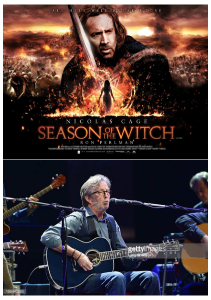
Informative Image Captioning with External Sources of Information

Sanqiang Zhao
University of Pittsburgh
sanqiang.zhao@pitt.edu

Piyush Sharma **Tomer Levinboim** **Radu Soricut**
Google Research, Venice, CA 90291
{piyushsharma,tomerl,rsoricut}@google.com

Abstract

An image caption should fluently present the essential information in a given image, including informative, fine-grained entity mentions and the manner in which these entities interact. However, current captioning models are usually trained to generate captions that only contain common object names, thus falling short on an important "informativeness" dimension. We present a mechanism for integrating image information together with fine-grained labels (assumed to be generated by some upstream models) into a caption that describes the image in a fluent and informative manner. We introduce a multimodal, multi-encoder model based on Transformer that ingests both image features and multiple sources of entity labels. We demonstrate that we can learn to control the appearance of these entity labels in the output, resulting in captions that are both fluent and informative.



Baseline Model: "return to the main poster page for film"

Our Approach: "extra large movie poster image for Season of the Witch"

Baseline Model: "folk rock artist performs on stage during festival"

Our Approach: "Eric Clapton performs on stage during the Crossroads Guitar Festival"

Figure 1: Generating informative captions using fine-grained entity information from external sources; baseline outputs from Sharma et al. (2018).

1 Introduction

Much of the visual information available on the web is in the form of billions of images, but that information is not readily accessible to those with visual impairments, or those with slow internet speeds. Automatic image captioning can help alleviate this problem, but its usefulness is directly proportional to how much information an automatically-produced caption can convey. As it happens, the goal of learning good models for image captioning (in terms of generalization power) is at odds with the goal of producing highly informative captions (in terms of fine-grained entity mentions). For this reason, previous approaches to learning image captioning models at web scale (Sharma et al., 2018) had to compromise on the informativeness aspect, and trained models that could not produce fine-grained entity mentions (e.g., "Season of the Witch") and instead settled for the conceptual (i.e., hypernym) variant for such entities (e.g., "film") as shown in Fig. 1. We present an approach that solves this problem by leveraging upstream models that are capable of producing fine-grained entity names, and integrating them in a controlled manner to produce captions that are both fluent and highly informative.

The standard approach to the image captioning task uses $\langle image, caption \rangle$ pairs to train an image-to-text encoder-decoder model. The "image encoder" is usually a Convolutional Neural Network that extracts image features. The "text decoder" is usually a Recurrent Neural Network or a Transformer Network (Vaswani et al., 2017) that depends solely on these image features to generate the target caption. We identify two limitations of this approach that restrict the amount of information that the generated captions contain:

1. Fine-grained entity recognition is a challenging task in itself, and solving it requires specialized datasets and models. Attempts to simultaneously recognize fine-grained entities and generate an informative caption have previously failed (for example, see Sharma et al. (2018), Fig. 3 therein). In addition, image metadata may be available and requires models capable of smoothly incorporating it.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6485–6494
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

2. The $\langle image, caption \rangle$ pairs on which such models are trained usually have *caption* capturing only a limited coverage of the entities present in the image (or its meta-information). At training time, this limitation gets baked into the models and inherently limits the amount of information presented in the output caption at inference time.

To address the above shortcomings, we define the caption generation task as a new $\langle image, entities \rangle$-to-caption task focused on fluently incorporating entities in the generated caption. We opt for an approach in which entity labels produced by some upstream model(s) are consumed as inputs to the captioning model, in addition to the image pixels. This allows us to use off-the-shelf image labeler models (for object labels, entity recognition, etc.), trained specifically for accuracy on their tasks. To address the second limitation above, we introduce a modeling mechanism that allows us to learn (at training time) and control (at inference time) the coverage of entity mentions in the generated captions. From a modeling perspective, we contribute along these lines by introducing

1. a multi-encoder model architecture that, paired with a multi-gated decoder, integrates image-based information with fine-grained entity information and allows us to generate entity-rich captions
2. a coverage control mechanism that enables us to learn how to control the appearance of fine-grained entities in the generated caption at inference time.

Furthermore, we perform empirical evaluations using both automatic metrics and human judgments, and show that the approach we propose achieves the effect of boosting the informativeness and correctness of the output captions without compromising their fluency.

2 Related Work

Automatic image captioning has a long history, starting with earlier work (Hodosh et al., 2013; Donahue et al., 2014; Karpathy and Fei-Fei, 2015; Kiros et al., 2015), and continuing with models inspired by sequence-to-sequence models (Sutskever et al., 2014; Bahdanau et al., 2014) adapted to work using CNN-based image representations ((Vinyals et al., 2015; Fang et al., 2015; Xu et al., 2015; Ranzato et al., 2015; Yang et al., 2016; Liu et al., 2017), etc.). As training data, the MS-COCO (Lin et al., 2014) is the most used dataset, while the Conceptual Captions dataset (Sharma et al., 2018) is a more recent web-centric, large volume resource.

