Learning to Select: 
A Fully Attentive Approach for Novel Object Captioning

Marco Cagrandi, Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, Rita Cucchiara

Abstract

Image captioning models have lately shown impressive results when applied to standard datasets. Switching to real-life scenarios, however, constitutes a challenge due to the larger variety of visual concepts which are not covered in existing training sets. For this reason, novel object captioning (NOC) has recently emerged as a paradigm to test captioning models on objects which are unseen during the training phase. In this paper, we present a novel approach for NOC that learns to select the most relevant objects of an image, regardless of their adherence to the training set, and to constrain the generative process of a language model accordingly. Our architecture is fully-attentive and end-to-end trainable, also when incorporating constraints. We perform experiments on the held-out COCO dataset, where we demonstrate improvements over the state of the art, both in terms of adaptability to novel objects and caption quality.

CCS Concepts

- Computing methodologies → Computer vision; Natural language generation.

Keywords

novel object captioning; region selector; constrained beam search.

ACM Reference Format:

1 INTRODUCTION

Describing images has recently emerged as an important task at the intersection of computer vision, natural language processing, and multimedia, thanks to the key role it can have to empower both retrieval and multimedia systems [4, 6, 8–10, 17, 31]. Recent advances in image captioning, indeed, have demonstrated that fully-attentive architectures can provide high-quality image descriptions when tested on the same data distribution they are trained [11, 13, 20, 27]. As the existing datasets for image captioning [23, 40] are limited in terms of the number of visual concepts they contain, though, the application of such systems in real-life scenarios is still challenging. For this reason, the task of Novel Object Captioning (NOC) has recently gained a lot of attention due to its affinity towards real-world applications [1, 12, 15]. This setting, indeed, requires a model to describe images containing objects unseen in the training image-text data, also referred to as out-of-domain visual concepts.

Since the language model behind a NOC algorithm cannot be trained to predict out-domain words, proper incorporation of such novel words during the generation phase is one of the most relevant issues in this task. Early NOC approaches [12, 35] tried to transfer knowledge from out-domain images by conditioning the model at training time on external unpaired visual and textual data. Further works [21, 38] proposed to integrate coping mechanisms in the language model to select words corresponding to the predictions of a tagger. However, these frameworks do not include a proper and explainable method to identify which objects on the scene are more relevant to be described, and consequently, lack on leveraging all the available information provided by visual inputs. On a different line, Anderson et al. [3] devised a Constrained Beam Search algorithm to force the inclusion of selected tag words in the output caption, following the predictions of a tagger.

Inspired by this last line of research, we combine the ability to constrain the predictions from a language model with the usage of object regions and of fully-attentive architectures, which is dominant in traditional image captioning. Precisely, we devise a model with a specific ability to select objects in the scene to be described, with a class-independent module that can work on both in-domain and out-of-domain objects. Further, we combine this with a variant of the Beam Search algorithm which can include constraints produced by the region selector, while assuring end-to-end differentiability. We provide extensive experiments to validate the proposed approach: when tested on the held-out portion of the COCO dataset, our model provides state-of-the-art results in terms of caption quality and adaptability to describe objects unseen in the training set. Given its simplicity and effectiveness, our approach can also be thought of as a powerful new baseline for NOC, which can foster future works in the same area.

2 PROPOSED METHOD

Our NOC approach can be conceptually divided into two modules: an image captioner and a region selector. While the image captioning model is conditioned on the input image and is in charge of modeling a sequence of output words, the region selector is in charge of choosing the most relevant objects which need to be described, regardless of their adherence to the training set. The objects picked by the selector are used as constraints during the generation process, so that the output caption is forced to contain their labels.
as predicted by an object detector. All the components of our architecture are based on fully-attentive structure, and end-to-end training is allowed also when adding constraints to the language model. Fig. 1 shows an outline of the approach.

2.1 Class-Independent Region Selector

The role of the region selector is to identify objects which must be described in the output sentence. Since the object selector will need to work on classes that are unseen in the training set, we adopt a class-independent strategy in which no information about the object class is employed in the feature extraction process. Instead, we model intra-class relationships between objects of the same class, to handle the case in which multiple objects of the same class are present on the scene.

Given a set of regions \( X = \{ x_i \} \) extracted from the input image, along with their classes \( \{ c_i \} \), we extract central coordinates, width, height and, additionally, we compute the object area. We also consider as an extra feature the confidence score \( s_i \) of the object, to obtain a class-independent feature vector:

\[
x_i = \left( \frac{x_c}{w}, \frac{y_c}{h}, \frac{w_i}{w}, \frac{h_i}{h}, s_i \right)
\]

where \( x_c \) and \( y_c \) are the coordinates of the center of the region, \( w_i \) and \( h_i \) its width and height, and \( w \) and \( h \) the image dimensions.

