Abstract

We propose an end-to-end model which generates captions for images embedded in news articles. News images present two key challenges: they rely on real-world knowledge, especially about named entities; and they typically have linguistically rich captions that include uncommon words. We address the first challenge by associating words in the caption with faces and objects in the image, via a multi-modal, multi-head attention mechanism. We tackle the second challenge with a state-of-the-art transformer language model that uses byte-pair-encoding to generate captions as a sequence of word parts. On the GoodNews dataset [3], our model outperforms the previous state of the art by a factor of four in CIDEr score (13 → 54). This performance gain comes from a unique combination of language models, word representation, image embeddings, face embeddings, object embeddings, and improvements in neural network design. We also introduce the NYTimes800k dataset which is 70% larger than GoodNews, has higher article quality, and includes the locations of images within articles as an additional contextual cue.

1. Introduction

The Internet is home to a large number of images, many of which lack useful captions. While a growing body of work has developed the capacity to narrate the contents of generic images [10, 49, 12, 19, 39, 30, 1, 6], these techniques still have two important weaknesses. The first weakness is in world knowledge. Most captioning systems are aware of generic object categories but unaware of names and places. Also generated captions are often inconsistent with commonsense knowledge. The second weakness is in linguistic expressiveness. The community has observed that generated captions tend to be shorter and less diverse than human-written captions [50, 24]. Most captioning systems rely on a fixed vocabulary and cannot correctly place or spell new or rare words.

News image captioning is an interesting case study for tackling these two challenges. Not only do news captions describe specific people, organizations and places, but the associated news articles also provide rich contextual information. The language used in news is evolving, with both the vocabulary and style changing over time. Thus news captioning approaches need to adapt to new words and concepts that emerge over a longer period of time (e.g. walkman in the 1990s or mp3 player in the 2000s). Existing approaches [44, 37, 3] rely on text extraction or template filling, which prevents the results from being linguistically richer than the template generator and are error-prone due to the difficulty in ranking entities for gap filling. Successful strategies for news image captioning can be generalized to images from domains with other types of rich context, such as web pages, social media posts, and user comments.

We propose an end-to-end model for news image captioning with a novel combination of sequence-to-sequence neural networks, language representation learning, and vision subsystems. In particular, we address the knowledge gap by computing multi-head attention on the words in the article, along with faces and objects that are extracted from the image. We address the linguistic gap with a flexible byte-pair-encoding that can generate unseen words. We
use dynamic convolutions and mix different linguistic representation layers to make the neural network representation richer. We also propose a new dataset, NYTimes800k, that is 70% larger than GoodNews [3] and has higher-quality articles along with additional image location information. We observe a performance gain of 6.8× in BLEU-4 (0.89 → 6.05) and 4.1× in CIDEr (13.1 → 53.8) compared to previous work [3]. On both datasets we observe consistent gains for each new component in our language, vision, and knowledge-aware system. We also find that our model generates names not seen during training, resulting in linguistically richer captions, which are closer in length (mean 15 words) to the ground truth (mean 18 words) than the previous state of the art (mean 10 words).

Our main contributions include:

1. A new captioning model that incorporates transformers, an attention-centric language model, byte-pair encoding, and attention over four different modalities (text, images, faces, and objects).

2. Significant performance gains over all metrics, with associated ablation studies quantifying the contributions of our main modeling components using BLEU-4, CIDEr, precision & recall of named entities and rare proper nouns, and linguistic quality metrics.

3. NYTimes800k, the largest news image captioning dataset to date, containing 445K articles and 793K images with captions from The New York Times spanning 14 years. NYTimes800k builds and improves upon the recently proposed GoodNews dataset [3]. It has 70% more articles and includes image locations within the article text. The dataset, code, and pretrained models are available on GitHub.

2. Related Works

A popular design choice for image captioning systems involves using a convolutional neural network (CNN) as the image encoder and a recurrent neural network (RNN) with a closed vocabulary as a decoder [19, 10, 49]. Attention over image patches using a multilayer perception was introduced in “Show, Attend and Tell” [53]. Further extensions include having the option to not attend to any image region [30] using a bottom-up approach to propose a region to attend to [1], and attending specifically to object regions [51] and visual concepts [55, 25, 51] identified in the image.

News image captioning includes the article text as input and focuses on the types of images used in news articles. A key challenge here is to generate correct entity names, especially rare ones. Existing approaches include extractive methods that use n-gram models to combine existing phrases [13] or simply retrieving the most representative sentence [44] in the article. Ramisa et al. [37] built an end-to-end LSTM decoder that takes both the article and image as inputs, but the model was still unable to produce names that were not seen during training.