The work in You et al. (2016); Yao et al. (2017) is related to our approach, as they integrate precomputed attributes into image captioning models. These attributes guide the model to generate captions with correct objects, and are obtained from upstream object detection models that use fairly coarse-grained object labels. Even closer, Lu et al. (2018) propose an approach for incorporating fine-grained entities by generating a "template" caption with fillable slots. They replace entity names in the data with a slot that indicates which entity type should be used to fill that slot, and use a postprocessing step to replace the type slot with the entity name.

The work we present here is novel both with respect to the data preparation and the proposed model. For data, we operate at web-scale level by enhancing Conceptual Captions (Sharma et al., 2018) (3.3M images) with fine-grained annotations. For modeling, we describe a framework that extends Transformer Networks (Vaswani et al., 2017), and allows for the principled integration of multiple, multimodal input signals. This framework allows us to test a variety of experimental conditions for training captioning models using fine-grained labels. In addition, our framework has a coverage control mechanism over fine-grained label inputs, which can differentiate between labels for which we need high-recall and labels for which we desire high-precision.

3 Data and Models

3.1 Data Preparation

The goal of this stage is to obtain annotations that contain (i) entity-rich captions as ground-truth and, (ii) entities associated with each image using fine-grain label detectors. To that end, we build on top of the Conceptual Captions dataset (Sharma et al., 2018), containing 3.3 Million $\langle image, caption \rangle$ pairs. For Conceptual Captions, the ground-truth *caption* is obtained by substituting fine-grained entity mentions in Alt-text[1] with their corresponding hypernyms (e.g., "Los Angeles" is substituted by "city"). Although this sim-

[1] https://en.wikipedia.org/wiki/Alt_attribute

plification makes the captions more amenable to learning, it leads to severe loss of information. To achieve goal (i) above, we reprocessed the URLs from Conceptual Captions and remapped the hypernyms back to their corresponding fine-grained entities (e.g., map “city” back to “Los Angeles”), using the surrounding text as anchors.

3.1.1 Fine-grained Image Labels

To achieve goal (ii) above, we employ pretrained models to extract from input images (1) object detection labels and, (2) web entity labels, using Google Cloud Vision APIs.[2] Object labels refer to fine-grained common objects (e.g., “eucalyptus tree” and “sedan”). Web entity labels, on the other hand, refer to fine-grained named entities (e.g., “Los Angeles” and “Toyota”). In addition to the image pixels, these labels serve as inputs, and during training the model needs to learn a mapping between these labels and the corresponding fine-grained entities in the ground-truth captions.

3.1.2 Selective Hypernym Substitution

An additional issue that needs to be resolved in this data preparation stage is that there is no guarantee of a complete mapping between the fine-grained labels and web entities in the input and the ones in the output (the former are produced by models, while the latter are coming from human-authored Alt-text). Training a model in which output fine-grained entities are not present in the additional input labels is problematic, because it would again require the model to perform both fine-grained entity recognition from pixels as well as caption generation (known to result in hallucination and mis-identification issues, see Sharma et al. (2018)).

To avoid this pitfall, we apply “selective hypernymization” (in the same vein as Sharma et al. (2018)), for which we retain a fine-grained entity in the ground-truth caption only if it is present in the input labels; otherwise, we substitute it by its corresponding hypernym (if present in the ground-truth) or remove it entirely (if not). This step ensures that the data contains a surjective mapping for the fine-grained labels between input and output labels, resulting in learnable mappings between input and output fine-grained labels. For example, in Fig. 1, the raw Alt-text is *“Eric Clapton performs on stage during the 2013 Crossroads Guitar Festival at Madison Square Garden”*. The additional input label are *“Eric Clapton”*, *“Musician”* and *“Crossroads Guitar Festival 2013”*. To ensure the surjective property of the fine-grained label mapping, the mention *“Madison Square Garden”* is removed, resulting in the ground-truth *“Eric Clapton performs on stage during the 2013 Crossroads Guitar Festival”*. Note that we do not enforce a fully bijective mapping between the labels, and may have input labels with no correspondence in the output; for these instances, the model needs to learn that they should not be covered.

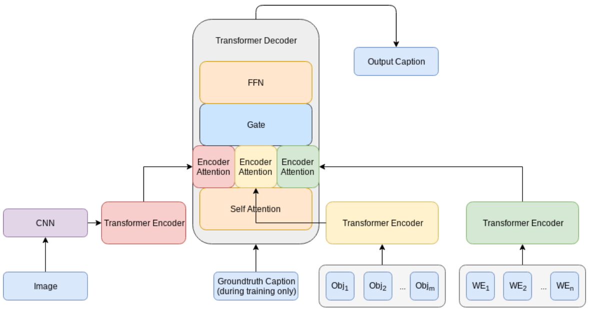


Figure 2: A multi-encoder Transformer Network processes the input image, object labels and web entity labels. The shared decoder attends to all encoders’ outputs and combines their information.

3.2 Models

3.2.1 Multi-Encoder Transformer

We introduce a multi-encoder extension to Transformer Networks (Vaswani et al., 2017) that is used to process our multimodal inputs: image features, object labels, and web entity labels (Fig. 2). Self-attention layers in the Transformer encoder help with learning label representations in the context of the other labels.

