captioning, sequential models are preferred where fluency is an important factor in evaluation, *e.g.*, n-gram metrics; however, sequential models generally result in over-generalized expressions that lack the details which may be present in an input image and offer no controllability. In this article, we propose two models to tackle this challenge from different perspective. In the first experiment, we aim to generate more detailed captions by incorporating compositional components into a sequential model. In the second experiment, we explore an attribute-based model with the ability to include selected tag words into a target sentence.

Contents

1 Introduction

Image Captioning: The task of image captioning lies at the intersection of computer vision and natural language processing. Given an image, the task is to generate a natural sentence describing the information present in the input image. Image captioning has received increasing attention over the years. The prevalent encoder-decoder frame work [20] serves as the backbone of many derived models. [21] and [35] introduced and refined the attention mechanism which allows for better feature extraction and interpretability. [30] further used Faster-RCNN [18] to replace the fixed resolution attention mechanism. Researchers [29] [37] also found out that high-level concepts such as attributes provide a more concise representation for an image.

Main drawbacks: First, the majority of existing approaches follows the sequential model where words in the caption are produced in a sequential manner– *i.e.*, the choice of each word depends on both the preceding word and the image feature. Such models largely ignore the fact that natural language has an inherent hierarchical structure [1] [5]. For example, each object can be associated with various attributes. Even with better feature representations and attention mechanisms, the sequential structure of these models tends to lead to generic descriptions that lack specificity. The models [38] [36] exploring compositionality have been shown to produce more accurate, specific, and out-of-distribution sentences. Compositional models, however, do not compare well to the sequential models on the n-gram metrics such as BLEU [2]. Because semantic evaluation metrics such as SPICE [23] ignore fluency and assume well-formed captions, the n-gram metrics are still important in judging the fluency of the generated captions.

Second, training a model on large datasets does not guarantee any controllablity on the output. Often a generated caption describes a given image without specific focus. If a more detailed description is desired for a class of objects, most state-of-the-art models do not accommodate additional inputs to control the focus of a caption on a user-defined target. For example, when a dedicated detector detects different hair styles, it is desired to include such detail in the target sentence where the appearance of a person should be emphasized. If the selected words can be included in the target sentence at appropriate place, users will have more control over the output and the model will be able to produce more detailed captions.

Our approach: In the first experiment we propose an image captioning model that combines the merit of sequential and compositional models by following a word-by-word generation process and combining *grounded* attributes from *specialized* modules. A high-level illustration of the workflow at one time step and visualization of the module attention is shown in Figure 1. More specifically, the algorithm first proposes regions of interest and then chooses a region to focus on depending on the context. The chosen region and the whole image

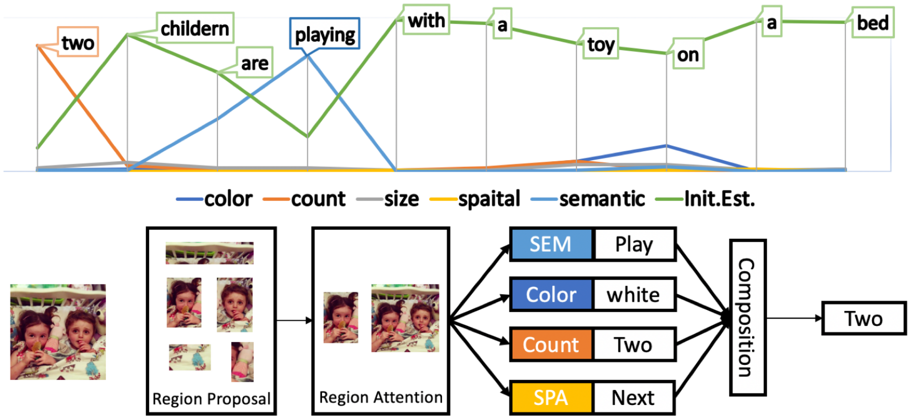


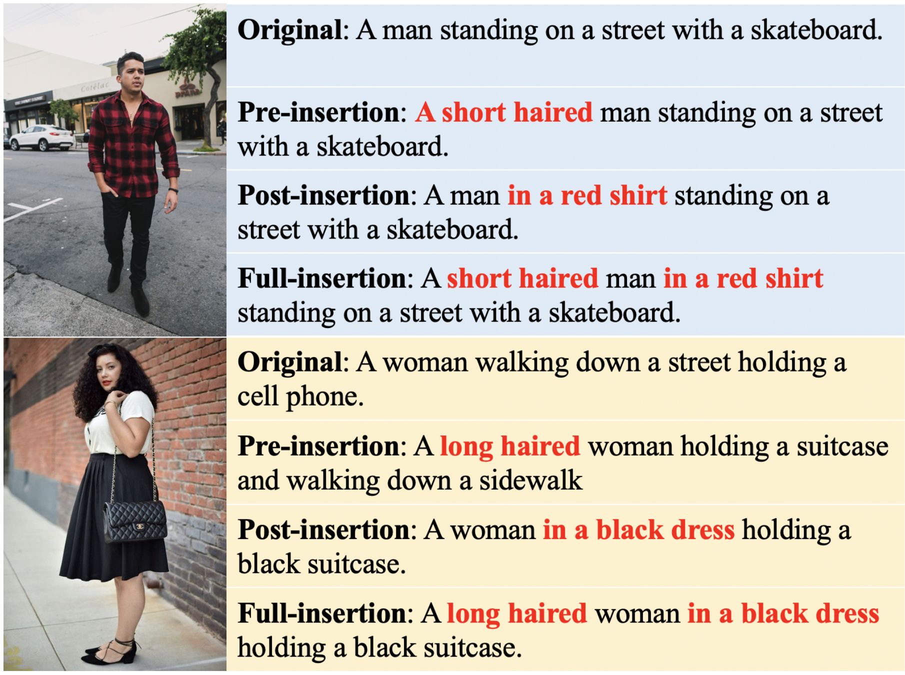
Figure 1: **Top:** Visualization of attribute attention over time: the line plot shows one instance of time varying module attention. Note: Init.Est. stands for initial estimation. **Bottom:** An example of the workflow is shown in a diagram for time step 1 where the word "two" is generated. The model first chooses a region to focus on in the input image and the modules predict the attributes associated with the region. Note: SEM and SPA stand for semantic and spatial modules, respectively.

are fed to a collection of functionally specialized modules where each module is delegated to predict one aspect of the objects such as count, color, and size. This is analogous to the Neural Module Networks (NMN) [24], where each module is responsible for a specialized functionality and the final result is a dynamic composition of different modules. In our case, the model generates the final caption by dynamically attending to different modules. The attributes, therefore, have a hierarchical dependency on and are grounded to the proposed regions. With the proposed Compositional Neural Module Networks, we aim to generate detailed, specific captions without losing fluency, *e.g.*, "a red apple" instead of "a piece of fruit" or "three people" instead of "a group of people." Overall, the main contributions of the first experiment are:

- We develop a hierarchical model, **Modular Attribute Network (MAN)**, that employs both *compositionality* and *sequentiality* of sentence generation.

- Quantitatively, the model outperforms a state-of-the-art model on a set of conventional n-gram metrics and yields a noticeable improvement over the subcategories f-scores of the SPICE metric that is a more meaningful measurement of the semantics of generated captions.

- Qualitatively, we perform human evaluation using Amazon Mechanical Turk. According to the results, our model more often produces more detailed and accurate sentences when compared to the state-of-the-art model.



Original: A man standing on a street with a skateboard.

Pre-insertion: **A short haired** man standing on a street with a skateboard.

Post-insertion: A man **in a red shirt** standing on a street with a skateboard.

Full-insertion: A **short haired** man **in a red shirt** standing on a street with a skateboard.

Original: A woman walking down a street holding a cell phone.

Pre-insertion: A **long haired** woman holding a suitcase and walking down a sidewalk

Post-insertion: A woman **in a black dress** holding a black suitcase.

Full-insertion: A **long haired** woman **in a black dress** holding a black suitcase.

Figure 2: Example of "soft" insertion. In the first example (top), "short hair", "red shirt are the attributes. In the second example (bottom), "long hair", "black skirt" are the attributes

A further analysis shows that the empirical results correlate positively with the quantitative results.

In the second experiment we propose a soft-insertion architecture and a trigger mechanism for an attribute based image captioning model. The model follows a two-step generation process and uses two attribute detectors. One detector,the *concept attribute detector*, is used for extracting high level concepts from an image in the form of unordered attributes. The other detector,the *human attribute detector*, is responsible for detecting specific user-defined attributes and trained on separate datasets. The first attribute detector is an integral part of the language model whereas the second detector is modular and used only during inference. Therefore, the trigger mechanism is used only at test time to "softly" insert specific attributes at appropriate time. Two examples are shown in Figure 2. Overall, the main contributions of the second experiment are:

- We develop a **Soft-Insertion Network (SIN)** for image captioning which follows a two-step generation process to predict the next word based on ranked attribute detection.
- We propose an inference algorithm based on a trigger signal to give more

detailed description about a user-defined class of objects by leveraging additional datasets.

- Qualitatively, we show that our model can generate more detailed description about people.

Paper Organization: In section 2, related works relevant to both models are discussed. The two proposed models will be discussed in sections 3, 4 with corresponding methods, experiments and results sub-sections. In the end, we will conclude by discussing future directions.