To overcome the limitation of a fixed-size vocabulary, template-based methods have been proposed. An LSTM first generates a template sentence with placeholders for named entities, e.g. “PERSON speaks at BUILDING in DATE.” [3]. Afterwards the best candidate for each placeholder is chosen via a knowledge graph of entity combinations [29], or via sentence similarity [3]. One key difference between our proposed model and previous approaches [3, 29] is that our model can generate a caption with named entities directly without using an intermediate template.

One tool that has seen recent successes in many natural language processing tasks are transformer networks. Transformers have been shown to consistently outperform RNNs in language modeling [36], story generation [11], summarization [43], and machine translation [4]. In particular, transformer-based models such as BERT [9], XLM [22], XLNet [54], RoBERTa [27], and ALBERT [23] are able to produce high level text representations suitable for transfer learning. Furthermore, using byte-pair encoding (BPE) [41] to represent uncommon words as a sequence of subword units enables transformers to function in an open vocabulary setting. To date the only image captioning work that uses BPE is [57], but they did not use it for rare named entities as these were removed during pre-processing. In contrast we explicitly examine BPE for generating rare names and compare it to template-based methods.

Transformers have been shown to yield competitive results in generating generic MS COCO captions [58, 25]. Zhao et al. [57] have gone further and trained transformers to produce some named entities in the Conceptual Captions dataset [42]. However, the authors used web-entity labels, extracted using Google Cloud Vision API, as inputs to the model. In our work, we do not explicitly give the model a list of entities to appear in the caption. Instead our model automatically identifies relevant entities from the provided news article.

3. The Transform and Tell Model

Our model consists of a set of pretrained encoders and a decoder, as illustrated in Figure 2. The encoders (Section 3.1) generate high-level vector representations of the images, faces, objects, and article text. The decoder (Section 3.2) attends over these representations to generate a caption at the sub-word level.

3.1. Encoders

Image Encoder: An overall image representation is obtained from a ResNet-152 [17] model pre-trained on Im-
Figure 2: Overview of the Transform and Tell model. Left: Decoder with four transformer blocks; Right: Encoder for article, image, faces, and objects. The decoder takes as input embeddings of byte-pair tokens (blue circles at the bottom). For example, the input in the final time step, 14980, represents “arsh” in “Varshini”) from the previous time step. The grey arrows show the convolutions in the final time step in each block. Colored arrows show attention to the four domains on the right: article text (green lines), image patches (yellow lines), faces (orange lines), and objects (blue lines). The final decoder outputs are byte-pair tokens, which are combined up to form whole words and punctuations.

ageNet. We use the output of the final block before the pooling layer as the image representation. This is a set of 49 different vectors $x^I_i \in \mathbb{R}^{2048}$ where each vector corresponds to a separate image patch after the image is divided into equally-sized 7 by 7 patches. This gives us the set $X^I = \{x^I_i \in \mathbb{R}^{D^I} \}_{i=1}^{M^I}$, where $D^I = 2048$ and $M^I = 49$ for ResNet-152. Using this representation allows the decoder to attend to different regions of the image, which is known to improve performance in other image captioning tasks [53] and has been widely adopted.

**Face Encoder:** We use MTCNN [56] to detect face bounding boxes in the image. We then select up to four faces since the majority of the captions contain at most four people’s names (see Section 4). A vector representation of each face is obtained by passing the bounding boxes to FaceNet [40], which was pre-trained on the VGGFace2 dataset [5]. The resulting set of face vectors for each image is $X^F = \{x^F_i \in \mathbb{R}^{D^F} \}_{i=1}^{M^F}$, where $D^F = 512$ for FaceNet and $M^F$ is the number of faces. If there are no faces in the image, $X^F$ is an empty set.

Even though the faces are extracted from the image, it is useful to consider them as a separate input domain. This is because a specialized face embedding model is tuned for identifying people and thus can help the decoder to generate more accurate named entities.

**Object Encoder:** We use YOLOv3 [38] to detect object bounding boxes in the image. We filter out objects with a confidence less than 0.3 and select up to 64 objects with the highest confidence scores to feed through a ResNet-152 pretrained on ImageNet. In contrast to the image encoder, we take the output after the pooling layer as the representation for each object. This gives us a set of object vectors $X^O = \{x^O_i \in \mathbb{R}^{D^O} \}_{i=1}^{M^O}$, where $D^O = 2048$ for ResNet-152 and $M^O$ is the number of objects.