Image Encoder To encode the image information, we use a Convolutional Neural Network (CNN) architecture to extract dense image features ($\mathbf{Img} = \{img_1, img_2, ..., img_k\}$) corresponding to a uniform grid of image regions. A Transformer-based encoder takes these image features and embeds them into the features space shared by all input modalities, $\mathbf{H}_{img} = f_{enc}(\mathbf{Img}, \theta_{enc\_img})$, where θ_{enc_img} refers to the parameters for this image encoder.

Object Label Encoder The input for this encoder is an ordered sequence of object labels,

[2] https://cloud.google.com/vision Google Cloud Vision API uses Google Image Search to find topical entities like celebrities, logos, or news events.

sorted by the confidence score of the model that predicts these labels. This allows the model to learn that labels appearing at the head of the sequence are more reliable. For each separate label, we create learnable segment embeddings (inspired by Devlin et al. (2018)) as shown in Fig. 3, using the subtokenization scheme described in Sennrich et al. (2015). A Transformer-based encoder network takes these object label features, $\mathbf{Obj} = \{obj_1, obj_2, ..., obj_m\}$, and embeds them into the features space shared by all input modalities, $\mathbf{H}_{obj} = f_{enc}(\mathbf{Obj}, \theta_{enc\_obj})$, where θ_{enc_obj} refers to the parameters for this object encoder. We do not apply positional embeddings because the relative positions of object labels are irrelevant.

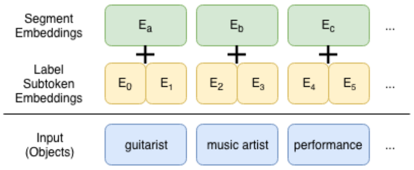


Figure 3: Learnable representations for the Object labels using their surface tokens.

Web Entity Label Encoder For modeling web entity labels, we experiment with two modeling variants that consider either (i) the web entity type, or (ii) the web entity surface tokens. For (i), we obtain entity types by using the Google Knowledge Graph (KG) Search API to match the web entity names to KG entries. Each of these types is subsequently represented by a trainable embedding vector. The model is trained to predict captions with entity types, which during post-processing are substituted by the highest scored web entity label of the predicted type. If no such typed label exists, we use the generic name of the type itself (e.g., "film").

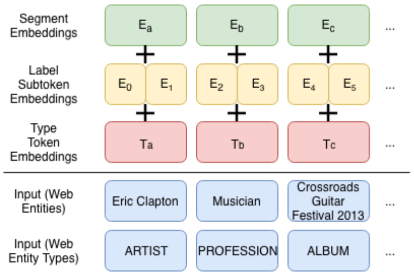


Figure 4: Learnable representations for the Web Entity labels using the surface tokens (and their types).

For variant (ii), the model directly attempts to model and generate a caption containing the surface tokens. In this case, the entity type is still provided as an input to the model as additional source of information. These input representations are constructed by summing up a trainable segment embedding with the subtoken embeddings and the type embedding (Fig. 4). A Transformer encoder network takes these web entity features, $\mathbf{WE} = \{we_1, we_2, ..., we_n\}$, and embeds them into the feature space shared by all input modalities, $\mathbf{H}_{we} = f_{enc}(\mathbf{WE}, \theta_{enc\_we})$, where θ_{enc_we} refers to the parameters for the web entity encoder. Similar to object labels, positional embeddings are not applied to the sequence of web entity labels.

3.2.2 Multi-gated Transformer Decoder

To accommodate the multi-encoder architecture on the input side, we propose a multi-gated extension to the Transformer decoder (Vaswani et al., 2017). As usual, this decoder is a stack of k identical layers, where each of these layers has 3 sub-layers: a self-attention layer, an encoder-attention layer, and a fully connected feed-forward layer. Among the 3 sub-layers, we modify the encoder-attention sub-layer by introducing a mechanism that combines information coming from different encoders.

Formally, we denote the hidden states of n-th layer by $\mathbf{Z}_n = z_{n,1}, ..., z_{n,T}$ ($\mathbf{Z}_0$ refers to decoder input embeddings, and T is the length of decoder inputs). The self-attention sub-layer equation is given in Eq. 1; as expected, the inputs to this layer are masked to the right, in order to prevent the decoder from attending to "future" positions (i.e., $z_{n,j}$ does not attend to $z_{n,j+1}, ..., z_{n,T}$).

$$\mathbf{Z'}_{n,j} = \text{SelfAttn}\big(z_{n,j}, \mathbf{Z}_{n,1:j}, \theta_{self\_attn}\big) \qquad (1)$$

Next, the encoder-attention sub-layer contains three attention modules, which enables it to attend to the three encoder outputs (Eq. 2):

$$\begin{aligned} \mathbf{Z''}^{img}_{n,j} &= \text{EncAttn}(z'_{n,j}, \mathbf{H}_{img}, \theta_{enc\_attn\_img}) \\ \mathbf{Z''}^{obj}_{n,j} &= \text{EncAttn}(z'_{n,j}, \mathbf{H}_{obj}, \theta_{enc\_attn\_obj}) \\ \mathbf{Z''}^{we}_{n,j} &= \text{EncAttn}(z'_{n,j}, \mathbf{H}_{we}, \theta_{enc\_attn\_we}) \end{aligned} \qquad (2)$$