2 Related Work

In this section, we briefly introduce related and similar works and emphasize the differences of our model.

Sequential Models: Most recent state-of-the-art models adopt the encode-decoder paradigm, **NIC** [20], where the image content is vectorized by a convolutional network and then decoded by a recurrent network into a caption. In this paradigm, attention-based models have been explored widely. **AdaptATT** [35] follow a top-down attention approach where attention is applied to the output of CNN layers. [29] used a word-based bottom-up attention mechanism. **Top-Down** [30] proposed a feature based *bottom-up attention mechanism* which retains spatial information whereas the word-based approach does not.

Compositional Models: [36] presented a coarse-to-fine two-stage model. First, a skeleton sentence is generated by *Skel-LSTM*, containing main objects and their relationship in the image. In the second stage, the skeleton is enriched by attributes predicted by an *Attr-LSTM* for each skeletal word. **ComCap** [38] proposed a compositional model, where a complete sentence is generated by recursively joining noun-phrases with connecting phrases. A *Connecting Module* is used to select a connecting phrase given both left and right phrases and an *Evaluation Module* is used to determine whether the phrase is a complete caption. In this work, noun-phrases are objects with associated attributes. In general, compositional models exhibit more variation and details in generated captions; however, they tend to perform poorly on the conventional n-gram metrics which are important measurements of fluency.

Neural Module Network (NMN): Researchers have tried to explicitly model the compositionality of language in Question Answering (QA). This line of research shares a similar paradigm, namely, module networks. Module networks are an attempt to exploit the representational capacity of neural networks and the compositional linguistic structure of questions. [24] learns a collection of *neural modules* and a *network layout predictor* to compose the modules into a complete network to answer a question. Rather than relying on a monolithic structure to answer all questions, the NMN can assemble a specialized network tailored to each question. We adopt this idea in QA to design a one-layer NMN with a collection of modules and a composition mechanism in the Modular Attribute Model. The model can compose a customized network depending on the context of a partially generated sentence.

Image Captioning with Attributes: [29] combines visual features with visual concepts in a recurrent neural network. **LSTM-A5** [37] also mines attributes as inputs to a captioning model. Although our models also use attributes, they differ significantly in several aspects. First, the **MAN** model is hierarchical because attributes are grounded exclusively to selected regions that change over time. Second, it is compositional because it combines grounded attributes and objects from separate detectors to predict the next word. Third, the attention is over the set of functionally specialized modules instead of individual visual concepts. Each module specializes in a single descriptive aspect of an object and determines the most probable attribute for that subcategory. For example, the color module generates different predictions for different objects in an image depending on where the model's focus is. Lastly, The **SIN** model uses two attribute detectors, one of which is only used during inference time and is trained on separate mulit-label classification datasets.

Guided Image Captioning: Similar in spirit to our **Soft-Insertion Network** is [22]. The paper introduces a constrained beam search algorithm to force the inclusion of selected tag words in the output sentence which can be expressed in a finite state machine. The model uses attributes at test without the need to retrain a model. This algorithm focuses on novel scenes and objects captioning whereas our model emphasises detailed description for a specific target. Also our model does not require expensive beam search at test time.

Multi-Instance Multi-Label Learning: Multi-Instance Multi-Label (MIML) learning is a more realistic problem. Traditionally, supervised learning associates one feature vector with one label exclusively. In the context of classification, an image is summarized into a single feature vector and an algorithm outputs a single class prediction for that that image. [6] provides a formal definition for MIML, *i.e.*, an image usually contains multiple patches, each of which can be described by a feature vector, and the image can belong to multiple categories since an image can contain more than one class of label. [13][37] adopt Noisy-Or Multi-Instance Learning formulation to classify multiple labels from the same image. [39] proposes an attention-based MIL model. Essentially, the Noisy-Or approach looks at each image patch separately whereas the attention-based model looks at the weighted sum of all image patches. We combine the two approaches into a Stacked Noisy-Or model.

3 Modular Attribute Networks (MAN)

3.1 Method

The proposed hierarchical model for image captioning consists of three main components: Recurrent Neural Network (RNN) Trio, Stacked Noisy-Or Object Detection, and Modular Attribute Detection. We describe the overall captioning architecture as shown in Figure 3, followed by technical details for the three components in Section 3.1.1 –3.1.3 and the objective function used for training in Section 3.1.4.

Inspired by recent successes of region-level attention mechanism [30] [40] [15], we use a Faster-RCNN in conjunction with a Resnet-101 backbone [27] to segment an image into a set of regions that likely contain objects of interest and encode each region r as a fixed-length feature vector $\{v_1, ..v_{D_r}\} \in R^{D_v}$ where D_r is the number of regions, and D_v, the size of the feature vector. The feature vectors are used as inputs to other parts of the network.

The captioning model selects which region to attend to depending on the context. Given the region proposals, the stacked noisy-or object detection mechanism operates to estimate all possible objects in the image regions. The modular attribute-detection mechanism operates on the attended regions to determine appropriate attributes for the attended region at each time step. The object and attribute detection makes up the compositional component while the RNN trio incorporates the detection results to generate a sentence in a sequential manner.

Visual-Nouns and Attributes: Similar to [23], we divide the vocabulary into meaningful subcategories: an object set and five attribute sets which are color, size, count, spatial relationship, and semantic relationship. We select the six word-lists based on the occurrence frequency[1]. The object set consists of visual nouns and the other attribute sets consist of adjectives. For example, *red, green, blue* are in the color set and *sitting, playing, flying* are in the semantic relationship set.

3.1.1 Recurrent Neural Network Trio

The captioning model uses three recurrent neural networks, namely, *Attention (A)-LSTM*, *Visual (V)-LSTM* and *Semantic (S)-LSTM*, to guide the process of generating captions sequentially. The input vector to the A-LSTM at each time step consists of the previous output of the S-LSTM, concatenated with the mean-pooled image feature $\bar{v} = \frac{1}{D}\sum_{i=1}^{D} v_i$ and encoding of the previous word. The attended image region feature, $\tilde{v}_t$, is used as input to the V-LSTM to make an initial estimation of the next word based purely on visual evidence. In the final step, the information from the initial estimation, h_v^t, objects detection, w_t^{obj}, and attributes detection, $\hat{c}_t$, are combined to make the final prediction of the next word.

[1] The lists will be provided in the appendix in the final version.

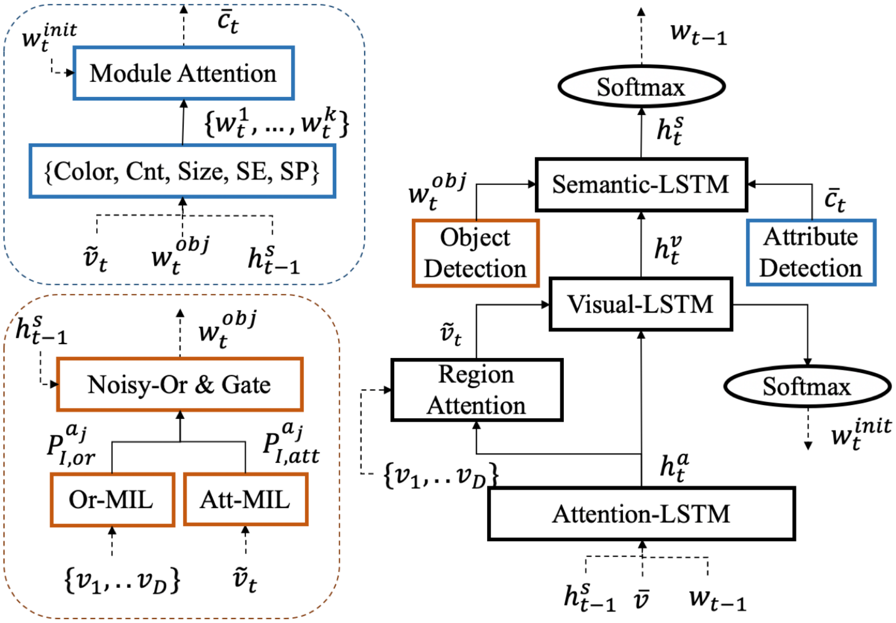


Figure 3: Overview of the architecture: **Right(Black):** Recurrent Neural Network Trio, **Top-Left(Blue):** Modular Attribute Detection, **Bottom-Left(Red):** Stacked Noisy-Or Object Detection. Note: SE denotes Semantic and SP denotes Spatial.

The attended image region feature $\tilde{v}_t$ is obtained through the *Region Attention* mechanism after the A-LSTM:

$$\begin{aligned} a_t &= \text{softmax}(W_b^T \tanh(W_v V + (W_o h_{t-1}))) \\ \tilde{v}_t &= \sum_{i=1}^{D} a_{t,i} v_i \end{aligned}$$

where $V \in R^{D_v \times D_r}$ is the set of image region features.

3.1.2 Stacked Noisy-Or Object Detection

Multi-label classification is a difficult task, where classes are not mutually exclusive in an image. Here, we propose a stacked model that consists of two types of Multiple Instance Learning (MIL) object detectors to consider both image regions and the entire image simultaneously. First, following the Noisy-Or Multiple Instance Learning (MIL) model used in [4] [16], we devise a noisy-or detector to predict a distribution over a set of b labels. The noisy-or operation (*Or-MIL*) is suitable to this task because it operates on each region separately and a positive detection from any region yields a high probability for the whole image. Second, inspired by [39], we adopt an attention based MIL (*Att-MIL*) detector to consider the whole image, which contains large background objects such as “grass.” The two detection probabilities are combined with a second Noisy-Or operation, thus named the stacked approach.