**Article Encoder:** To encode the article text we use RoBERTa [27], a recent improvement over the popular BERT [9] model. RoBERTa is a pretrained language representation model providing contextual embeddings for text. It consists of 24 layers of bidirectional transformer blocks.

Unlike GloVe [35] and word2vec [31] embeddings, where each word has exactly one representation, the bidirectionality and the attention mechanism in the transformer allow a word to have different vector representations depending on the surrounding context.

The largest GloVe model has a vocabulary size of 1.2 million. Although this is large, many rare names will still get mapped to the unknown token. In contrast, RoBERTa uses BPE [41, 36] which can encode any word made from Unicode characters. In BPE, each word is first broken down into a sequence of bytes. Common byte sequences are then merged using a greedy algorithm. Following [36], our vocabulary consists of 50K most common byte sequences.

Inspired by Tenney et al. [46] who showed that different layers in BERT represent different steps in the tradi-
tional NLP pipeline, we mix the RoBERTa layers to obtain a richer representation. Given an input of length $M^T$, the pretrained RoBERTa encoder will return 25 sequences of embeddings, $G = \{g_{li} \in \mathbb{R}^{2048} : l \in \{0, 1, \ldots, 24\}, i \in \{1, 2, \ldots, M^T\}\}$. This includes the initial uncontextualized embeddings and the output of each of the 24 transformer layers. We take a weighted sum across all layers to obtain the article embedding $x_i^A$:

$$x_i^A = \sum_{\ell=0}^{24} \alpha_{\ell} g_{\ell i} \quad (1)$$

where $\alpha_{\ell}$ are learnable weights.

Thus our RoBERTa encoder produces the set of token embeddings $X^A = \{x_i^A \in \mathbb{R}^{D^T} \}_{i=1}^{M^T}$, where $D^T = 1024$ in RoBERTa.

### 3.2. Decoder

The decoder is a function that generates caption tokens sequentially. At time step $t$, it takes as input: the embedding of the token generated in the previous step, $z_{t-1} \in \mathbb{R}^{D_E}$ where $D_E$ is the hidden size; embeddings of all other previously generated tokens $Z_{0:t-1} = \{z_{00}, z_{01}, \ldots, z_{t-1}\}$; and the context embeddings $X^I, X^A, X^F$, and $X^O$ from the encoders. These inputs are then fed through $L$ transformer blocks:

$$z_{1t} = \text{Block}_1(z_{0t}, Z_{0:t-1}, X^I, X^A, X^F, X^O) \quad (2)$$
$$z_{2t} = \text{Block}_2(z_{1t}, Z_{1:t-1}, X^I, X^A, X^F, X^O) \quad (3)$$
$$\vdots$$

$$z_{Lt} = \text{Block}_L(z_{L-1:t}, Z_{L-1:t}, X^I, X^A, X^F, X^O) \quad (4)$$

where $z_{Lt}$ is the output of the $t$th transformer block at time step $t$. The final block’s output $z_{Lt}$ is used to estimate $p(y_t)$, the probability of generating the $t$th token in the vocabulary via adaptive softmax [16]:

$$p(y_t) = \text{AdaptiveSoftmax}(z_{Lt}) \quad (5)$$

By dividing the vocabulary into three clusters based on frequency—5K, 15K, and 30K—adaptive softmax makes training more efficient since most of the time, the decoder only needs to compute the softmax over the first cluster containing the 5,000 most common tokens.

In the following two subsections, we will describe the transformer block in detail. In each block, the conditioning on past tokens is achieved using dynamic convolutions, and the conditioning on the contexts is achieved using multi-head attention.

**Dynamic Convolutions:** Introduced by Wu et al. [52], the goal of dynamic convolution is to provide a more efficient alternative to self-attention [47] when attending to past tokens. At block $\ell + 1$ and time step $t$, we have the input $z_{t\ell} \in \mathbb{R}^{D_E}$. Given kernel size $K$ and $H$ attention heads, for each head $h \in \{1, 2, \ldots, H\}$, we first project the current and last $K - 1$ steps using a feedforward layer to obtain $z'_{t\ell j} \in \mathbb{R}^{D_E/K}$:

$$z'_{t\ell j} = \text{GLU}(W_{\ell h}^z z_{t\ell j} + b_{\ell h}^z) \quad (6)$$

for $j \in \{t - K + 1, t - K + 2, \ldots, t\}$. Here GLU is the gated linear unit activation function [7]. The output of each head’s dynamic convolution is the weighted sum of these projected values:

$$z_{t\ell} = \sum_{j=t-K+1}^{t} \gamma_{\ell hj} z'_{t\ell j} \quad (7)$$

where the weight $\gamma_{\ell hj}$ is a linear projection of the input (hence the term “dynamic”), followed by a softmax over the kernel window:

$$\gamma_{\ell hj} = \text{Softmax} \left( (W_h^\gamma)^T z'_{t\ell j} \right) \quad (8)$$

