Describing Multimedia Content using Attention-based Encoder–Decoder Networks

Kyunghyun Cho[°], Aaron Courville[°] and Yoshua Bengio[°⋆]

Abstract—Whereas deep neural networks were first mostly used for classification tasks, they are rapidly expanding in the realm of structured output problems, where the observed target is composed of multiple random variables that have a rich joint distribution, given the input. We focus in this paper on the case where the input also has a rich structure and the input and output structures are somehow related. We describe systems that learn to *attend* to different places in the input, for each element of the output, for a variety of tasks: machine translation, image caption generation, video clip description and speech recognition. All these systems are based on a shared set of building blocks: gated recurrent neural networks and convolutional neural networks, along with trained *attention mechanisms*. We report on experimental results with these systems, showing impressively good performance and the advantage of the attention mechanism.

I. INTRODUCTION

IN this paper we focus on the application of deep learning to structured output problems where the task is to map the input to an output that possesses its own structure. The task is therefore not only to map the input to the correct output (e.g. the classification task in object recognition), but also to model the structure within the output sequence.

A classic example of a structured output problem is machine translation: to automatically translate a sentence from the *source language* to the *target language*. To accomplish this task, not only does the system need to be concerned with capturing the semantic content of the source language sentence, but also with forming a coherent and grammatical sentence in the target language. In other words, given an input source sentence, we cannot choose the elements of the output (i.e. the individual words) independently: they have a complex joint distribution.

Structured output problems represent a large and important class of problems that include classic tasks such as speech recognition and many natural language processing problems (e.g. text summarization and paraphrase generation). As the range of capabilities of deep learning systems increases, less established forms of structured output problems, such as image caption generation and video description generation ([1] and references therein,) are being considered.

One important aspect of virtually all structured output tasks is that *the structure of the output is intimately related to the structure of the input*. A central challenge to these tasks is therefore the problem of *alignment*. At its most fundamental, the problem of alignment is the problem of how to relate sub-elements of the input to sub-elements of the output. Consider again our example of machine translation. In order to translate the source sentence into the target language we need to first decompose the source sentence into its constituent semantic parts. Then we need to map these semantic parts to their counterparts in the target language. Finally, we need to use these semantic parts to compose the sentence following the grammatical regularities of the target language. Each word or phrase of the target sentence can be aligned to a word or phrase in the source language.

In the case of image caption generation, it is often appropriate for the output sentence to accurately describe the spatial relationships between elements of the scene represented in the image. For this, we need to *align* the output words to spatial regions of the source image.

In this paper we focus on a general approach to the alignment problem known as the soft attention mechanism. Broadly, attention mechanisms are components of prediction systems that allow the system to sequentially focus on different subsets of the input. The selection of the subset is typically conditioned on the state of the system which is itself a function of the previously attended subsets.

Attention mechanisms are employed for two purposes. The first is to reduce the computational burden of processing high dimensional inputs by selecting to only process subsets of the input. The second is to allow the system to focus on distinct aspects of the input and thus improve its ability to extract the most relevant information for each piece of the output, thus yielding improvements in the quality of the generated outputs.

As the name suggests, soft attention mechanisms avoid a hard selection of which subsets of the input to attend and instead uses a soft weighting of the different subsets. Since all subset are processed, these mechanisms offer no computation advantage. Instead, the advantage brought by the soft-weighting is that it is readily amenable to efficient learning via gradient backpropagation.

In this paper, we present a review of the recent work in applying the soft attention to structured output tasks and speculate about the future course of this line of research. The soft-attention mechanism is part of a growing litterature on more flexible deep learning architectures that embed a certain amount of *distributed decision making*.

II. BACKGROUND: RECURRENT AND CONVOLUTIONAL NEURAL NETWORKS

A. Recurrent Neural Network

A recurrent neural network (RNN) is a neural network specialized at handling a variable-length input sequence $x = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ and optionally a corresponding variable-length

° Université de Montréal ⋆ CIFAR Senior Fellow

output sequence $y = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$, using an internal hidden state $\mathbf{h}$. The RNN sequentially reads each symbol $\mathbf{x}_t$ of the input sequence and updates its internal hidden state $\mathbf{h}_t$ according to

$$\mathbf{h}_t = \phi_\theta \left(\mathbf{h}_{t-1}, \mathbf{x}_t \right), \qquad (1)$$

where ϕ_θ is a nonlinear activation function parametrized by a set of parameters θ. When the target sequence is given, the RNN can be trained to sequentially make a prediction $\hat{\mathbf{y}}_t$ of the actual output $\mathbf{y}_t$ at each time step t:

$$\hat{\mathbf{y}}_t = g_\theta \left(\mathbf{h}_t, \mathbf{x}_t \right), \qquad (2)$$

where g_θ may be an arbitrary, parametric function that is learned jointly as a part of the whole network.