Suppose that, for a given image I, there are $V = \{v_1, v_2, ..., v_{Dr}\} \in R^{D_v}$ image region features proposed by the Faster-RCNN network. The probability of an image containing object a_j is calculated by a Noisy-Or operation on all image regions of this image as follows:

$$P_{I,or}^{a_j} = 1 - \prod_{v_i \in V} \left(1 - p_i^{a_j}\right)$$

where $p_i^{a_j}$ is the probability of object a_j in image region v_i; $p_i^{a_j}$ is calculated through a sigmoid layer on top of the image region features.

For the attention-based MIL detector, instead of an additional attention mechanism, we use the mean-pooled image region feature $\bar{v}$ as follows:

$$P_{I,att}^{a_j} = \frac{1}{1 + e^{-f_{j,att}(\bar{v})}}$$

where $f_{j,att}$ denotes parameters in a two-layer fully connected network.

The final prediction, $P_I^{a_j}$, is computed using a second Noisy-or operation to combine the two probabilities $P_{I,att}^{a_j}$ and $P_{I,or}^{a_j}$:

$$P_I^{a_j} = 1 - \left(1 - P_{I,or}^{a_j}\right)\left(1 - P_{I,att}^{a_j}\right)$$

We also design a gating mechanism to refine the object detection result at each time step. For example, if the word "cat" has already appeared in a sentence, we decrease its priority in the detection result for later time steps even though "cat" remains a positive instance for the image:

$$P_{I,t}^{a_j} = \text{relu}\left(W_h h_{t-1}^s + W_v \tilde{v}_t\right) \circ P_I^{a_j}$$

where blue $P_{I,t}^{a_j} \in R^{D_{obj}}$ is the time-dependent prediction; D_{obj}, the size of the object set; h_{t-1}^s, the output of the S-LSTM at the previous time step; and $\tilde{v}_t$, the attended image region feature at time t.

The output of the object detection module is a word-vector, $w_t^{obj} = E_{obj} P_{I,t}^{a_j}$, where $E_{obj} \in R^{D_{voc} \times D_{obj}}$ is a word embedding matrix from distribution over labels, D_{obj}, to the word-embedding space, D_{voc}. The word-vector w_t^{obj} is used as an input to the S-LSTM for the final decoding.

3.1.3 Modular Attribute Detection

Attribute detection is achieved by using a collection of modules, each module $m \in M = \{m_1, ...m_k\}$ with associated detection parameters θ_m and a *Module Attention* mechanism to predict the layout of the modules. In this section, we describe the set of modules and the composition mechanism.

We use $k = 5$ modules corresponding to different attributes of an object. They are: color, count, size, spatial relationship and semantic relationship modules. The modules map inputs to distributions over discrete sets of attributes. Each module has its own labels and, therefore, learns different behaviours.

The modules all share the same simple architecture. Customizing module architectures for different purposes might result in better performances as in [40] and [31]; in this paper, however, we focus on the overall architecture and leave more sophisticated module architecture designs to future work. The distribution, P_t^m, over labels for module m at time t is computed using a softmax-activated function denoted by f_m:

$$P_t^m = f_m(\tilde{v}_t, h_{t-1}^s, w_t^{obj}).$$

The outputs of the modules are word vectors $w_t^m = E_m P_t^m$, where E_m is the word embedding matrix for module m.

Next, we describe the compositional Module Attention mechanism that selects which module to use depending on the context. Inspired by [35], we use an adaptive attention mechanism and a softmax operation to get an attention distribution of the modules:

$$\begin{aligned} z_t &= W_z^T \tanh(W_m w_{m,t} + (W_g h_{l,t-1})) \\ \alpha_t &= \text{softmax}(z_t) \\ c_t &= \sum_{i=1}^{k} \alpha_{t,i} w_{t,i} \end{aligned}$$

where $w_{m,t} \in R^{D_{voc} \times k}$ is the module network outputs at time t. k denotes the number of modules in consideration. We add a new element $w_t^{init} = Ey_t^{init}$ to the attention formulation. This element is the word vector of the initial estimation of the next word from the V-LSTM.

$$\begin{aligned}
\hat{\alpha} &= \text{softmax}\left([z_t; W_z^T \tanh(W_i w_t^{init} + (W_g h_{t-1}^s))]\right) \\
\beta_t &= \hat{\alpha}[k+1] \\
\hat{c_t} &= \beta_t w_{v,t} + (1 - \beta_t) c_t
\end{aligned}$$

Depending on the context, the network composes a different set of modules to obtain word-vector $\hat{c_t} \in R^{D_{voc}}$ for the S-LSTM.

3.1.4 Objectives

Our system is trained with two levels of losses, **sentence-level loss** and **word-level loss**. We first describe the more conventional sentence-level loss and then the auxiliary word-level losses.

Sentence-Level Loss We apply two cross entropy losses to the V-LSTM and S-LSTM respectively:

$$L_{V/S} = -\sum_{t=1}^{T} \log p(y_t | y_1, ..., y_{t-1}; I; \theta)$$

where θ are the parameters of the models; I, the image; and $y = \{y_1, y_2, ..., y_T\}$, the ground truth sequence.

Word-Level Loss We subdivide the word-level loss into two types: loss $L_{mil}^{att/or}$ to train the object and attribute detectors, and loss L^m to train the module attention mechanism for composing attributes.

Loss from Stacked Noisy-Or object detection: as described in 3.1.2, the MIL object detection has a stacked design. We train the noisy-or detector and attention-based detector using the two sigmoid cross entropy losses respectively:

$$L_{mil}^{att/or} = \sum_{a_j} -y^{a_j} \log(p^{a_j}) + (1 - y^{a_j}) \log(1 - p^{a_j})$$

where y^{a_j} is 1 when ground-truth object a_j is present and 0 otherwise. $p^{a_j} \in \{P_{I,att}^{a_j}, P_{I,or}^{a_j}\}$ is a sigmoid-activated function.

Loss from Modular Attribute detection: we use five masked cross entropy loss to train the attribute detection modules:

$$L^m = \sum_{t=1}^{T} M_t^m \left(-y_t \log(P_t^m) + (1 - y_t) \log(1 - P_t^m)\right)$$

where $m \in M$ and M_t^m is 1 if an attribute from set m is present and 0 otherwise at time t.

Model	BL1	BL4	ROUGE	CIDER	SPICE
NIC**	-	30.2	52.3	92.6	17.4
AdaptATT**	-	31.2	53.0	97.0	18.1
LSTM-A5**	-	31.2	53.0	96.6	18.0
Top-Down**	-	32.4	53.8	101.1	18.7
CompCap*	-	25.1	47.8	86.2	19.9
Top-Down	76.7	32.0	59.0	105.4	19.9
Ours:Module	**77.2**	**33.0**	**59.4**	**108.9**	**20.4**

Table 1: Performance on the COCO Karpathy test split [13]. Higher is better in all columns. * indicates results from the original paper. ** indicates re-implementation of the original papers by [38]. Note: our implementation of the Top-Down model and the proposed model do not use beam-search whereas other results do. BL4/1 denotes BLEU-4 and BLEU-1 respectively.

The composition mechanism is trained with the following additional loss:

$$L_c = \sum_{t=1}^{T} M_t \left(y_{m,t} \log(\hat{\alpha}) + (1 - y_{m,t}) \log(1 - \hat{\alpha}) \right)$$

where M_t is 1 if any ground-truth attribute is present and 0 otherwise. $y_{m,t} \in R^{k+1}$ is a one-hot vector indicating which module is active at time t.

The final loss is a summation of all losses:

$$L = L_V + L_S + L_{mil}^{att} + L_{mil}^{or} + \sum_{m \in M} L_m + L_c$$

where $m \in M$ denotes an individual loss for each attribute module.

3.2 Experiments

3.2.1 Datasets

We use MSCOCO [10] for evaluation. MSCOCO contains 82,783 training and 40,504 validation images; for each image, there are 5 human-annotated sentences. We use the widely-used Karpathy Split [13] to incorporate portion of the validation images into the training set. In total, we use 123,287 images for training and leave 5K for testing. As a standard practice, we convert all the words in the training set to lower cases and discard those words that occur fewer than 5 times and those do not intersect with the GloVe embedding. The result is a vocabulary of 9,947 unique words. For usage of the Visual Genome dataset [34], we reserve 5K images for validation, 5K for testing and 98K images as training data. We refer the readers to [30] for more details on training of the Faster-RCNN network.

Model	OBJ	ATTR	RE	CL	CT	SZ
Top-Down	38.0	8.27	6.83	6.59	9.12	3.86
Ours:Module	**38.7**	**9.39**	**7.23**	**7.92**	**14.70**	4.10

Table 2: SPICE subcategory f-score breakdown on the COCO Karpathy test split [13]. Higher is better in all columns. Note the following abbreviations: OBJ-object, ATTR-attribute, RE-relations, CL-color, CT-count, SZ-size.