The set of feature vectors obtained for an image is then fed to a sequence of Transformer-like [33] layers, each of them composed by an inner-attention operator and a self-attention operator. The inner-attention operator is devised to connect together regions belonging to the same class, while the self-attention operator provides complete connectivity between elements in \( X \). The combination of these two operators allows the region selector to independently focus on specific clusters of objects, in order to exchange semantically related information and learn intra-class dependencies, and then, to model long-range and diverse dependences.

Given a partition of \( X \) computed according to the class each region belongs to, i.e. \( \{ r_c \subseteq X \mid \forall x_i, x_j \in r_c, c_i = c_j \} \), the result of the inner-attention operator applied over an element of the partition is a new set of elements \( I(r_c) \), with the same cardinality as \( r_c \), in which each element is replaced with a weighted sum of values computed from regions of the same class. Formally, it can be defined as:

\[
I(r_c) = \text{Attention}(W_q r_c, W_k r_c, W_o r_c),
\]

where \( r_c \) is the set of all elements of \( X \) belonging to class \( c \), \( W_q \) are learnable projection matrices, and Attention indicates the standard dot-product attention [33].

The inner attention layer is applied independently over each element of the above-defined partition so that the overall encoding of \( X \) is a new sequence of elements defined as follows:

\[
I(X) = (I(r_1), I(r_2), ..., I(r_C)),
\]

where \( C \) indicates the number of classes. After each inner-attention layer, a self-attention layer is employed to connect elements of different classes together. Formally, it is defined as:

\[
S(X) = \text{Attention}(W_q X, W_k X, W_o X),
\]

where \( W_q \) are, again, learnable projection matrices.

After a sequence of inner- and self-attention layers, in which each pair of operators is followed by a position-wise feed-forward network [33], the region selector outputs a selection score \( Y_i \) for each object proposal. To do so, we apply an affine transformation and a non-linear activation to the output of the last layer:

\[
Y_i = \sigma(\text{RegionSelector}(X_i) W_o),
\]

where \( W_o \in \mathbb{R}^{d \times 1} \) are learnable weights and \( \sigma \) is a sigmoid.

Training. The region selector is trained using a binary cross-entropy loss. To build ground-truth data, for each image we collect the object classes identified by the object detector and construct a binary ground-truth vector indicating whether a class name is contained in at least one of the ground-truth captions associated with the image. We also consider as positives synonyms and plural forms of the object class names. At inference time, we extract the selected objects for each image adopting 0.5 as threshold.

2.2 Image Captioner

After object selection, our image captioning model is responsible for generating a caption using the chosen class names as constraints. Inspired by recent works which employ fully-attentive models in image captioning [11, 13, 25], we create a captioning model with an encoder-decoder structure, where the encoder refines image region features and the decoder generates captions auto-regressively.
when decoding from the language model, we devise a variant of work. Specifically, for each cross-attention, keys and values are pairwise similarities, it exhibits a significant limitation on encoding [13, 16, 25, 27, 32]. However, as self-attention can only encode pairwise distribution to continue the generation of a partially-decoded sequence, it will contain sequences that satisfy all constraints.

To evaluate caption quality, we use standard captioning metrics (i.e. BLEU-4 [28], METEOR [5], ROUGE [22], CIDER [34], and SPICE [2]), while we employ F1-scores [12] to measure the model ability to incorporate new objects in generated captions.

### Algorithm 1: Grid Beam Search

**Input:** Grid $G$ with $n$ columns and $T$ rows, beam size $k$.

**Output:** Top-$k$ hypotheses.

1. **Initialize:** $G \leftarrow \text{initGrid}(n, T, k)$
2. **While:** $t < T$ and $G$ is not full
   1. **For each cell:** $(c, t)$ in $G$
      1. **If:** $c$ is not full
         1. **For each beam:** $b$ in $G[c, \cdot, \cdot]$ with $t + b$ not full
            1. **If:** $b$ is not full
               1. **Add:** $b$ to $G[c, t + b]$ with $t + b$.

3. **Return:** $\text{argmax}_c \{G[c, :] : |c| = k\}$. Remove sequences w/o EOS.