The recurrent activation function ϕ in Eq. (1) may be as simple as an affine transformation followed by an element-wise logistic function such that

$$\mathbf{h}_t = \tanh \left(\mathbf{U} \mathbf{h}_{t-1} + \mathbf{W} \mathbf{x}_t \right),$$

where $\mathbf{U}$ and $\mathbf{W}$ are the learned weight matrices.[1]

It has recently become more common to use more sophisticated recurrent activation functions, such as a long short-term memory (LSTM, [2]) or a gated recurrent unit (GRU, [3], [4]), to reduce the issue of vanishing gradient [5], [6]. Both LSTM and GRU avoid the vanishing gradient by introducing gating units that adaptively control the flow of information across time steps.

The activation of a GRU, for instance, is defined by

$$\mathbf{h}_t = \mathbf{u}_t \odot \tilde{\mathbf{h}}_t + (1 - \mathbf{u}_t) \odot \mathbf{h}_{t-1},$$

where $\odot$ is an element-wise multiplication, and the update gates $\mathbf{u}_t$ are

$$\mathbf{g}_t = \sigma \left(\mathbf{U}_u \mathbf{h}_{t-1} + \mathbf{W}_u \mathbf{x}_t \right).$$

The candidate hidden state $\tilde{\mathbf{h}}_t$ is computed by

$$\tilde{\mathbf{h}}_t = \tanh \left(\mathbf{U} \mathbf{h}_{t-1} + \mathbf{W} \left(\mathbf{r}_t \odot \mathbf{x}_t \right) \right),$$

where the reset gates $\mathbf{r}_t$ are computed by

$$\mathbf{r}_t = \sigma \left(\mathbf{U}_r \mathbf{h}_{t-1} + \mathbf{W}_r \mathbf{x}_t \right).$$

All the use cases of the RNN in the remaining of this paper use either the GRU or LSTM.

B. RNN-LM: Recurrent Neural Network Language Modeling

In the task of language modeling, we let a model learn the probability distribution over natural language sentences. In other words, given a model, we can compute the probability of a sentence $s = (w_1, w_2, \ldots, w_T)$ consisting of multiple words, i.e., $p(w_1, w_2, \ldots, w_T)$, where the sentence is T words long.

This task of language modeling is equivalent to the task of predicting the next word. This is clear by rewriting the sentence probability into

$$p(w_1, w_2, \ldots, w_T) = \prod_{t=1}^{T} p(w_t \mid w_{<t}), \qquad (3)$$

<hr>

[1] We omit biases to make the equations less cluttered.

where $w_{<t} = (w_1, \ldots, w_{t-1})$. Each conditional probability on the right-hand side corresponds to the predictive probability of the next word w_t given all the preceding words $(w_1, \ldots, w_{t-1})$.

A recurrent neural network (RNN) can, thus, be readily used for language modeling by letting it predict the next symbol at each time step t (RNN-LM, [7]). In other words, the RNN predicts the probability over the next word by

$$p(w_{t+1} = w | w_{\leq t}) = g_\theta^w \left(\mathbf{h}_t, \mathbf{w}_t \right), \qquad (4)$$

where g_θ^w returns the probability of the word w out of all possible words. The internal hidden state $\mathbf{h}_t$ summarizes all the preceding symbols $w_{\leq t} = (w_1, \ldots, w_t)$.

We can generate an exact sentence sample from an RNN-LM by iteratively sampling from the next word distribution $p(w_{t+1} | w_{\leq t})$ in Eq. (4). Instead of stochastic sampling, it is possible to approximately find a sentence sample that maximizes the probability $p(s)$ using, for instance, beam search [8], [9].

The RNN-LM described here can be extended to learn a *conditional* language model. In conditional language modeling, the task is to model the distribution over sentences given an additional input, or context. The context may be anything from an image and a video clip to a sentence in another language. Examples of textual outputs associated with these inputs by the *conditional RNN-LM* include respectively an image caption, a video description and a translation. In these cases, the transition function of the RNN will take as an additional input the context c such that

$$\mathbf{h}_t = \phi_\theta \left(\mathbf{h}_{t-1}, \mathbf{x}_t, c \right). \qquad (5)$$

Note the c at the end of the r.h.s. of the equation.

This conditional language model based on RNNs will be at the center of later sections.

C. Deep Convolutional Network

A convolutional neural network (CNN) is a special type of a more general feedforward neural network, or multilayer perceptron, that has been specifically designed to work well with two-dimensional images [10]. The CNN often consists of multiple convolutional layers followed by a few fully-connected layers.