3.2.2 Implementation details

We set the number of hidden state units in all LSTMs to 512, and the size of input word embedding to 300. We use a pre-trained GloVe embedding [12] and do not finetune the embedding during training. The pre-trained embedding is from a public website[2] and consists of 6B tokens in total. In training, we set the initial learning rate as 1e-4 and anneal the learning rate to 5e-3 at the end of training starting from the 20th epoch using a fixed batch size of 128. We use the Adam optimizer [9] with $\beta 1$ to be 0.8. We train the Stacked Noisy-Or Object Detector jointly for 5 epoches and stop. The training is complete in 50K iterations.

To ensure fair comparison, we re-train the Top-Down using the same hyperparameters as the proposed model. We report the results with greedy decoding to reduce the effect of hyperparameter search for different models.

We use the top 36 features in each image as inputs to the captioning models and do not finetune the image features during training.

3.2.3 Amazon Mechanical Turk setup

Amazon Mechanical Turk (AMT) is a popular crowdsourcing service from Amazon. To investigate the effect of using compositional modules qualitatively, we design a Human Intelligence Task (HIT) to compare two captions generated from our implementation of the top-down model and the proposed compositional module networks. Each turker is asked to select from four options as shown in Figure 5: either of the two captions, equally good, or equally bad. For each image, we ask 5 workers to evaluate.

For 1,250 images, 6,250 turkers participated. The images are uniformly sampled from the test split; those images with identical captions from the two models are discarded. We design a qualification test to test workers' understanding of the problem and English proficiency. We adopt a max voting scheme to determine the quality of captions per image. When there is a clear winner, we use it as the result for that image. In the case of ties, we give one vote to each tied option.

[2] https://nlp.stanford.edu/projects/glove/

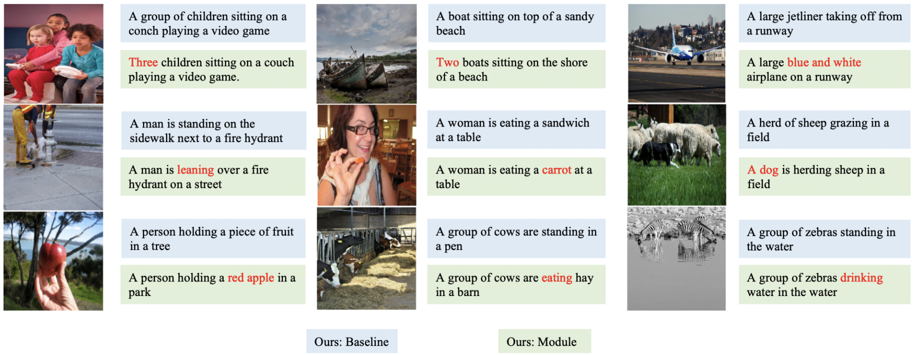

A group of children sitting on a conch playing a video game
Three children sitting on a couch playing a video game.
A man is standing on the sidewalk next to a fire hydrant
A man is leaning over a fire hydrant on a street
A person holding a piece of fruit in a tree
A person holding a red apple in a park
A boat sitting on top of a sandy beach
Two boats sitting on the shore of a beach
A woman is eating a sandwich at a table
A woman is eating a carrot at a table
A group of cows are standing in a pen
A group of cows are eating hay in a barn
A large jetliner taking off from a runway
A large blue and white airplane on a runway
A herd of sheep grazing in a field
A dog is herding sheep in a field
A group of zebras standing in the water
A group of zebras drinking water in the water
Ours: Baseline
Ours: Module

Figure 4: Qualitative examples of captions generated by the Top-Down model (blue) and the proposed compositional module model (green). The proposed model produces more specific action attributes, *e.g.*, "leaning" instead of "standing," due to the semantic module.

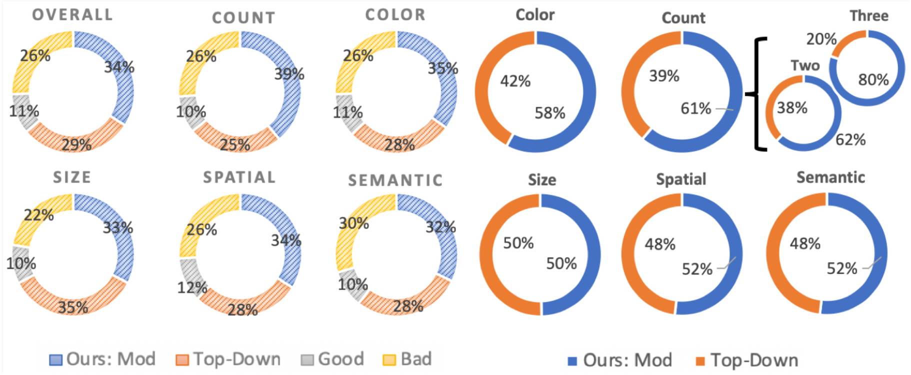


Figure 5: **Left:** Human evaluation results on the Caption Comparison task. The pie plot shows percentage of votes for different options. There are four options for participants, Option 1: *caption 1*, Option 2: *caption 2*, Option 3: *equally good*, Option 4: *equally bad*. **Right:** We count the number of occurrences of words from each subcategory word list in the 5K test split. The pie plot shows the ratio of word occurrences between the two models. We also show two specific examples from the count list, *e.g.*, *two* and *three*.

3.3 Results

We compare our proposed model with our implementation of the **Top-Down** model [30], which achieved state-of-the-art performance on all evaluation metrics previously. We also list the published results of **CompCap** [38], which is another recent compositional model. We also include the published performance of **NIC** [20], **AdaptATT** [35], **Top-Down** and **LSTM-A5** [37] re-implemented by [38] because the re-implementations use comparable visual features and are evaluated on the same test split. There are other models with better performances such as the model proposed by [40], which uses additional datasets to train spatial and semantic relationship detectors. Our work is a fair comparison to the Top-Down model since both models use only MSCOCO as the main training data and Visual-Genome to train the Faster-RCNN, which is also used in [40]. Our implementation of the Top-Down achieves better performance than the implementation by [38] and we use our implementation as the baseline for all comparison.

Shown on the right side of Figure 5, a preliminary analysis of the generated captions shows that our proposed compositional module modle is able to generate captions that include more specific attribute words such as color and count. For example, the proposed model includes **4** times more of specific counts such as *three* in its generated captions.

3.3.1 Evaluation Metrics

We evaluate the approaches on the test portion of the Karpathy Split and compare the proposed approach against best-performing existing models using a set of standard metrics SPICE [23], CIDEr [19], BLEU [2], ROUGE [3], and METEOR [8] as in Table 1. Our proposed model obtains significantly better performance across all n-gram based metrics.

The n-gram metrics alone do not tell the whole story. We also report the performance on a recent metric, SPICE, and its subcategories f-scores in Table 2. When compared to Top-Down, our module model achieves noticeable improvement on all subcategories but one. The **count** subcategory is improved the most. We hypothesize that counting is an inherently difficult task for neural networks and sequential models tend to "play safe" by using generic descriptions instead. This result demonstrates the effect of having dedicated functional modules for composition. It also shows that our proposed model can generate more detailed captions while improving fluency according to the n-gram metrics.

We also note that the **size** subcategory does not gain improvement over the Top-Down model. We hypothesize that this is due to the simple design of the module. Because the concept of size is a comparison between one object and its environment, our design only considers the object itself and the whole image. A more explicit representation of the concept of size such as bounding box might also be helpful.

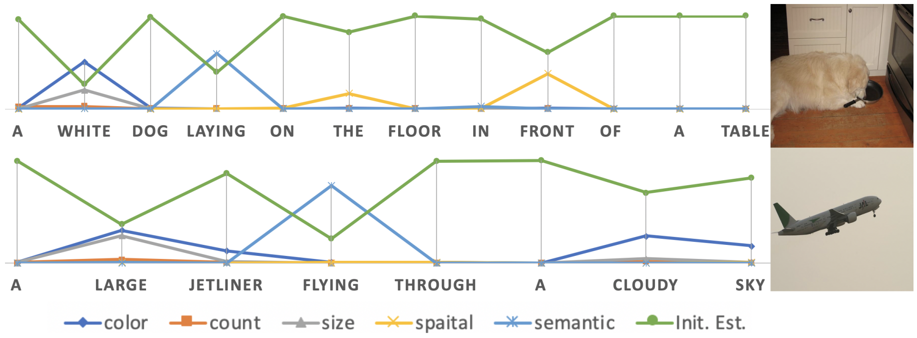


Figure 6: Interpretable visualization of Module attention over time. Note: Init.Est. stands for the Initial Estimation from the V-LSTM

3.3.2 Human Evaluation using Amazon Mechanical Turk

We report the human judgment on the captions generated by the module model and the Top-Down model. As shown in Figure 5, **5**% more people prefer our model over the Top-Down. The difference becomes more significant when we consider subsets of the images. We split the evaluation set into subsets depending on whether their 5 ground truth sentences contain related attributes. For example, images in the Color subset contain at least one ground-truth sentence with a color attribute. The difference is **7**% in the color subset and **14**% in the count subset. This highlights the strength of our model in the subcategories. The human evaluation results qualitatively indicates that there is a discernible improvement recognized by human users.

3.3.3 Qualitative Analysis

Figure 10 shows sample captions generated by the Top-Down model and our proposed model. The examples show that our model gives more accurate description of counting and actions *e.g.*, more precisely describing a person's bent-over pose in the picture by using "leaning" instead of "standing."