The overall output is the concatenation of all the head outputs, followed by a feedforward with a residual connection and layer normalization [2], which does a z-score normalization across the feature dimension (instead of the batch dimension as in batch normalization [18]):

$$z_{tt} = [\hat{z}_{tt1}, \hat{z}_{tt2}, \ldots, \hat{z}_{ttM}] \quad (9)$$
$$d_{tt} = \text{LayerNorm} \left( z_{tt} + W_{t}^z \hat{z}_{tt} + b_{t}^z \right) \quad (10)$$

The output $d_{tt}$ can now be used to attend over the context embeddings.

**Multi-Head Attention:** The multi-head attention mechanism [47] has been the standard method to attend over encoder outputs in transformers. In our setting, we need to attend over four context domains—images, text, faces, and objects. As an example, we will go over the image attention module, which consists of $H$ heads. Each head $h$ first does a linear projection of $d_{tt}$ and the image embeddings $X^I$ into a query $q_h^I \in \mathbb{R}^{D^E/H}$, a set of keys $K_h^I = \{k_{hi}^I \in \mathbb{R}^{D^E/H} \}_{i=1}^{M^I}$, and the corresponding values $V_h^I = \{v_{hi}^I \in \mathbb{R}^{D^E/H} \}_{i=1}^{M^I}$:

$$q_h^I = W_h^{IQ} d_{tt} \quad (11)$$
$$k_{hi}^I = W_h^{IK} x_i^I \quad \forall i \in \{1, 2, \ldots, M^I\} \quad (12)$$
$$v_{hi}^I = W_h^{IV} x_i^I \quad \forall i \in \{1, 2, \ldots, M^I\} \quad (13)$$

Then the attended image for each head is the weighted sum of the values, where the weights are obtained from the dot product between the query and key:

$$\lambda_{hi}^I = \text{Softmax} \left( K_h^I q_{hi}^I \right) \quad (14)$$

$$z_{th}^I = \sum_{i=1}^{M^I} \lambda_{hi}^I v_{hi}^I \quad (15)$$

13038
The attention from each head is then concatenated into $x_{i,t}^I \in \mathbb{R}^{D_E}$:

$$x_{i,t}^I = [x_{i,t1}^I, x_{i,t2}^I, ..., x_{i,THt}^I]$$ (16)

and the overall image attention $x_{i,t}^E \in \mathbb{R}^{D_E}$ is obtained after adding a residual connection and layer normalization:

$$x_{i,t}^E = \text{LayerNorm}(d_{it} + x_{i,t}^I)$$ (17)

We use the same multi-head attention mechanism (with different weight matrices) to obtain the attended article $x_{i,t}^A$, faces $x_{i,t}^F$, and objects $x_{i,t}^O$. These four are finally concatenated and fed through a feedforward layer:

$$x_{i,t}^C = [x_{i,t}^A, x_{i,t}^F, x_{i,t}^O]$$ (18)

$$x_{i,t}^{C'} = W_C x_{i,t}^C + b_C$$ (19)

$$x_{i,t}^{C''} = \text{ReLU}(W_C' x_{i,t}^{C'} + b_C'')$$ (20)

$$z_{i+1,t} = \text{LayerNorm}(x_{i,t}^{C''} + W_F x_{i,t} + b_F')$$ (21)

The final output $z_{i+1,t} \in \mathbb{R}^{D_E}$ is used as the input to the next transformer block.

4. News Image Captioning Datasets

We describe two datasets that contain news articles, images, and captions. The first dataset, GoodNews, was recently proposed in Biten et al. [3], while the second dataset, NYTimes800k, is our contribution.

**GoodNews**: The GoodNews dataset was previously the largest dataset for news image captioning [3]. Each example in the dataset is a triplet containing an article, an image, and a caption. Since only the article text, captions, and image URLs are publicly released, the images need to be downloaded from the original source. Out of the 466K image URLs provided by [3], we were able to download 463K images, or 99.2% of the original dataset—the remaining are broken links.