**Oracle Constraints**

<table>
<thead>
<tr>
<th>Region Selector (w/o Inner)</th>
<th>Cross-Entropy Loss</th>
<th>CIDEr Optimization</th>
<th>CIDEr Optimization with DGBS</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Constraints</td>
<td>M C</td>
<td>S</td>
<td>M C</td>
</tr>
<tr>
<td>Top-1</td>
<td>27.2</td>
<td>108.9</td>
<td>20.2</td>
</tr>
<tr>
<td>Top-2</td>
<td>26.2</td>
<td>97.4</td>
<td>19.2</td>
</tr>
<tr>
<td>Top-3</td>
<td>24.4</td>
<td>81.9</td>
<td>16.4</td>
</tr>
<tr>
<td>Oracle</td>
<td>22.9</td>
<td>78.2</td>
<td>19.0</td>
</tr>
</tbody>
</table>

**Table 1:** Evaluation on the held-out COCO test set, when using different constraint selection approaches.

**Encoder.** Recent captioning literature has shown that object regions are the leading solution to encode visual inputs [4, 37, 39], followed by self-attentive layers to model region relationships [11, 13, 16, 25, 27, 32]. However, as self-attention can only encode pairwise similarities, it exhibits a significant limitation on encoding knowledge learned from data. To overcome this shortcoming, we enrich our encoder with memory slots [7, 11]. Specifically, we extend the set of keys and values of self-attention layers with additional learnable vectors, which are independent of the input sequence and can encode a priori information retrieved through attention.

**Decoder.** The decoder is the actual language model, conditioned on both previously generated words and image region encodings. As in the standard Transformer [33], our language model is composed of a stack of decoder layers, each performing a masked self-attention and a cross-attention followed by a position-wise feed-forward network. Specifically, for each cross-attention, keys and values are inferred from the encoder output, while for the masked self-attention, queries, keys, and values are exclusively extracted from the input sequence of the decoder. This self-attention is right-masked so that each query can only attend to keys obtained from previous words.

### 2.3 Including Lexical Constraints

To include the lexical constraints produced by the region selector when decoding from the language model, we devise a variant of the Beam Search algorithm [14, 26] which supports the adoption of single-word constraints. Given a number of word constraints $W = \{w_0, w_1, ..., w_n\}$ and a maximum decoding length $T$, we frame the decoding process in a matrix $G$ with $n$ rows and $T$ columns, where the horizontal axis covers the time steps in the output sequence, and the vertical axis indicates the constraints coverage. Each cell of the matrix can contain a beam of partially decoded sequences.

At iteration $t$, each row $i$ of $G[i, \cdot]$ can be filled in two ways: either by continuing the beam contained in $G[i, t - 1]$ by sampling from the probability distribution of the language model, or by forcing the inclusion of a constraint from $W$. In the former case, the resulting updated beam of sequences is stored in $G[i, t]$, while in the latter case it is stored in $G[i + 1, t]$. At the end of the generation process, the last row of $G$ will contain sequences that satisfy all constraints.

Algorithm 1 reports the pseudo-code of our constrained beam search procedure. There, $k$ indicates the number of elements in each bin, model.step indicates sampling from the language model probability distribution to continue the generation of a partially-decoded sequence, while model.add_constr indicates a function which continues a beam by adding all the available constraints, excluding those which have already been generated for a sequence. Because all the operations required to include constraints are differentiable, we call our constraint inclusion approach Differentiable Grid Beam Search (DGBS), and employ it to fine-tune the image captioneer also when using a CIDEr-D optimization strategy.

### 3 EXPERIMENTS

#### 3.1 Evaluation Protocol

**Dataset.** We conduct experiments on the held-out COCO dataset [12], which consists of a subset of the COCO dataset [23] for standard image captioning, where the training set excludes all image-caption pairs that mention at least one of the following eight objects: bottle, bus, couch, microwave, pizza, racket, suitcase, and zebra. We follow the splits defined in [12] and take half of COCO validation set for validation and the other half for testing.

**Metrics.** To evaluate caption quality, we use standard captioning metrics (i.e. BLEU-4 [28], METEOR [5], ROUGE [22], CIDER [34], and SPICE [2]), while we employ F1-scores [12] to measure the model ability to incorporate new objects in generated captions.

**Implementation details.** To extract geometric features and confidence scores for our region selector, we employ Faster R-CNN [30] with ResNet-50-FPN backbone, trained on COCO [23]. For both training and inference, we discard the detections of the *person* and *background* classes. During training, we use different loss weights...
we use the learning rate scheduling strategy of [33] with a warmup.

Following [11], we use three layers for both encoder and decoder and F1-score averaged over the eight held-out classes, using 3.2 Experimental Results

Table 2: Comparison with the state of the art on the held-out COCO test set.