Figure 7 shows two examples of changing module attention over time. From the visualization we can analyze the model's choice of attributes in the generated caption. We observe that the color, count, and size modules are more active at the beginning of a sentence and the initial estimation appears more dominant in the later half. More investigation will be needed to draw a conclusive explanation, but we hypothesize that it may be due to the fact that verbs and objects come first in the English language structure.

3.3.4 Ablation Study

To show the effectiveness of each component, we conduct ablation study on different variants of our model and compare the performance on SPICE f-scores and n-gram metrics. To be more specific, **Mod** stands for the modular attribute

Model	SPICE	OBJ	ATTR	RE	CL	CT	SZ
1. Top-Down	19.9	38.0	8.27	6.83	6.59	9.12	3.86
2. Ours:w/o Mod	20.5	38.8	8.80	7.02	6.29	11.9	4.33
3. Ours:w/o MIL	20.0	38.0	8.94	6.87	7.80	12.2	4.11
4. Ours:w/o (Mod+MIL)	20.1	38.3	8.20	6.89	6.03	9.23	4.32
5. Ours:w/o (Mod+AMIL)	20.2	38.5	8.64	7.01	6.51	9.37	4.45
6. Ours:Complete	**20.4**	**38.7**	**9.39**	**7.23**	**7.92**	**14.70**	4.10

Table 3: Ablation study: SPICE subcategory f-score breakdown on the COCO Karpathy test split [13]. Higher is better in all columns. Note the following abbreviations: OBJ-object, ATTR-attribute, RE-relations, CL-color, CT-count, SZ-size.

detectors; **MIL** stands for the stacked Noisy-Or object detectors; **AMIL** stands for the attention based MIL detector. For example, **Ours:w/o (Mod+AMIL)** is a model without modular attribute detectors or stacked MIL detector (but it has a single layer Noisy-Or detector).

In **??**, comparing row 2 and row 6 shows that the modular attribute detectors do not contribute to the improvement on n-gram metrics. Comparing row 4, 5, and 6 indicates that the MIL object detectors contribute the most to improvement on those metrics (cider 106.1→107.1→108.9) and our stacked design further improves the single layer Noisy-Or detector.

In table 3, comparing row 3 and 6, we can see that the MIL object detectors contribute to the object subcategory the most and also affects the performance on other subcategories a little. However, the absence of modular attribute detectors hurts the performance on other subcategories more, such as count (11.9→14.0) and color (6.29→7.95) when comparing row 2 and 6.

In summary, the MIL object detectors contribute to the improvement on n-gram metrics and object subcategory, while the attributes modules improve on the other subcategories. The attribute detectors are responsible for improved semantics and object detectors are primarily responsible for improved fluency.

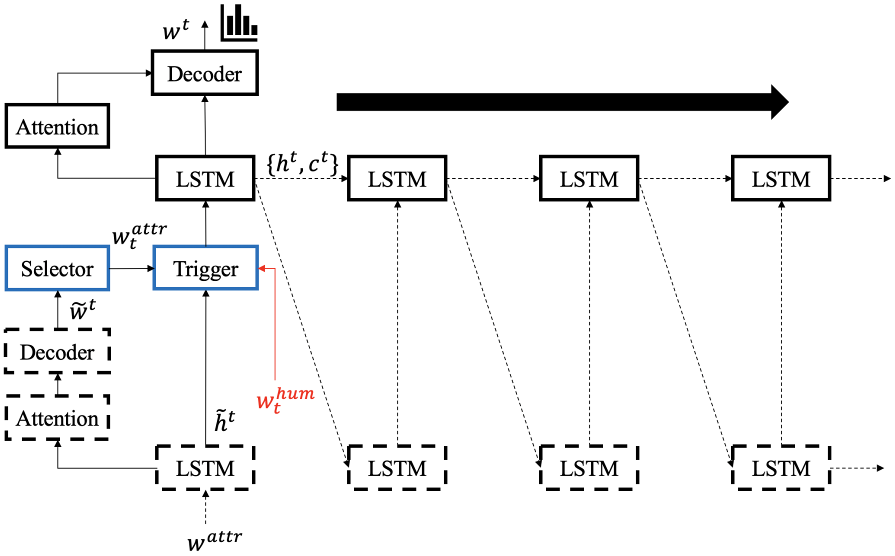


Figure 7: Overview of the two step generation architecture.w^{attr} is the attributes detected with a staked MIL detector described in Section 3.1.2. Dashed lines mean no gradient calculation and backprobagation. w^{attr} is the attributes from the *concept attribute detector and* w^{hum} is the attributes from the *human attribute detector.*

4 Soft-Insertion Network (SIN)

4.1 Method

In this section, we will describe the proposed soft-insertion network and trigger mechanism. Section 4.1.1 describes the overall architecture. Section 4.2.1 gives more details on the trigger mechanism and Section 4.2.2 discusses the objective functions used in training the model.

Normally, during inference time, the trajectory of generating a sentence is not controllable and the generated sentences are subject to strong dataset bias. To give more informative description about an object of interest in the target sentence, we design a look-ahead architecture to predict the next word and a trigger module to signal the network to shift focus. For example, if we want to focus more on the appearance of a person, relevant attributes to the person will be prioritized at appropriate time. We show example captions on pictures crawled from Google Image. The examples demonstrate that the model is able to emphasize specific attributes of a person with the help of an additional *human attribute detector*.

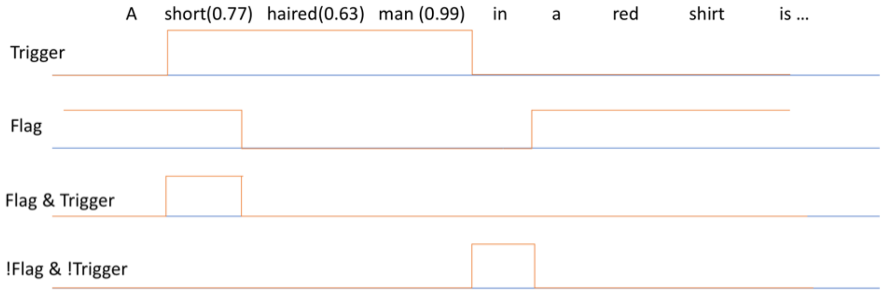


Figure 8: a diagram showing how the trigger signal is used to inject human-centric attributes. The ($Flag\&Trigger$) and ($!Flag\&!Trigger$) signals show when human-centric attributes are injected. $Flag$ and $Trigger$ are boolean variables and ! is a not operator

4.1.1 Two-Step Generation

The goal of the two-step generation is to look ahead on the predicted words and to produce more relevant word proposals. More importantly, it allows for a trigger mechanism to bring in task specific attributes at appropriate time.

At each time step, a *look-ahead prediction* is generated by stepping forward using the LSTM conditioning on the previous word and attributes w^{attr} from the *concept attribute detector*. The previous hidden and cell states h^{t-1}, c^{t-1} are saved. A selector, similar to the gating mechanism in section 3.1.2, is used to refine a list of attributes by conditioning on the predicted next word $\tilde{w}^t$. *A refined prediction* is made by using the ranked attributes and saved previous states. The trigger mechanism is used only during inference time and therefore has no impact in training. In training, losses do back-probagate to the LSTM during look-ahead prediction.
Look-ahead prediction:

$$\tilde{h}^t = f_L(w^{attr}, h^{t-1}, c^{t-1}, w^{t-1})$$

Refined prediction:

$$h^t = f_L(w_t^{attr}, h^{t-1}, c^{t-1}, w^{t-1})$$
$$w^t = f_d(I_t, h^t)$$

where f_L is the LSTM and f_d is the decoder which is a two layer MLP with softmax. Note that during training, the trigger mechanism and human attriubte detecotr are not used. During test time, w_t^{attr} is a combination of attributes from the concept attribute detector and the human attribute detecotr. An inference algorithm with trigger mechanism is presented in the next section.

The two detectors used in the model are stacked Noisy-Or MIL detectors similar to the detector in section 3.1.2. We use the concept attribute detector to extract high level concepts from the image in the form of unordered list of attributes,w^{attr}. The list is refined at everytime step to give more appropriate proposals w_t^{attr} via a *selector* which is similar to the *gating mechanism* in Section 3.1.2. The human attribute detector is used only during inference to detect specific attributes relevant to the appearance of a person,w_t^{hum}.

4.2 Stacked Noisy-Or Attribute Detection

The *concept attribute detector* in this model is slightly different from the Noisy-Or model in **MAN** because we use a ResNet[27] instead of a Faster-RCNN backbone as the vision engine. Therefore, it has a dedicated attention module. Given the output from the last convolution layer from a pre-trained ResNet, let $H = \{h_1, h_2, ..., h_{49}\}$ denotes the features of an image, where $h_i \in R^{2048}$. The attention module is formulated as follows:

$$z = \sum_{k=1}^{49} \alpha_k h_k \tag{1}$$

$$\alpha_k = \text{softmax}(w^T(tanh(Vh_k^T) \odot \sigma(Uh_k^T))) \tag{2}$$

where $w \in R^L, V \in R^{L\times2048}, U \in R^{L\times2048}$ are parameters to be learned and z is the resultant context vector.

The Noisy-Or operator applies a classifier to each region of an image. As long as one region gives high visual confidence on the presence of an attribute, the whole image will be marked as a positive example for that attribute.