We use this 99.2% sample of GoodNews and the train-validation-test split provided by [3]. There are 421K training, 18K validation, and 23K test captions. Note that this split was performed at the level of captions, so it is possible for a training and test caption to share the same article text (since articles have multiple images).

We observe several issues with GoodNews that may limit a system’s ability to generate high-quality captions. Many of the articles in GoodNews are partially extracted because the generic article extraction library failed to recognize some of the HTML tags specific to The New York Times. Importantly, the missing text often included the first few paragraphs which frequently contain important information for captioning images. In addition GoodNews contains some non-English articles and captioned images from the recommendation sidebar which are not related to the main article.

**NYTimes800k**: The aforementioned issues motivated us to construct NYTimes800k, a 70% larger and more complete dataset of New York Times articles, images, and captions. We used The New York Times public API\(^2\) for the data collection and developed a custom parser to resolve the missing text issue in GoodNews. The average article in NYTimes800k is 963 words long, whereas the average article in GoodNews is 451 words long. Our parser also ensures that NYTimes800k only contains English articles and images that are part of the main article. Finally, we also collect information about where an image is located in the corresponding article. Most news articles have one image at the top that relates to the key topic. However 39% of the articles have at least one more image somewhere in the middle of text. The image placement and the text surrounding the image is important information for captioning as we will show in our evaluations. Table 1 presents a comparison between GoodNews and NYTimes800k.

![Table 1: Summary of news captioning datasets](https://developer.nytimes.com/apis)

<table>
<thead>
<tr>
<th></th>
<th>GoodNews</th>
<th>NYTimes800k</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of articles</td>
<td>257 033</td>
<td>444 914</td>
</tr>
<tr>
<td>Number of images</td>
<td>462 642</td>
<td>792 971</td>
</tr>
<tr>
<td>Average article length</td>
<td>451</td>
<td>974</td>
</tr>
<tr>
<td>Average caption length</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>Collection start month</td>
<td>Jan 10</td>
<td>Mar 05</td>
</tr>
<tr>
<td>Collection end month</td>
<td>Mar 18</td>
<td>Aug 19</td>
</tr>
</tbody>
</table>

We observe several issues with GoodNews that may limit a system’s ability to generate high-quality captions. Many of the articles in GoodNews are partially extracted because the generic article extraction library failed to recognize some of the HTML tags specific to The New York Times. Importantly, the missing text often included the first few paragraphs which frequently contain important information for captioning images. In addition GoodNews contains some non-English articles and captioned images from the recommendation sidebar which are not related to the main article.

2https://developer.nytimes.com/apis
Figure 3: Co-occurrence of faces and people’s names in NYTimes800k training data. The blue bars count how many images containing a certain number of faces. The orange bars count how many captions containing a certain number of people’s names.

We split the training, validation, and test sets according to time, as shown in Table 2. Compared to the random split used in GoodNews, splitting by time allows us to study the model performance on novel news events and new names, which might be important in a deployment scenario. Out of the 100K proper nouns in our test captions, 4% never appear in any training captions.

5. Experiments

This section describes settings for neural network learning, baselines and evaluation metrics, followed by a discussion of key results.

5.1. Training Details

Following Wu et al. [52], we set the hidden size $D^E$ to 1024; the number of heads $H$ to 16; and the number of transformer blocks $L$ to four with kernel sizes 3, 7, 15, and 31, respectively. For parameter optimization we use the adaptive gradient algorithm Adam [21] with the following parameter: $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-6}$. We warm up the learning rate in the first 5% of the training steps to $10^{-4}$, and decay it linearly afterwards. We apply $L_2$ regularization to all network weights with a weight decay of $10^{-5}$ and using the fix [28] that decouples the learning rate from the regularization parameter. We clip the gradient norm at 0.1. We use a maximum batch size of 16 and training is stopped after the model has seen 6.6 million examples. This is equivalent to 16 epochs on GoodNews and 9 epochs on NYTimes800k.

The training pipeline is written in PyTorch [34] using the AllenNLP framework [15]. The RoBERTa model and dynamic convolution code are adapted from fairseq [32]. Training is done with mixed precision to reduce the memory footprint and allow our full model to be trained on a single GPU. The full model takes 5 days to train on one Titan V GPU and has 200 million trainable parameters—see the supplementary material for the size of each model variant.