We expect the model to have the ability to adaptively weight each of these three source of information, and therefore we introduce a multi-gate sub-layer. For each source S, we compute a gate $Gate^S_{n,j}$ value that determines the amount of information that flows through it (Eq. 3). Each gate value is computed by transforming the concatenate of the outputs from the three encoder attentions.

$$
\begin{aligned}
Gate^{img}_{n,j} &= \tanh(\mathbf{U}_{img} * \text{concat}(\mathbf{Z}''^{img}_{n,j}; \mathbf{Z}''^{obj}_{n,j}; \mathbf{Z}''^{we}_{n,j})) \\
Gate^{obj}_{n,j} &= \tanh(\mathbf{U}_{obj} * \text{concat}(\mathbf{Z}''^{img}_{n,j}; \mathbf{Z}''^{obj}_{n,j}; \mathbf{Z}''^{we}_{n,j})) \\
Gate^{we}_{n,j} &= \tanh(\mathbf{U}_{we} * \text{concat}(\mathbf{Z}''^{img}_{n,j}; \mathbf{Z}''^{obj}_{n,j}; \mathbf{Z}''^{we}_{n,j}))
\end{aligned} \tag{3}
$$

The output of gate sub-layer is a soft switch that controls the information flow from the three encoders (Eq. 4):

$$
\begin{aligned}
\mathbf{Z}'''_{n,j} = Gate^{img}_{n,j} &* \mathbf{Z}''^{img}_{n,j} \\
+\ Gate^{obj}_{n,j} &* \mathbf{Z}''^{obj}_{n,j} \\
+\ Gate^{we}_{n,j} &* \mathbf{Z}''^{we}_{n,j}
\end{aligned} \tag{4}
$$

Finally, as in the vanilla Transformer decoder, the third sub-layer is a feed-forward network that processes the representation for the next n+1 layer:

$$
\mathbf{Z}_{n+1,j} = \text{FFN}(\mathbf{Z}'''_{n,j}) \tag{5}
$$

The three sources of information (image, object labels, and web entity labels) are treated symmetrically in the above equations. However, the only "true" source of information in this case is the image, whereas the other labels are automatically-produced annotations that can vary both in quality and other properties (e.g., redundancy). We introduce an asymmetry in the modeling that will allow us to capture this important distinction.

3.2.3 Label Coverage Control

Because of the asymmetry between the image input and the label inputs, we introduce a mechanism to control the coverage of the supplied object and web entity labels in the generated caption. This mechanism consists of two parts: (i) two regressor models that learn coverage scores correlating input labels (one for objects, one for web entities) with output mentions, and (ii) two control "knobs" that allow us to specify desired coverage scores at inference time. This coverage control mechanism is inspired by the Label-Fine-Tuning model of Niu and Bansal (2018), although it is used here to achieve a different goal.

Coverage of Object Labels An interesting property of the object labels is that they may be repetitive, often at various levels of granularity, for instance "table", "office table" and "office". A model that would require to reproduce all of them in the output caption will likely produce a disfluent caption containing repetitive mentions of the same object. We introduce object level coverage as a precision-like score for object labels, Cov^p_{obj}, defined as the fraction of output caption tokens present in the input object labels (Eq. 6).

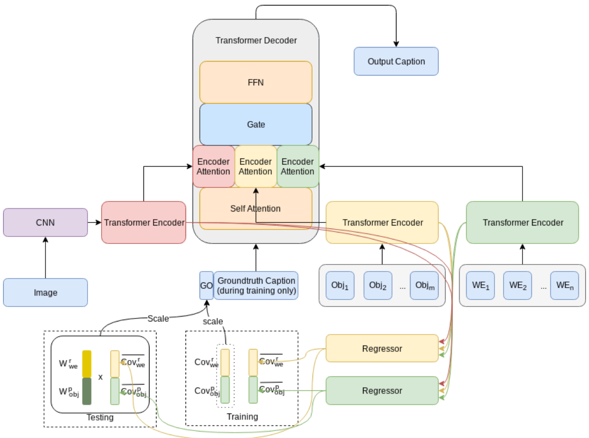


Figure 5: The Image Transformer Encoder (left side) is the only "true" source of information; the Transformer Encoders on the right side encode model-produced sources of information. The control mechanism (two Regressor models) learns this asymmetry (during training), and provides additional run-time control.

Coverage of Web Entities In contrast with the object labels, web entity labels are not repetitive, and tend to have high information value. For that reason, we want a high fraction of input web entities to be used in the output caption. Therefore, we introduce the web entity coverage as a recall-like score for web entity labels, Cov^r_{we}, defined as the fraction of input tokens that are present in the caption (Eq. 6).

$$
\begin{aligned}
Cov^p_{obj} &= \frac{|\{\text{objects tokens}\} \cap \{\text{caption tokens}\}|}{|\{\text{caption tokens}\}|} \\
Cov^r_{we} &= \frac{|\{\text{entities tokens}\} \cap \{\text{caption tokens}\}|}{|\{\text{entities tokens}\}|}
\end{aligned} \tag{6}
$$