The only difference is that the concept attribute detector uses a dedicated attention module and context vector z whereas the object detector in section 3.1.2 uses a mean-pooled image region feature $\tilde{v}$

4.2.1 Trigger Mechanism

The trigger module is responsible for swapping between the two attribute detectors. In this model, the module is trained to respond to key words related to people. When the predicted next word is about a person, the module will prioritize detected human attributes and force the model to describe more about the person in the picture.

A carefully designed trigger signal,*Trig*, ensures that human attributes are prioritized at the appropriate time. The human attributes are injected once before the key word appears in the target sentence and once after the word appears.

Empirically, we found that some human attributes appear after the word "person" and some before it, *e.g.*, "a short haired man" and "a man in short hair" are both valid descriptions. To accommodate this heuristic, one auxiliary signal,the *Flag* signal, is introduced and a visualization is provided in fig. 8. In Figure 2, we show that by manipulating the timing of insertion the same set of attributes shows up in different places in the target sentences.

We also remove human attributes that have been used in the process to avoid repeated inclusion. This is done by keeping track of a list of used attributes.

The architecture of the trigger mechanism is a two-layer MLP. It is trained with sigmoid cross entropy loss jointly with the overall model. A complete algorithm with the trigger mechanism is presented below.

Algorithm: Inference with Trigger Mechanism

```
Given detected attributes $w^{attr}$;
$Flag = 1$;
$UsedList = \{\}$;
while sentence not end do
    Detect $w_t^{hum}$;
    Obtain ranked attributes $w_t^{attr}$;
    if $w_{t-1}$ in $w_t^{hum}$ then
        if $w_{t-1}$in $UsedList$ then
            $UsedList += w_{t-1}$;
        else
            Remove $w_{t-1}$ from $w_t^{hum}$;
        end
    end
    if $w_t^{hum}$ is emtpy then
        $EFlag = 0$;
    else
        $EFlag = 1$
    end
    if $triggerlevel \geq 0.3$ then
        $Trig = 1$;
    end
    $w_t^{attr} = (1 - Trig \times Flag \times EFlag) \times w_t^{attr} + Trig \times Flag \times EFlag \times w_t^{hum}$;

    $w_t^{attr} = (1 - (1 - Trig) \times (1 - Flag) \times EFlag) \times w_t^{attr} + (1 - Trig) \times (1 - Flag) \times EFlag \times w_t^{hum}$;
    $Flag = 1 - Trig$;
end
```

4.2.2 Objectives

The Noisy-Or MIL detector and overall language model are trained with the same losses as described in section 3.1.4. The human attribute detector also uses the same MIL detector design. We will describe the training objective for the trigger mechanism below. A bag of trigger words[3] related to humans are

[3] The list will be provided in the appendix

Model	BL1	BL2	BL3	BL4	ROUGE	CIDEr
Baseline	71.8	51.5	36.3	26.4	55.2	88.3
Ours:MLE	**72.9**	**53.6**	**39.0**	**28.4**	**56.3**	**92.7**
Ours:REINFORCE	75.3	56.7	42.0	30.9	57.8	103

Table 4: Model performance on BLEU, ROUGE and CIDEr

selected.

$$L^{trig} = \sum_{t=1}^{T} z_t * -\log(\sigma(f_{trig}(\tilde{h}_t, I_t))) + (1 - z_t) * -\log(1 - \sigma(f_{trig}(\tilde{h}_t, I_t)))$$

where $f_{trig}(\tilde{h}_t, I_t))$ is a two layer MLP that takes the current estimation of the next hidden state and the attended image as input.

4.3 Experiments

4.3.1 Datasets

We use MSCOOCO for training and evalutation. To train the concept attribute detector, we sample 1000 most frequently used words without stop words in MSCOCO dataset as a predefined set of attributes. This setting fits perfectly in multi-instance multi-label learning. Please refer to section 3.2.1 for more details on MSCOCO.

4.3.2 Web-crawled Multi-label Data

To demonstrate the capability of our trigger mechanism, we create a multi-label classification dataset for training and evaluation of the human attribute detector. The images are crawled from Google Image search and do not overlap with the MSCOCO dataset. To be more specific, we specify 2 attributes for the style of hair and shirt respectively, *i.e.*, "long hair" , "short hair", "striped shirt", "plaid shirt". The images fall into four categories "long hair striped shirt", "long hair plaid shirt", "short hair striped shirt" and "short hair plaid shirt". There are 597, 659, 366 and 499 images for each category. We mix and divide the image into a training set consisting of 2000 images and an evaluation set consisting of 121 images. This is a multi-label dataset where each image contains two labels.

4.4 Results

We first show the performance of our proposed model on the standard n-gram metrics which measure fluency of the generated captions. We compare our attribute boosted model with a baseline model that dose not have attributes. Note that the trigger mechanism is not used in this comparison. We also used REINFORCE[28] to train the model. Table 4 shows that attribute boosting

Model	Recall	Precision	AUC
Overall	0.87	0.79	0.90
Short Hair	0.85	0.73	
Long Hair	0.86	0.82	
Plaid Shirt	0.85	0.84	
Stripped Shirt	0.86	0.70	

Table 5: Human Attribute Detector performance

	Shirt	**Plaid**	Striped	**Short Hair**	Long Hair	Ave.length
W/Trigger	104	4	3	4	0	9.94
Trigger	184	3	2	9	0	10.04

Table 6: Comparison on 499 "Short hair Plaid shirt" images from Google Image.

increases performance and our model is able to generate fluent captions.

We also report the performance of our human attribute detector on our dataset. Table 5 shows that our detector works very well on the small multi-label dataset. It can accurately predict the occurrence of the four attributes in an image.

4.4.1 Gated Attribute Example

In this section, we show the effect of our attribute selector. It functions similarly to the gating mechanism in section 3.1.2. In fig. 9 we can see that when predicting the phrase "a man standing", the attention mechanism focuses on the person and the concept attributes are mostly related to the person whereas when predicting the phrase "on a street", the attention is on the surrounding and the attributes are more related to the environment. Note that the concept attribute detector and the selector proposes multiple possibilities at each time step; "standing","wearing", "young" are all valid attributes to a person.; "road", "sidewalks" and "street" are interchangeable in meaning. A possible way to increase diversity in caption would be to sample from the list of attributes from the concept attribute detector.

4.4.2 Insertion Example

In this section, we show results on our multi-label dataset. We run the captioning model together with the human attribute detector and trigger mechanism on the whole multi-label dataset. We count the number of mentioning of the key words in the generated sentence.

Examining table 6 to 10 shows that our the insertion model is able to significantly increase the number of mentioning of 'Long hair' (21 → 88 in table 7, 19 → 90 in table 9) and "Striped shirt" (32 → 62 in table 8, 45 → 94 in

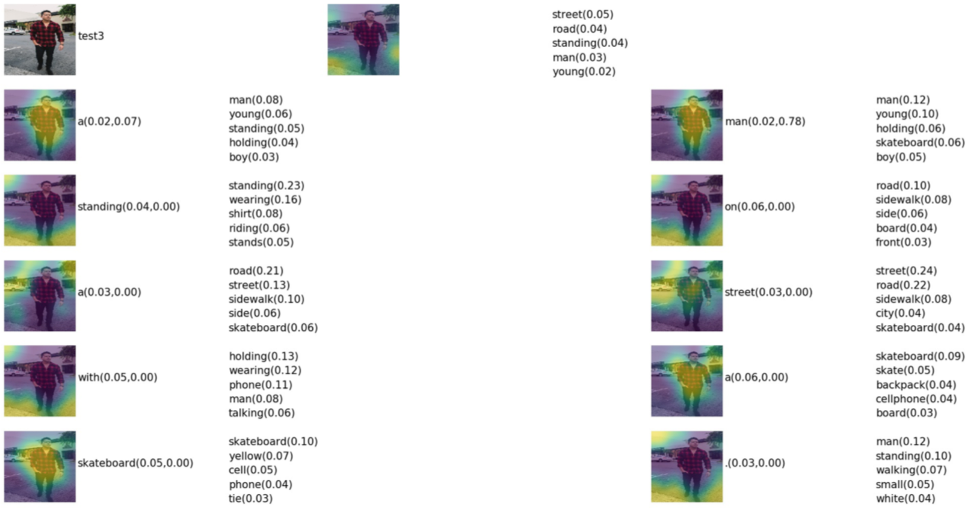
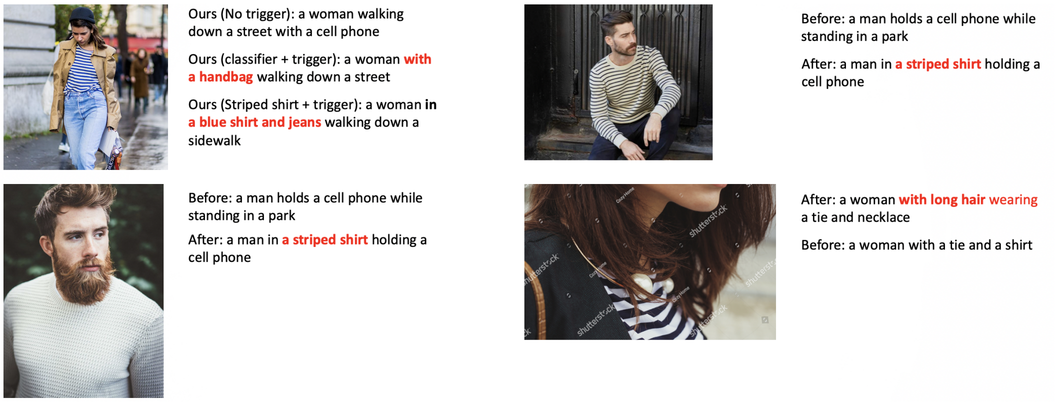


Figure 9: Example of spatial attention and selector. The attributes proposal changes with time depending on the context



Ours (No trigger): a woman walking down a street with a cell phone

Ours (classifier + trigger): a woman with a handbag walking down a street

Ours (Striped shirt + trigger): a woman in a blue shirt and jeans walking down a sidewalk

Before: a man holds a cell phone while standing in a park

After: a man in a striped shirt holding a cell phone

Before: a man holds a cell phone while standing in a park

After: a man in a striped shirt holding a cell phone

After: a woman with long hair wearing a tie and necklace

Before: a woman with a tie and a shirt

Figure 10: Example of insertion

	Shirt	**Plaid**	Striped	Short Hair	**Long Hair**	Ave.length
W/Trigger	105	0	10	0	21	9.27
Trigger	191	1	10	0	88	10.04

Table 7: Comparison on 659 "Long hair Plaid shirt" images from Google Image.