Table 2: NYTimes800k training, validation, and test splits

<table>
<thead>
<tr>
<th></th>
<th>Training</th>
<th>Validation</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of articles</td>
<td>433,561</td>
<td>2,978</td>
<td>8,375</td>
</tr>
<tr>
<td>Number of images</td>
<td>763,217</td>
<td>7,777</td>
<td>21,977</td>
</tr>
<tr>
<td>Start month</td>
<td>Mar 15</td>
<td>May 19</td>
<td>Jun 19</td>
</tr>
<tr>
<td>End month</td>
<td>Apr 19</td>
<td>May 19</td>
<td>Aug 19</td>
</tr>
</tbody>
</table>

5.2. Evaluation Metrics

We use BLEU-4 [33] and CIDEr [48] scores as they are standard for evaluating image captions. These are obtained using the COCO caption evaluation toolkit. The supplementary material additionally reports BLEU-1, BLEU-2, BLEU-3, ROUGE [26], and METEOR [8]. Note that CIDEr is particularly suited for evaluating news captioning models as it puts more weight than other metrics on uncommon words. In addition, we evaluate the precision and recall on named entities, people’s names, and rare proper names. Named entities are identified in both the ground-truth captions and the generated captions using SpaCy. We then count exact string matches between the ground truths and generated entities. For people’s names we restrict the set of named entities to those marked as PERSON by the SpaCy parser. Rare proper nouns are nouns that appear in a test caption but not in any training caption.

5.3. Baselines and Model Variants

We show two previous state-of-the-art models: Biten (Avg + CtxIns) and Biten (TBB + Attns) [3]. To provide a fair comparison we used the full caption results released by Biten et al. [3] and re-evaluated with our evaluation pipeline on a slightly smaller test set (a few test images are no longer available due to broken URLs). The final metrics are the same as originally reported if rounded to the nearest whole number.

We evaluate a few key modeling choices: the decoder type (LSTM vs Transformer), the text encoder type (GloVe vs RoBERTa vs weighted RoBERTa), and the additional context domains (location-aware, face attention, and object attention). The location-aware models select the 512 tokens surrounding the image instead of the first 512 tokens of the article. Note that all our models use BPE in the decoder with adaptive softmax. We ensure that the total number of trainable parameters for each model is within 7% of one another (148 million to 159 million), with the exception of face attention (171 million) and object attention (200 million) since the latter two have extra multi-head attention modules. The results reported over GoodNews are based on a model trained solely on GoodNews, using the original random split of [3] for easier comparison to previous work.

https://github.com/tylin/coco-caption
Table 3: Results on GoodNews (rows 1–10) and NYTimes800k (rows 11–19). We report BLEU-4, ROUGE, CIDEr, and precision (P) & recall (R) of named entities, people’s names, and rare proper nouns. Precision and recall are expressed as percentages. Rows 1–2 contain previous state-of-the-art results [3]. Rows 3–5 and 11–13 are ablation studies where we swap the Transformer with an LSTM and/or RoBERTa with GloVe. These models only have the image attention (IA). Rows 6 & 14 are our baseline RoBERTa transformer language model that only has the article text (and not the image) as inputs. Building on top of this, we first add attention over image patches (rows 7 & 15). We then take a weighted sum of the RoBERTa embeddings (rows 8 & 16) and attend to the text surrounding the image instead of the first 512 tokens of the article (row 17). Finally we add attention over faces (rows 9 & 18) and objects (rows 10 & 19) in the image.