3.2.4 Label Coverage Prediction & Control

We train two regressors to predict the coverage scores for object labels ($\overline{Cov^p_{obj}}$) and web entity labels ($\overline{Cov^r_{we}}$), using as features the outputs of the Transformer encoders (Eq. 7). At training time, ground-truth captions are known, so the regression target values Cov^p_{obj} and Cov^r_{we} are computed using Eq. 6. When training regressors parameters ($\mathbf{U}^p_{obj}$ and $\mathbf{U}^r_{we}$), we fix the Transformer parameters and minimize the regression losses (Eq. 8).

$$
\begin{aligned}
\overline{Cov^p_{obj}} &= \text{sigmoid}(\mathbf{U}^p_{obj}\text{concat}(\mathbf{H}_{img}; \mathbf{H}_{obj}; \mathbf{H}_{we})) \\
\overline{Cov^r_{we}} &= \text{sigmoid}(\mathbf{U}^r_{we}\text{concat}(\mathbf{H}_{img}; \mathbf{H}_{obj}; \mathbf{H}_{we}))
\end{aligned} \tag{7}
$$

$$
\begin{aligned}
loss^{reg}_{obj} &= (Cov^p_{obj} - \overline{Cov^p_{obj}})^2 \\
loss^{reg}_{we} &= (Cov^r_{we} - \overline{Cov^r_{we}})^2
\end{aligned} \tag{8}
$$

We compose a coverage indicator vector of the same dimensionality as the word embeddings by tiling the two coverage scores, and use this coverage indicator vector to scale (element-wise) the start token of the output caption. At training time, we use the actual coverage scores, Cov^r_{we} and Cov^p_{obj} (see 'Training' box, the lower part of Fig. 5). At run-time, we use the scores predicted by the regressors, $\overline{Cov^r_{we}}$ and $\overline{Cov^p_{obj}}$ (box labeled 'Testing' in Fig. 5), which we can additionally scale using the two scalars W^r_{we} and W^p_{obj}. These additional scalars act as coverage boost factors and allows us to control, at inference time, the degree to which we seek increased coverage and therefore obtain captions that are both fluent and more informative (by controlling W^r_{we} and W^p_{obj}).

4 Experiments

Dataset We extended the Conceptual Captions dataset as described in Section 3.1. We use the standard (v1.0) splits with 3.3M training samples, and approximately 28K each for validation and test. The human evaluations use a random sample of 2K images from the test set.

Image Processing In this work, we use ResNet (He et al., 2016) for processing the image pixels into features (output features size 7x7x2048), pretrained on the JFT dataset (Hinton et al., 2015).[3] Input images undergo random perturbations and cropping before the CNN stage for better generalization.

Text Handling We use subtoken embeddings (Sennrich et al., 2015) with a maximum vocabulary size of 8k for modeling caption tokens, web entities and object labels. Captions are truncated to 128 tokens. We use an embedding size of 512, with shared input and output embeddings.

Model Specification We use 1 layer and 4 attention heads for Web Entity label encoder; 3 layers and 1 attention head for Object label encoder; 1 layer and 4 attention heads for the CNN encoder; and 6 layers and 8 heads for the shared decoder.

Optimization MLE loss is minimized using Adagrad (Duchi et al., 2011) with learning rate 0.01 and mini-batch size 32. Gradients are clipped to global norm 4.0. We use 0.2 dropout rate on image features to avoid overfitting. For each configuration, the best model is selected to maximize the CIDEr score on the development set.

Inference During inference, the decoder prediction of the previous position is fed to the input of the next position. We use a beam search of size 4 to compute the most likely output sequence.

4.1 Quantitative Results

We measure the performance of our approach using three automatic metrics (see Eq. 6):

CIDEr measures similarity between output and ground-truth (Vedantam et al., 2015).

Web-Entity coverage Cov^r_{we} measures the recall of input web entity labels in the generated caption.

Object coverage Cov^p_{obj} measures the precision of the output caption tokens w.r.t. input object labels.

To measure how well our model combines information from *both* modalities of inputs (i.e. image and entity labels), we compare its peformance against several baselines:

Image only : Anderson et al. (2018) (using Faster R-CNN trained on Visual Genome) and Sharma et al. (2018) (using ResNet pretrained on JFT)

Entity-labels only : Transformer model trained to predict captions from a sequence of entity labels (vanilla Transformer encoder/decoder with 6 layers and 8 attention heads).

Image&Entity-labels : Lu et al. (2018) (w/ Transformer), their template approach implemented on top of a Transformer Network.

Baseline	Image|Label	CIDEr	Cov^r_{we}	Cov^p_{obj}
Labels-to-captions	N|Y	62.08	21.01	6.19
(Anderson et al., 2018)	Y|N	51.09	7.30	4.95
(Sharma et al., 2018)	Y|N	62.35	10.52	6.74
(Lu et al., 2018) w/ T	Y|Y	**69.46**	**36.80**	**6.93**

Table 1: Baseline model results, using either image or entity labels (2nd column). The informativeness metric Cov^r_{we} is low when additional input labels are not used, and high when they are.

Table 1 shows the performance of these baselines. We observe that the image-only models perform poorly on Cov^r_{we} because they are unable to identify them from the image pixels alone. On the other hand, the labels-only baseline and the proposal of Lu et al. (2018) has high performance across all three metrics.

[3] This configuration performed best among the CNN and pretraining conditions we evaluated against.