At each convolutional layer, the input image of width n_i, height n_j and c color channels ($\mathbf{x} \in \mathbb{R}^{n_i \times n_y \times c}$) is first convolved with a set of local filters $\mathbf{f} \in \mathbb{R}^{n_i' \times n_y' \times c \times d}$. For each location/pixel (i, j) of $\mathbf{x}$, we get

$$\mathbf{z}_{i,j} = \sum_{i'=1}^{n_i'} \sum_{j'=1}^{n_j'} f \left(\mathbf{f}_{i',j'}^\top \mathbf{x}_{i+i',j+j'} \right), \qquad (6)$$

where $\mathbf{f}_{i',j'} \in \mathbb{R}^{c \times d}$, $\mathbf{x}_{i+i',j+j'} \in \mathbb{R}^c$ and $\mathbf{z}_{i,j} \in \mathbb{R}^d$. f is an element-wise nonlinear activation function.

The convolution in Eq. (6) is followed by local max-pooling:

$$\mathbf{h}_{i,j} = \max_{\substack{i' \in \{ri, \ldots, (r+1)i - 1\}, \\ j' \in \{rj, \ldots, (r+1)j - 1\}}} \mathbf{z}_{i',j'}, \qquad (7)$$

for all $i \in \{1, \ldots, n_i/r\}$ and $j \in \{1, \ldots, n_j/r\}$. r is the size of the neighborhood.

The pooling operation has two desirable properties. First, it reduces the dimensionality of a high-dimensional output of the convolutional layer. Furthermore, this spatial max-pooling summarizes the activation of the neighbouring feature activations, leading to the (local) translation invariance.

After a small number of convolutional layers, the final feature map from the last convolutional layer is flattened to form a vector representation $\mathbf{h}$ of the input image. This vector $\mathbf{h}$ is further fed through a small number of fully-connected nonlinear layers until the output.

Recently, the CNNs have been found to be excellent at the task of large-scale object recognition. For instance, the annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC) has a classification track where more than a million annotated images with 1,000 classes are provided as a training set. In this challenge, the CNN-based entries have been dominant since 2012 [11], [12], [13], [14].

D. Transfer Learning with Deep Convolutional Network

Once a deep CNN is trained on a large training set such that the one provided as a part of the ILVRC challenge, we can use any intermediate representation, such as the feature map from any convolutional layer or the vector representation from any subsequent fully-connected layers, of the whole network for tasks other than the original classification.

It has been observed that the use of these intermediate representation from the deep CNN as an image descriptor significantly boosts subsequent tasks such as object localization, object detection, fine-grained recognition, attribute detection and image retrieval (see, e.g., [15], [16].) Furthermore, more non-trivial tasks, such as image caption generation [17], [18], [19], [20], [21], have been found to benefit from using the image descriptors from a pre-trained deep CNN. In later sections, we will discuss in more detail how image representations from a pre-trained deep CNN can be used in these non-trivial tasks such as image caption generation [22] and video description generation [23].

III. ATTENTION-BASED MULTIMEDIA DESCRIPTION

Multimedia description generation is a general task in which a model generates a natural language description of a multimedia input such as speech, image and video as well as text in another language, if we take a more general view. This requires a model to capture the underlying, complex mapping between the spatio-temporal structures of the input and the complicated linguistic structures in the output. In this section, we describe a neural network based approach to this problem, based on the encoder–decoder framework with the recently proposed attention mechanism.

A. Encoder–Decoder Network

An encoder–decoder framework is a general framework based on neural networks that aims at handling the mapping between highly structured input and output. It was proposed recently in [24], [3], [25] in the context of machine translation, where the input and output are natural language sentences written in two different languages.

As the name suggests, a neural network based on this encoder–decoder framework consists of an encoder and a decoder. The encoder f_{enc} first reads the input data x into a continuous-space representation c:

$$c = f_{\text{enc}}(x), \tag{8}$$

The choice of f_{enc} largely depends on the type of input. When x is a two-dimensional image, a convolutional neural network (CNN) from Sec. II-D may be used. A recurrent neural network (RNN) in Sec. II-A is a natural choice when x is a sentence.

The decoder then generates the output y conditioned on the continuous-space representation, or context c of the input. This is equivalent to computing the conditional probability distribution of y given x:

$$p(Y|x) = f_{\text{dec}}(c). \tag{9}$$

Again, the choice of f_{dec} is made based on the type of the output. For instance, if y is an image or a pixel-wise image segmentation, a conditional restricted Boltzmann machine (CRBM) can be used [26]. When y is a natural language description of the input x, it is natural to use an RNN which is able to model natural languages, as described in Sec. II-B.
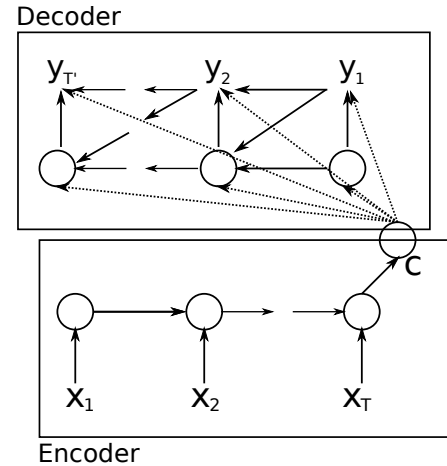