<table>
<thead>
<tr>
<th></th>
<th>BLEU-4</th>
<th>ROUGE</th>
<th>CIDEr</th>
<th>Named entities</th>
<th>People’s names</th>
<th>Rare proper nouns</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td></td>
<td>P</td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>GoodNews</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1) Biten (Avg + CtxIns) [3]</td>
<td>0.89</td>
<td>12.2</td>
<td>13.1</td>
<td>8.23</td>
<td>6.06</td>
<td>9.38</td>
</tr>
<tr>
<td>(2) Biten (TBB + AttIns) [3]</td>
<td>0.76</td>
<td>12.2</td>
<td>12.7</td>
<td>8.87</td>
<td>5.64</td>
<td>11.9</td>
</tr>
<tr>
<td>(3) LSTM + GloVe + IA</td>
<td>1.97</td>
<td>13.6</td>
<td>13.9</td>
<td>10.7</td>
<td>7.09</td>
<td>9.07</td>
</tr>
<tr>
<td>(4) Transformer + GloVe + IA</td>
<td>3.48</td>
<td>17.0</td>
<td>25.2</td>
<td>14.3</td>
<td>11.1</td>
<td>14.5</td>
</tr>
<tr>
<td>(5) LSTM + RoBERTa + IA</td>
<td>3.45</td>
<td>17.0</td>
<td>28.6</td>
<td>15.5</td>
<td>12.0</td>
<td>16.4</td>
</tr>
<tr>
<td>(6) Transformer + RoBERTa</td>
<td>4.60</td>
<td>18.6</td>
<td>40.9</td>
<td>19.3</td>
<td>16.1</td>
<td>24.4</td>
</tr>
<tr>
<td>(7) + image attention</td>
<td>5.45</td>
<td>20.7</td>
<td>48.5</td>
<td>21.1</td>
<td>17.4</td>
<td>26.9</td>
</tr>
<tr>
<td>(8) + weighted RoBERTa</td>
<td>6.0</td>
<td>21.2</td>
<td>53.1</td>
<td>21.8</td>
<td>18.5</td>
<td>28.8</td>
</tr>
<tr>
<td>(9) + face attention</td>
<td>6.05</td>
<td>21.4</td>
<td>54.3</td>
<td>22.0</td>
<td>18.6</td>
<td>29.3</td>
</tr>
<tr>
<td>(10) + object attention</td>
<td>6.05</td>
<td>21.4</td>
<td>53.8</td>
<td>22.2</td>
<td>18.7</td>
<td>29.2</td>
</tr>
<tr>
<td>NYTimes800k</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(11) LSTM + GloVe + IA</td>
<td>1.77</td>
<td>13.1</td>
<td>12.1</td>
<td>10.2</td>
<td>7.24</td>
<td>8.83</td>
</tr>
<tr>
<td>(12) Transformer + GloVe + IA</td>
<td>2.75</td>
<td>15.9</td>
<td>20.3</td>
<td>13.2</td>
<td>10.8</td>
<td>13.2</td>
</tr>
<tr>
<td>(13) LSTM + RoBERTa + IA</td>
<td>3.29</td>
<td>16.1</td>
<td>24.9</td>
<td>15.1</td>
<td>12.9</td>
<td>17.7</td>
</tr>
<tr>
<td>(14) Transformer + RoBERTa</td>
<td>4.26</td>
<td>17.3</td>
<td>33.9</td>
<td>17.8</td>
<td>16.3</td>
<td>23.6</td>
</tr>
<tr>
<td>(15) + image attention</td>
<td>5.01</td>
<td>19.4</td>
<td>40.3</td>
<td>20.0</td>
<td>18.1</td>
<td>28.2</td>
</tr>
<tr>
<td>(16) + weighted RoBERTa</td>
<td>5.75</td>
<td>19.9</td>
<td>45.1</td>
<td>21.1</td>
<td>19.6</td>
<td>29.7</td>
</tr>
<tr>
<td>(17) + location-aware</td>
<td>6.36</td>
<td>21.4</td>
<td>52.8</td>
<td>24.0</td>
<td>21.9</td>
<td>35.4</td>
</tr>
<tr>
<td>(18) + face attention</td>
<td>6.26</td>
<td>21.5</td>
<td>53.9</td>
<td>24.2</td>
<td>22.1</td>
<td>36.5</td>
</tr>
<tr>
<td>(19) + object attention</td>
<td>6.30</td>
<td>21.7</td>
<td>54.4</td>
<td>24.6</td>
<td>22.2</td>
<td>37.3</td>
</tr>
</tbody>
</table>

5.4. Results and Discussion

Table 3 summarizes evaluation metrics on GoodNews and NYTimes800k, while Figure 4 compares generated captions from different model variants. Our full model (row 10) performs substantially better than the existing state of the art [3] across all evaluation metrics. On GoodNews, the full model yields a CIDEr score of 53.8, whereas the previous state of the art [3] achieved a CIDEr score of only 13.1.

Our most basic LSTM model (row 3) differs from Biten et al. [3] in that we use BPE in the caption decoder instead of template generation and filling. The slight improvement in CIDEr (from 13.1 to 13.9) shows that BPE offers a competitive end-to-end alternative to the template filling method. This justifies the use of BPE in the remaining experiments.

Models that encode articles using GloVe embeddings (rows 3–4 and 11–12) are unable to generate rare proper nouns, giving a precision and recall of 0. This is because the encoder skips words that are not part of the fixed GloVe vocabulary. This motivates the switch from GloVe to RoBERTa, which has an unbounded vocabulary. This switch shows a clear advantage in rare proper noun generation. On NYTimes800k, even the worst performing model that uses RoBERTa (row 13) achieves a precision of 7.47%, a recall of 9.50%, and a CIDEr gap of 12.8 points over the model without RoBERTa (row 11).