	Shirt	Plaid	**Striped**	**Short Hair**	Long Hair	Ave.length
W/Trigger	71	0	32	0	8	10.0
Trigger	156	0	62	10	9	10.27

Table 8: Comparison on 366 "Short hair Striped shirt" images from Google Image.

	Shirt	Plaid	**Striped**	Short Hair	**Long Hair**	Ave.length
W/Trigger	144	4	45	3	19	9.86
Trigger	248	0	94	1	80	10.08

Table 9: Comparison on 597 "Long hair Striped shirt" images from Google Image.

table 9 and 23 → 49 in table 10) in images where those attributes are actually present.

However, attributes "short hair" and "plaid shirt" do not get boosted from the insertion network. We hypothesize that this is due to a dataset bias in the MSCOCO dataset. Our insertion mechanism is "soft" because the proposed attributes are used as inputs to the LSTM instead of directly to the final decoding layer. This means that the LSTM can choose to use or ignore input attributes based on its trianing. The hypothesis is supported by the fact that the word "plaid" is only mentioned 169 times in MSCOCO whereas "striped" is mentioned 528 times in a total of 82783 captions. Whether the word embedding of "plaid" is not trained well or the LSTM fails to recognize a "plaid" attribute needs further analysis. In either case, dataset bais clearly affects the success rate of "soft" insertion.

In the first example of fig. 10, we compare the results from three scenarios. The first sentence is generated with no trigger mechanism and therefore no human attributes are present, a typical over-generalized caption from sequential models. The second sentence is generated with the trigger mechanism and human attribute classifier trained on the multi-label dataset. The last sentence is generated with the trigger mechanism and a ground truth attribute "striped shirt". Comparing the three results shows that even though the suggested human attributes are not inserted into the target sentece, the sentences with trigger

	Shirt	Plaid	**Striped**	Short Hair	Long Hiar	Ave.length
W/Trigger	52	0	23	0	0	9.86
Trigger	102	0	49	2	0	10.04

Table 10: Comparison on 185 "person in Striped shirt" images from Google Image.

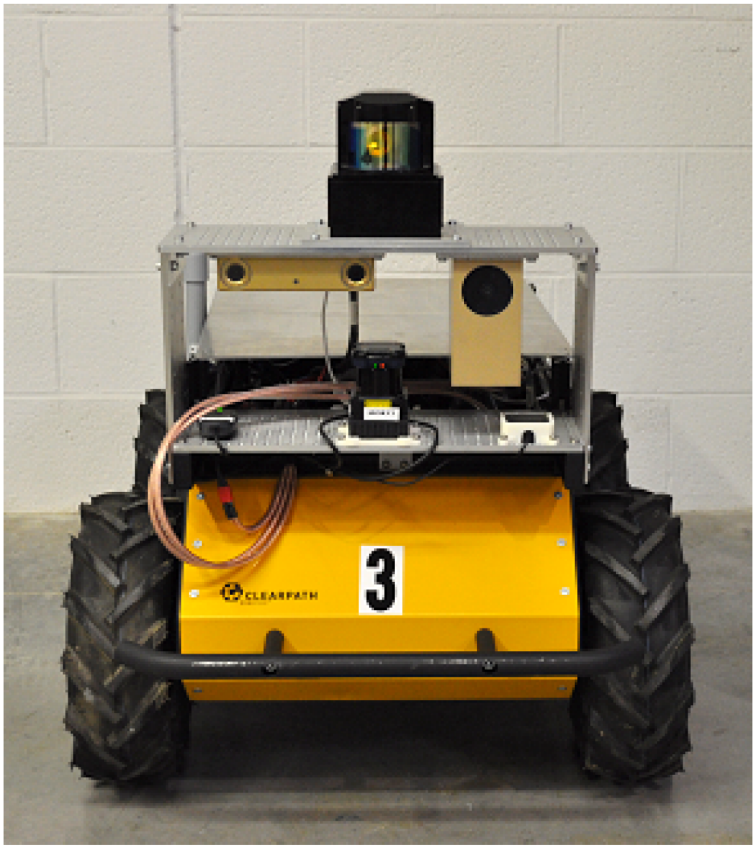


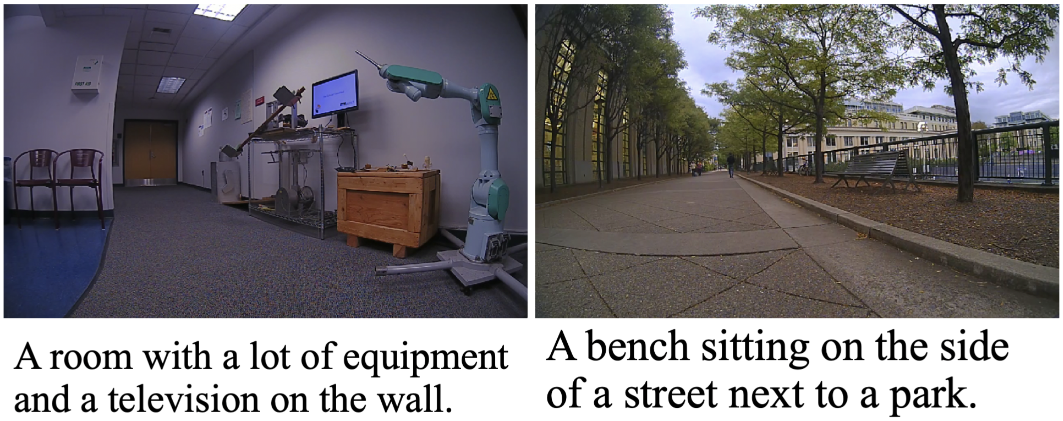
Figure 11: Example of captioning model on robot

mechanism empahsize more on the appearence of the woman in the image. This reflects the "soft" nature of this approach since the influence of the human attributes can be direct in the form of insertion and "soft" in the sense that they shift the focus of the target sentece.

5 Robot Testing

As part of the research project, we depolyed the Insertion Network model on a robot platform, named Husky. Husky is a four wheel mobile robot fig. 11. It has a range of capabilities and is loaded with sensors, *e.g.*, LIDAR sensors, Navigation sensors, platform electronics, etc. The main software interface is Robotics Operating System (ROS)[7].

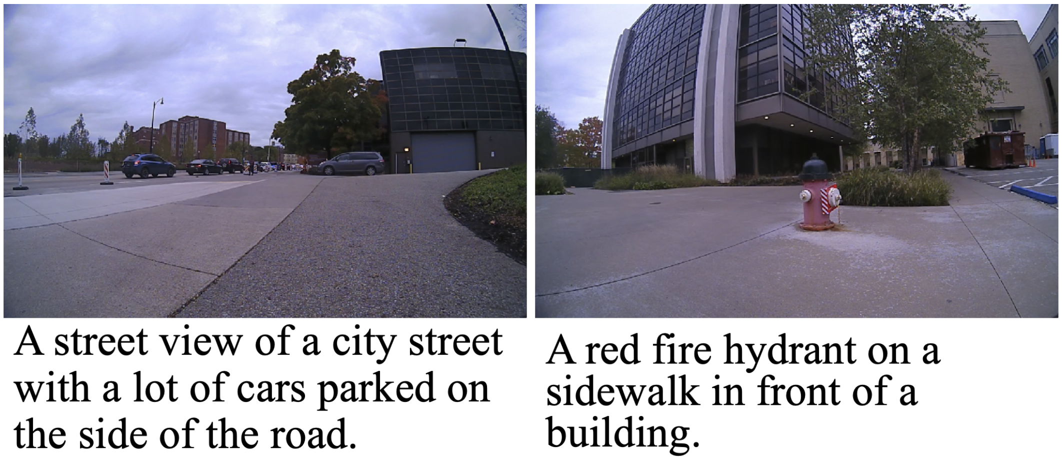
Here we show example captions from the perspective of the robot in fig. 12–14.



A room with a lot of equipment and a television on the wall.

A bench sitting on the side of a street next to a park.