Entity	W^r_{we}	W^p_{obj}	CIDEr	Cov^r_{we}	Cov^p_{obj}
Type	1.0	1.0	74.60	40.39	6.87
Type	1.5	1.0	70.81	**42.95**	7.04
Type	1.0	1.5	73.82	40.03	8.38
Type	1.5	1.5	71.11	41.94	**8.48**
Name	1.0	1.0	**87.25**	31.01	6.27
Name	1.5	1.0	83.62	38.08	6.76
Name	1.0	1.5	83.34	30.64	7.74
Name	1.5	1.5	82.18	**38.17**	**7.93**

Table 2: Variants of our proposed approach using both image and entity labels as inputs. We present ablations on the coverage boost factors (W^r_{we} and W^p_{obj}) and entity-label modeling (type-only versus surface-form names). Informativeness of captions (Cov^r_{we} and Cov^p_{obj}) increases as coverage boost factors are increased (correlations highlighed in yellow and green).

Table 2 shows the performance of our model. Using both image and input labels improves performance on all metrics, compared to the baselines in Table 1. This indicates the effectiveness of our multi-encoder, multi-gated decoder architecture (§ 3.2.1) in generating captions that are both informative and fluent. Moreover, boosting the weight for web entity labels (W^r_{we}) and object labels (W^p_{obj}) improves informativeness for each of these types, see patterns highlighted in Table 2 for Cov^r_{we} and Cov^p_{obj}, respectively.[4] In terms of label modeling (type-only versus surface-form + type), the CIDEr score tends to be significantly higher when modeling surface-forms directly, while coverage metrics favor the type-only setting. We attribute the former to the fact that the ground-truth captions are not very sensitive to label accuracy, and the latter to the fact that it is easier for the model to learn and generalize using the closed set of token types (approx. 4500 types).

We mention here that evaluating the performance of the image labeler models used (for object labels, entity recognition) is outside the scope of this work. Their (possibly noisy) outputs are assumed given as input, and our evaluation measures the extent to which various image captioning models are capable of incorporating this information.

4.2 Qualitative Results

To get a better intuition on the behavior of our models, we compare output captions of different model variants using two sample images (Fig. 6). The baseline model without any input labels tends to produce generic-sounding captions (i.e., refer to the people in the image simply as 'person'). When we supply web entity types as inputs, the outputs become more informative as evident from the use of output types, e.g. ⟨ARTIST⟩, which in turn is postprocessed to match the web entity label "eric clapton". Furthermore, increasing the Coverage Boost Factor to 1.5 (i.e., both W^r_{we} and W^p_{obj} set to 1.5) results in more informative captions that add previously-missing aspects such as "concert" and "geffen playhouse".

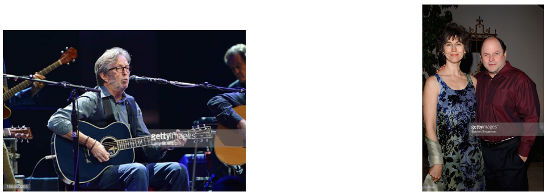
Similar trends are seen with direct modeling of the fine-grained labels for web entities. While successfully adding additional information under the Coverage Boost 1.5 condition for the first image ("the crossroads guitar festival"), we observe an error pattern this model exhibits, namely, the presence of ill-formed named entities in the output ("daena e. title daena playtitle", indicated in red in Fig. 6). The human evaluation results show that this model configuration performs worse compared to the one using entity types only, which are both easier to learn by the model and guaranteed to preserve the full name of the entity as copied during postprocessing from the input labels.

Please see Fig. 8 for more test images and output captions. Our model generates captions that fluently incorporate fine-grained entity mentions that are provided as input labels, e.g. "tromsø" (city in northern Norway), "basmati" (type of rice), "aruba" (island in the Caribbean Sea) and "kia ceed" (model of car). In the cases where such specific details are not available as inputs, the model uses a generic term to describe the scene, e.g. "**musician** playing the saxophone on stage".

4.3 Human Evaluation

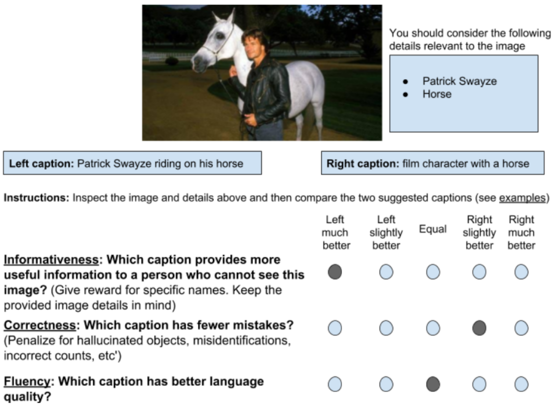
We conducted a human evaluation study to determine whether the gains on label coverage scores (Table 2) correlate with accuracy as judged by humans. Hence we used the models with high coverages of object labels and web entities in Table 2, which correspond to 1.5 Coverage Boost. Each of our proposed models was independently compared in a side-by-side evaluation (randomized order) against the same baseline model on three different dimensions: Informativeness, Correctness and Fluency. We used the model by Sharma et al. (2018) as the baseline model because the goal was to measure gains obtained from using the input labels, and this baseline performed best (in terms of

[4] Note that scaling weights at 2.0 or larger lead to repetition of input labels in the captions, resulting in fluency degradation without additional gains in informativeness.