Another important modeling choice is the functional form of the caption decoder. We find that the Transformer architecture provides a substantial improvement over the LSTM with respect to all evaluation metrics. For example, when we swap the LSTM with a Transformer (from row 13 to 15), the CIDEr score on NYTimes800k jumps from 24.9 to 40.3.

Adding attention over faces improves both the recall and precision of people’s names. It has no significant effect on other entity types (see the supplementary material for more details).
**Japan Desperately Needs More Day Care Workers. New Mothers Need Not Apply.**

TOKYO — Ever since she was a young girl, all Erica Takato wanted to do was work with small children. A few weeks into her term, she requested time off for bed rest ordered by her doctor. 

Union officials and former teachers cite a major obstacle to the aspirations: 

<table>
<thead>
<tr>
<th>Ground-truth caption</th>
<th>A nursery school teacher showing a bug to his class.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer + RoBERTa</td>
<td>Ms. Takato, who was born in Japan, was forced out of the day care program because she was pregnant.</td>
</tr>
<tr>
<td>+ image attention</td>
<td>Ms. Takato with her son, Kishiko, and their children, from left, Kaiti, 3, and Kaitama, 3, at a day care center in Tokyo.</td>
</tr>
<tr>
<td>+ weighted RoBERTa</td>
<td>Ms. Takato, with her son, Shiro, and son, at home in Tokyo. Ms. Takato, who was pregnant, said she was “so frustrated and lost hope of being able to work.”</td>
</tr>
<tr>
<td>+ location-aware</td>
<td>A day care center in Tokyo.</td>
</tr>
<tr>
<td>+ face attention</td>
<td>A child care center in Tokyo. The government is eager to bring more women into the work force, and is trying to come up with enough child care for mothers.</td>
</tr>
<tr>
<td>+ object attention</td>
<td>A day care worker in Tokyo. The government is trying to bring more women into the work force, and the government is trying to come up with enough child care for mothers to go back to work.</td>
</tr>
</tbody>
</table>

Figure 4: An example article (left) and the corresponding news captions (right) from the NYTimes800k test set. The model with no access to the image makes a sensible but incorrect guess that the image is about Ms. Takato. Since the image appears in the middle of the article, only the location-aware models correctly state that the focus of the image is on a day care center.

a detailed breakdown. Importantly, people’s names are the most common entity type in news captions and so we also see an improvement in CIDEr. Attention over objects also improves performance on most metrics, especially on NYTimes800k. More broadly, this result suggests that introducing specialized vision models tuned to the common types of objects such as organizations (via logos or landmarks) is a promising future direction to improve the performance on news image captioning.

The location-aware models (rows 17–19) focus the article context using the image location in the article, information which is only available in our NYTimes800k dataset. This simple focusing of context offers a big improvement to CIDEr, from 45.1 (row 16) to 52.8 (row 17). This suggests a strong correspondence between an image and the closest text that can be easily exploited to generate better captions.

The supplementary material additionally reports three caption quality metrics: caption length, type-token ratio (TTR) [45], and Flesch reading ease (FRE) [14, 20]. TTR is the ratio of the number of unique words to the total number of words in a caption. The FRE takes into account the number of words and syllables and produces a score between 0 and 100, where higher means being easier to read. As measured by FRE, captions generated by our model exhibit a level of language complexity that is closer to the ground truths. Additionally, captions generated by our model are 15 words long on average, which is closer to the ground-truths (18 words) than those generated by the previous state of the art (10 words) [3].

### 6. Conclusion

In this paper, we have shown that by using a carefully selected novel combination of the latest techniques drawn from multiple sub-fields within machine learning, we are able to set a new SOTA for news image captioning. Our model can incorporate real-world knowledge about entities across different modalities and generate text with better linguistic diversity. The key modeling components are byte-pair encoding that can output any word, contextualized embeddings for article text, specialized face & object encoding, and transformer-based caption generation. This result provides a promising step for other image description tasks with contextual knowledge, such as web pages, social media feeds, or medical documents. Promising future directions include specialized visual models for a broader set of entities like countries and organizations, extending the image context from the current article to recent or linked articles, or designing similar techniques for other image and text domains.

### Acknowledgement

This research was supported in part by the Data to Decisions Cooperative Research Centre whose activities are funded by the Australian Commonwealth Government’s Cooperative Research Centres Programme. The research was also supported in part by the Australian Research Council through project number DP180101985. We thank NVIDIA for providing us with Titan V GPUs through their GPU Grant Program.
References