<table>
<thead>
<tr>
<th></th>
<th>F1 Scores</th>
<th>Captioning Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>In-Domain</td>
<td>Out-Domain</td>
</tr>
<tr>
<td></td>
<td>λ₀, λ₁</td>
<td>M  C  S</td>
</tr>
<tr>
<td>Region Select</td>
<td>0.4 0.6</td>
<td>27.4 111.9 20.3</td>
</tr>
<tr>
<td>Region Select</td>
<td>0.4 0.6</td>
<td>28.1 119.2 21.3</td>
</tr>
<tr>
<td>Region Select</td>
<td>0.3 0.7</td>
<td>27.2 108.7 19.9</td>
</tr>
<tr>
<td>Region Select</td>
<td>0.3 0.7</td>
<td>28.0 116.4 21.2</td>
</tr>
<tr>
<td>Region Select</td>
<td>0.2 0.8</td>
<td>27.4 108.0 19.7</td>
</tr>
<tr>
<td>Region Select</td>
<td>0.2 0.8</td>
<td>27.9 115.3 21.0</td>
</tr>
<tr>
<td>Region Select</td>
<td>0.1 0.9</td>
<td>26.9 97.8 18.2</td>
</tr>
<tr>
<td>Region Select</td>
<td>0.1 0.9</td>
<td>27.9 114.3 20.8</td>
</tr>
</tbody>
</table>

(i.e., λ₀ = 0.2 and λ₁ = 0.8) to balance the importance of zero and one ground-truth values, and we limit the number of object proposals for each image to 10 according to their confidence scores. Region selector features are projected to a 128-dimensional embedding space and passed through N = 2 identical layers, each composed of inner-attention, self-attention, and feed-forward.

For our image captioning model, we extract object features from Faster R-CNN [30] with ResNet-101 finetuned on Visual Genome [4, 19]. Following [11], we use three layers for both encoder and decoder and employ 40 memory vectors for each encoder layer. We represent words with GloVe word embeddings [29], using two fully-connected layers to convert between the GloVe dimensionality (i.e., 300) and the captioning model dimensionality (i.e., 512) before the first decoding layer and after the last decoding layer. Before the final softmax, we multiply with the transpose of the word embeddings. We pre-train our captioning model using cross-entropy and fine-tune it using RL with CIDEr-D reward. During this phase, we use the classes detected by Faster R-CNN, trained on COCO, that are mentioned in the ground-truth captions as constraints for our DGBS algorithm. We limit the number of possible constraints to 5.

All experiments are performed with a batch size equal to 50. For training the region selector and pre-training the captioning model, we use the learning rate scheduling strategy of [33] with a warmup equal to 10,000 iterations and Adam [18] as optimizer. CIDEr-D optimization is done with a learning rate equal to 5 × 10⁻⁶.

3.2 Experimental Results

Table 1 shows the results of our model in terms of captioning metrics and F1-score averaged over the eight held-out classes, using different strategies to train the captioning model. We compare with a variant of our region selector without inner-attention (i.e., w/o Inner) and using the top-k detections, with k = 1, 2, 3, instead of our selection strategy. For reference, we also report the performance when using oracle constraints coming from ground-truth captions. As it can be seen, our solution achieves the best results in terms of both caption quality and F1-score, demonstrating the effectiveness of our region selector for choosing constraints for the captioning model and the importance of the inner-attention operator. Furthermore, by comparing the results with standard CIDEr optimization and those obtained using our DGBS algorithm during training, we can see improved results on both in-domain and out-domain captions, thus confirming the usefulness of our training strategy.

In Table 3, we show the results when using different weights to balance the importance of zero and one ground-truth values. As it can be seen, our complete region selector achieves better performance than the variant without inner-attention, thus further demonstrating the effectiveness of the proposed attention operator. Additionally, employing λ₀ = 0.2 and λ₁ = 0.8 provides the best balance in terms of captioning metrics and F1-score.

Finally, in Table 2, we compare our model with NOC state-of-the-art approaches. As it can be noticed, our region selector obtains the best results in terms of both F1-scores and captioning metrics, achieving a new state of the art on the held-out COCO dataset.

4 CONCLUSION

We have presented a fully-attentive approach for NOC that learns to select and describe unseen visual concepts. Our method is based on a class-independent region selector and an image captioning model trained with a differentiable grid beam search algorithm that generates sentences with given constraints, in an end-to-end fashion. Experimental results have showed that our model achieves a new state of the art on the held-out COCO dataset, demonstrating its effectiveness in successfully describing novel objects.

ACKNOWLEDGMENTS

This work has been supported by “Fondazione di Modena”, by the national project “IDEHA: Innovation for Data Elaboration in Heritage Areas” (PON ARS01_00421), cofunded by the Italian Ministry of University and Research, and by the project “Artificial Intelligence for Cultural Heritage (AI for CH)”, cofunded by the Italian Ministry of Foreign Affairs and International Cooperation.