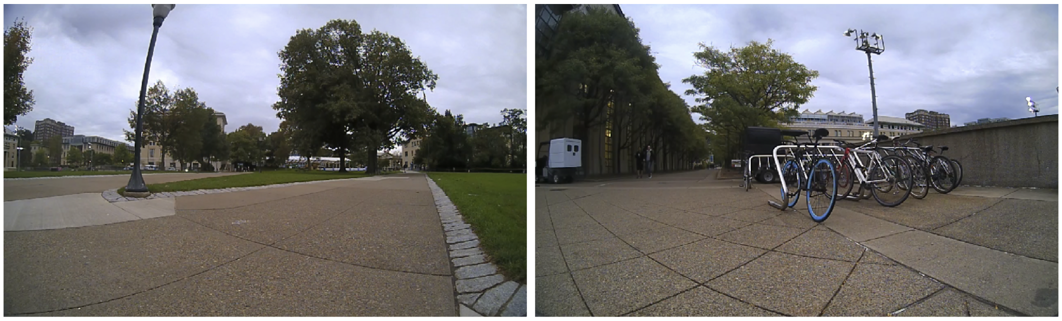
Figure 12: Example of captioning model on robot



A street view of a city street with a lot of cars parked on the side of the road.

A red fire hydrant on a sidewalk in front of a building.

Figure 13: Example of captioning model on robot



A street light on a street with trees in the background.

A group of bikes parked on a sidewalk.

Figure 14: Example of captioning model on robot

6 Future Work

6.1 Visual Reasoning

Image captioning finds its root in machine translation. Inspired by the stellar performance of Recurrent Neural Networks in machine translation, researchers have made great progress with the prevalent CNN-RNN framework. Attention mechanism is also originated from the machine translation community[17][11] and is now widely used in image captioning to provide spatial attention over the entire image. I want to argue that image captioning is very different from machine translation and that the direct input-output mapping algorithm is not sufficient to perform complex scene understanding. In machine translation, input and output contain equal amount of information and direct mapping is desired. However, in image captioning, the information in an image is much richer than the target sentence which only captures the essence and salient objects. Also, the "translation" between sentence and image is the least bijective,*i.e.*, an image can be described in so many different ways. Therefore, the prevalent sequential models tend to explore dataset biases rather than learning to performing visual reasoning on objects' attributes and relations. [32] introduces a model for visual reasoning that has a *program generator* which constructs an explicit representation of the reasoning process. [40] constructs a scene graph which summarizes semantic and spatial relationships, for all the objects in an image in the form of feature vectors. It is apparent that more explicit modeling of reasoning patterns is needed to achieve data-efficiency and generalization. Neural Module Networks provide a good starting point to think about the task in a distributed manner.

6.2 Bayesian Deep Learning

Attribute detection has become an import part of compositional image captioning models. Unlike classification where a single prediction is suffice, image captioning needs more than just a prediction,*i.e.*, uncertainty in prediction is also desired. A common mistake of sequential models is the hallucination of objects that are not present in an image due to dataset bias. Ideally, attribute detectors should detect objects with confidence scores and the captioning model generates a sentence conditioning on detection results. Low confidence should result in omission whereas high confidence should be given more attention.

Most deep learning models do not come with uncertainty measurement. The softmax output is often incorrectly interpreted as confidence. Bayesian Nets provide a framework for estimating uncertainty on parameters as well as on the predictions. [26] provides a mathematical proof that Stochastic Optimization Techniques such as dropout, is in effect performing Bayesian inference. Uncertainty measurements for classification, regression [33], CNN [14] and RNN [25] can be extracted from existing models with no modification. The only trade-off is the increase of time complexity at test time because the method requires multiple stochastic passes to obtain uncertainty.

Research in Bayesian Deep Learning can be integrated into the construction of image captioning models to further improve performance and break dataset bias.

7 Conclusion

In this article, we exploit the benefit of using attributes to boost image captioning models to give more detailed descriptions. We investigate two different models; the modular attribute model uses module-level attention to encourage more details about objects in an image by detecting grounded attributes; the soft-insertion model adopts a test time insertion algorithm to promote more details about a user-defined class of objects in an image, which offers some degrees of controllability in the output. A direction for future research is about compositional reasoning on various properties of objects and sequential integration of the properties into fluent natural language.

References

[1] Christopher D Manning, Christopher D Manning, and Hinrich Schütze. *Foundations of statistical natural language processing.* MIT press, 1999.

[2] Kishore Papineni et al. "BLEU: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting on association for computational linguistics.* Association for Computational Linguistics. 2002, pp. 311–318.

[3] Chin-Yew Lin. "Rouge: A package for automatic evaluation of summaries". In: *Text Summarization Branches Out* (2004).

[4] Cha Zhang, John C. Platt, and Paul A. Viola. "Multiple Instance Boosting for Object Detection". In: *Advances in Neural Information Processing Systems.* Ed. by Y. Weiss, B. Schölkopf, and J. C. Platt. MIT Press, 2006, pp. 1417–1424. URL: `http://papers.nips.cc/paper/2926-multiple-instance-boosting-for-object-detection.pdf`.

[5] Andrew Carnie. "Syntax: A generative introduction". In: (2007).

[6] Zhi-Hua Zhou and Min-Ling Zhang. "Multi-instance multi-label learning with application to scene classification". In: *Advances in neural information processing systems.* 2007, pp. 1609–1616.

[7] Morgan Quigley et al. "ROS: an open-source Robot Operating System". In: *ICRA workshop on open source software.* Vol. 3. 3.2. Kobe, Japan. 2009, p. 5.

[8] 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.* 2014, pp. 376–380.

[9] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[10] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *European conference on computer vision.* Springer. 2014, pp. 740–755.

[11] Minh-Thang Luong et al. "Addressing the rare word problem in neural machine translation". In: *arXiv preprint arXiv:1410.8206* (2014).

[12] Jeffrey Pennington, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP).* 2014, pp. 1532–1543.

[13] Hao Fang et al. "From captions to visual concepts and back". In: (2015).

[14] Yarin Gal and Zoubin Ghahramani. "Bayesian convolutional neural networks with Bernoulli approximate variational inference". In: *arXiv preprint arXiv:1506.02158* (2015).

[15] Junqi Jin et al. "Aligning where to see and what to tell: image caption with region-based attention and scene factorization". In: *arXiv preprint arXiv:1506.06272* (2015).

[16] 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*. 2015, pp. 3128–3137.

[17] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. “Effective approaches to attention-based neural machine translation”. In: *arXiv preprint arXiv:1508.04025* (2015).

[18] Shaoqing Ren et al. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems*. 2015, pp. 91–99.

[19] 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*. 2015, pp. 4566–4575.

[20] Oriol Vinyals et al. “Show and tell: A neural image caption generator”. In: *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE. 2015, pp. 3156–3164.

[21] Kelvin Xu et al. “Show, attend and tell: Neural image caption generation with visual attention”. In: *International Conference on Machine Learning*. 2015, pp. 2048–2057.

[22] Peter Anderson et al. “Guided open vocabulary image captioning with constrained beam search”. In: *arXiv preprint arXiv:1612.00576* (2016).

[23] Peter Anderson et al. “Spice: Semantic propositional image caption evaluation”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 382–398.

[24] Jacob Andreas et al. “Learning to compose neural networks for question answering”. In: *arXiv preprint arXiv:1601.01705* (2016).

[25] Yarin Gal and Zoubin Ghahramani. “A theoretically grounded application of dropout in recurrent neural networks”. In: *Advances in neural information processing systems*. 2016, pp. 1019–1027.

[26] Yarin Gal and Zoubin Ghahramani. “Dropout as a bayesian approximation: Representing model uncertainty in deep learning”. In: *international conference on machine learning*. 2016, pp. 1050–1059.

[27] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[28] Steven J Rennie et al. “Self-critical sequence training for image captioning”. In: *arXiv preprint arXiv:1612.00563* (2016).

[29] Quanzeng You et al. “Image captioning with semantic attention”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4651–4659.

[30] Peter Anderson et al. “Bottom-up and top-down attention for image captioning and VQA”. In: *arXiv preprint arXiv:1707.07998* (2017).

[31] Bo Dai, Yuqi Zhang, and Dahua Lin. “Detecting visual relationships with deep relational networks”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2017, pp. 3298–3308.

[32] Justin Johnson et al. “Inferring and executing programs for visual reasoning”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2989–2998.

[33] Alex Kendall and Yarin Gal. “What uncertainties do we need in bayesian deep learning for computer vision?” In: *Advances in neural information processing systems*. 2017, pp. 5574–5584.

[34] Ranjay Krishna et al. “Visual genome: Connecting language and vision using crowdsourced dense image annotations”. In: *International Journal of Computer Vision* 123.1 (2017), pp. 32–73.

[35] Jiasen Lu et al. “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*. 2017, pp. 375–383.

[36] Yufei Wang et al. “Skeleton Key: Image Captioning by Skeleton-Attribute Decomposition”. In: *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE. 2017, pp. 7378–7387.

[37] Ting Yao et al. “Boosting Image Captioning With Attributes”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 4894–4902.

[38] Bo Dai, Sanja Fidler, and Dahua Lin. “A Neural Compositional Paradigm for Image Captioning”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 656–666.

[39] Maximilian Ilse, Jakub M Tomczak, and Max Welling. “Attention-based deep multiple instance learning”. In: *arXiv preprint arXiv:1802.04712* (2018).

[40] Ting Yao et al. “Exploring Visual Relationship for Image Captioning”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 684–699.