Labels (as additional inputs)		
Web Entity labels (WE) / Types	eric clapton / ⟨ARTIST⟩ musician / ⟨PROFESSION⟩ crossroads guitar festival 2013 / ⟨ALBUM⟩	jason alexander / ⟨ACTOR⟩ daena e. title / ⟨PERSON⟩ geffen playhouse / ⟨THEATER⟩
Object labels	guitarist, music artist, performance, stage, concert	lady, fashion, formal wear
Model Variants	***Output***	***Output***
Image only	person performs on stage	people arrive at the premiere
Image + WE Types	⟨ARTIST⟩ performs on stage	⟨ACTOR⟩ and ⟨PERSON⟩ attend the opening night
(Above) + Postprocessing	eric clapton performs on stage	jason alexander and daena e. title attend the opening night
Image + WE Types + 1.5 Boost	⟨ARTIST⟩ performs live during a concert	⟨ACTOR⟩ and ⟨PERSON⟩ attend the premiere at ⟨THEATER⟩
(Above) + Postprocessing	eric clapton performs live during a concert	**jason alexander and daena e. title attend the premiere at geffen playhouse**
Image + WE	eric clapton performs live during a concert	actor jason alexander and wife daena title arrive at the geffen playhouse premiere
Image + WE + 1.5 Boost	**eric clapton performs on stage during the crossroads guitar festival**	jason alexander and daena e. title daena playtitle attend the premiere at geffen playhouse

Figure 6: Sample outputs for various model configurations for two images and their additional label inputs. In both cases, we notice that boosting the coverage at inference time leads to more informative captions without a loss in fluency. Note: Object labels are provided as inputs to the model in all cases except the baseline *Image only*.



You should consider the following details relevant to the image

- Patrick Swayze
- Horse

Left caption: Patrick Swayze riding on his horse

Right caption: film character with a horse

Instructions: Inspect the image and details above and then compare the two suggested captions (see examples)

	Left much better	Left slightly better	Equal	Right slightly better	Right much better
Informativeness: Which caption provides more useful information to a person who cannot see this image? (Give reward for specific names. Keep the provided image details in mind)	●	○	○	○	○
Correctness: Which caption has fewer mistakes? (Penalize for hallucinated objects, misidentifications, incorrect counts, etc')	○	○	○	●	○
Fluency: Which caption has better language quality?	○	○	●	○	○

Figure 7: Interface for the human evaluation.

CIDEr score) amongst baselines not using input labels (Table 1).

The evaluation setup and a description of each of the evaluation dimensions is given in Fig. 7. Note that the web entities that are fed to the model were also provided to the raters as reference (to help with fine-grained identification). In this example, the left caption is judged higher on the Informativeness scale because it correctly identifies the person in the image, but it is rated lower on the Correctness dimension due to the incorrect action ("riding"); both captions are judged as equally fluent.

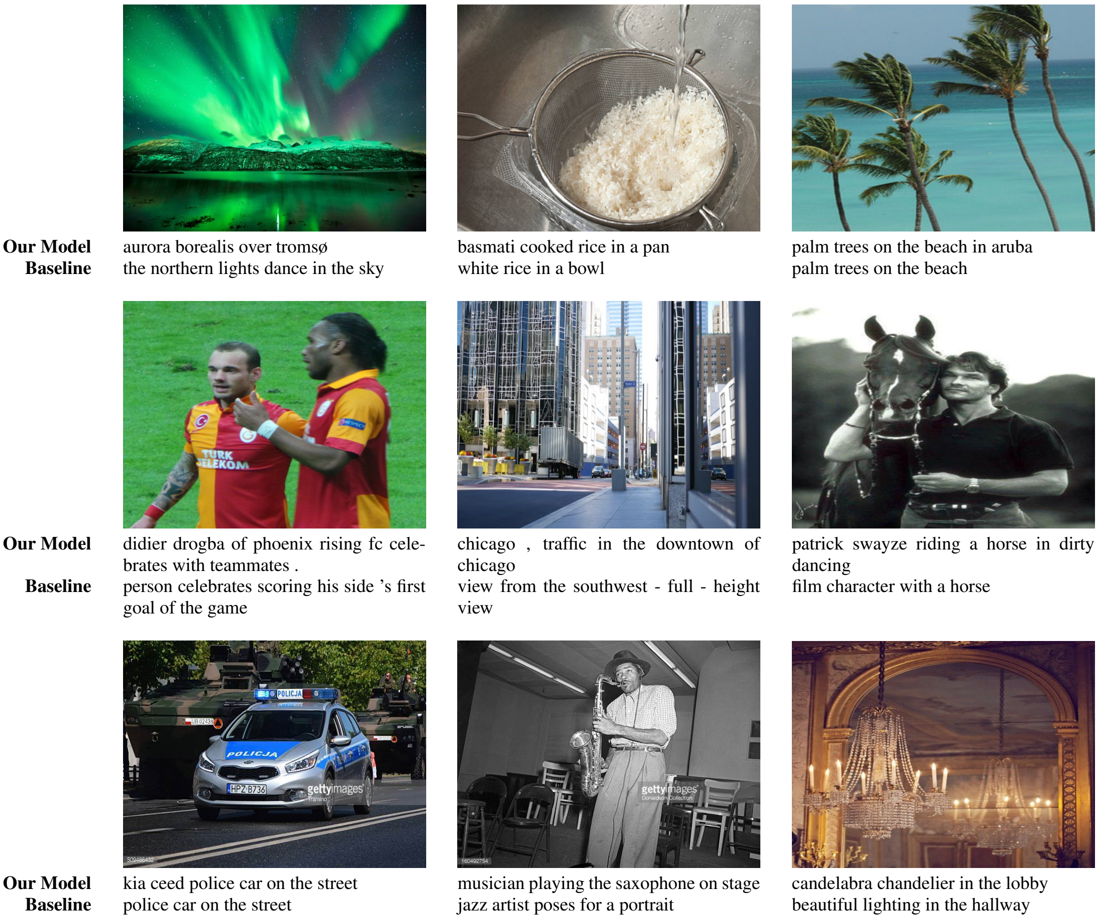
In each evaluation, three raters evaluate a 2K random sample batch from the test set. The human ratings were mapped to the corresponding scores using the following scheme:

The baseline caption is much better	-1.0
The baseline caption is slightly better	-0.5
The two captions seem equal	0
Our model's caption is slightly better	+0.5
Our model's caption is much better	+1.0

Table 3 reports the improvements in human evaluations using our setup against the baseline captions. We observe that the gains in coverage scores in Table 2 are now reflected in the human judgements, with the best model (using labels, type-only) judged as 24.33% more informative and 7.79% more correct, with virtually no loss in fluency. Furthermore, these results validate the claim from Sharma et al. (2018) that generating captions containing fine-grained entities from image pixels only (without additional fine-grained labels) leads to inferior performance in both informativeness (-7.91%) and correctness (-7.33%) (second row in Table 3).

L	T	W_{we}^{r}	W_{obj}^{p}	Info'	Correct'	Fluency
Sharma et al. (2018)				0.00%	0.00%	0.00%
No	-	-	-	-7.91%	-7.33%	0.11%
Lu et al. (2018) w/ T				7.45%	2.60%	**2.47%**
Yes	No	1.5	1.5	16.18%	7.94%	-0.06%
Yes	**Yes**	**1.5**	**1.5**	**24.33%**	**7.79%**	-0.87%

Table 3: Side-by-side human evaluation results (first entry is the system used in all the comparisons). First column (L) indicates if entity-labels are used. Second column (T) indicates if entity type is used instead of the surface-form.



Our Model aurora borealis over tromsø
Baseline the northern lights dance in the sky

Our Model basmati cooked rice in a pan
Baseline white rice in a bowl

Our Model palm trees on the beach in aruba
Baseline palm trees on the beach

Our Model didier drogba of phoenix rising fc celebrates with teammates .
Baseline person celebrates scoring his side 's first goal of the game

Our Model chicago , traffic in the downtown of chicago
Baseline view from the southwest - full - height view

Our Model patrick swayze riding a horse in dirty dancing
Baseline film character with a horse

Our Model kia ceed police car on the street
Baseline police car on the street

Our Model musician playing the saxophone on stage
Baseline jazz artist poses for a portrait

Our Model candelabra chandelier in the lobby
Baseline beautiful lighting in the hallway

Figure 8: Qualitative results comparing baseline captions (Sharma et al., 2018) with our model that use web entity types and Coverage Boost Factor of 1.5 (i.e., both W^r_{we} and W^p_{obj} set to 1.5).

5 Conclusion

We present an image captioning model that combines image features with fine-grained entities and object labels, and learns to produce fluent and informative image captions. Additionally, our model learns to estimate entity and object label coverage, which can be used at inference time to further boost the generated caption's informativeness without hurting its fluency.

Our human evaluations validate that training a model against ground-truth captions containing fine-grained labels (but without the additional help for fine-grained label identification), leads to models that produce captions of inferior quality. The results indicate that the best configuration is one in which fine-grained labels are precomputed by upstream models, and handled by the captioning model as types, with additional significant benefits gained by boosting the coverage of the fine-grained labels via a coverage control mechanism.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and VQA. In *CVPR*.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2014. Long-term recurrent convolutional networks for visual recognition and description. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John Platt, et al. 2015. From captions to visual concepts and back. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *JAIR*.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. 2015. Unifying visual-semantic embeddings with multimodal neural language models. *Transactions of the Association for Computational Linguistics*.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.

Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. 2017. Optimization of image description metrics using policy gradient methods. In *International Conference on Computer Vision (ICCV)*.

Di Lu, Spencer Whitehead, Lifu Huang, Heng Ji, and Shih-Fu Chang. 2018. Entity-aware image caption generation. *arXiv preprint arXiv:1804.07889*.

Tong Niu and Mohit Bansal. 2018. Polite dialogue generation without parallel data. *arXiv preprint arXiv:1805.03162*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *CoRR*, abs/1511.06732.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2556–2565.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

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

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

Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proc. of the 32nd International Conference on Machine Learning (ICML)*.

Z. Yang, Y. Yuan, Y. Wu, R. Salakhutdinov, and W. W. Cohen. 2016. Review networks for caption generation. In *NIPS*.

Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. 2017. Boosting image captioning with attributes. In *IEEE International Conference on Computer Vision, ICCV*, pages 22–29.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659.