Fig. 1. Graphical illustration of the simplest form encoder-decoder model for machine translation from [3]. $x = (x_1, \ldots, x_T)$, $y = (y_1, \ldots, y_{T'})$ and c are respectively the input sentence, the output sentence and the continuous-space representation of the input sentence.

This encoder–decoder framework has been successfully used in [25], [3] for machine translation. In both work, an RNN was used as an encoder to summarize a source sentence (where the summary is the last hidden state $\mathbf{h}_T$ in Eq. (1)) from which a conditional RNN-LM from Sec. II-A decoded out the corresponding translation. See Fig. 1 for the graphical illustration.

In [19], [20], the authors used a pre-trained CNN as an encoder and a conditional RNN as a decoder to let model generate a natural language caption of images. Similarly, a simpler feedforward log-bilinear language model [27] was

used as a decoder in [21]. The authors of [28] applied the encoder–decoder framework to video description generation, where they used a pre-trained CNN to extract a feature vector from each frame of an input video and averaged those vectors.

In all these recent applications of the encoder–decoder framework, the continuous-space representation c of the input x returned by an encoder, in Eq. (8) has been a fixed-dimensional vector, regardless of the size of the input.[2] Furthermore, the context vector was not structured by design, but rather an arbitrary vector, which means that there is no guarantee that the context vector preserves the spatial, temporal or spatio-temporal structures of the input. Henceforth, we refer to an encoder–decoder based model with a fixed-dimensional context vector as a *simple encoder–decoder model*.

B. Incorporating an Attention Mechanism

1) Motivation: A naive implementation of the encoder–decoder framework, as in the simple encoder–decoder model, requires the encoder to compress the input into a single vector of predefined dimensionality, regardless of the size of or the amount of information in the input. For instance, the recurrent neural network (RNN) based encoder used in [3], [25] for machine translation needs to be able to summarize a variable-length source sentence into a single fixed-dimensional vector. Even when the size of the input is fixed, as in the case of a fixed-resolution image, the amount of information contained in each image may vary significantly (consider a varying number of objects in each image).

In [29], it was observed that the performance of the neural machine translation system based on a simple encoder–decoder model rapidly degraded as the length of the source sentence grew. The authors of [29] hypothesized that it was due to the limited capacity of the simple encoder–decoder's fixed-dimensional context vector.

Furthermore, the interpretability of the simple encoder–decoder is extremely low. As all the information required for the decoder to generate the output is compressed in a context vector without any presupposed structure, such structure is not available to techniques designed to inspect the representations captured by the model [12], [30], [31].

2) Attention Mechanism for Encoder–Decoder Models: We the introduction of an attention mechanism in between the encoder and decoder, we address these two issues, i.e., (1) limited capacity of a fixed-dimensional context vector and (2) lack of interpretability.

The first step into introducing the attention mechanism to the encoder–decoder framework is to let the encoder return a structured representation of the input. We achieve this by allowing the continuous-space representation to be a set of fixed-size vectors, to which we refer as a *context set*, i.e.,

$$c = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_M\}$$

See Eq. (8). Each vector in the context set is localized to a certain spatial, temporal or spatio-temporal component of the input. For instance, in the case of an image input, each

context vector $\mathbf{c}_i$ will summarize a certain spatial location of the image (see Sec. IV-B), and with machine translation, each context vector will summarize a phrase centered around a specific word in a source sentence (see Sec. IV-A.) In all cases, the number of vectors M in the context set c may vary across input examples.

The choice of the encoder and of the kind of context set it will return is governed by the application and the type of the input considered. In this paper, we assume that the decoder is a conditional RNN-LM from Sec. II-B, i.e., the goal is to describe the input in a natural language sentence.

The attention mechanism controls the input actually seen by the decoder and requires another neural network, to which refer as the attention model. The main job of the attention model is to score each context vector $\mathbf{c}_i$ with respect to the current hidden state $\mathbf{z}_{t-1}$ of the decoder:[3]

$$e_i^t = f_{\text{ATT}}(\mathbf{z}_{t-1}, \mathbf{c}_i, \{\alpha_j^{t-1}\}_{j=1}^M), \tag{10}$$

where α_j^{t-1} represents the attention weights computed at the previous time step, from the scores e_i^{t-1}, through a softmax that makes them sum to 1:

$$\alpha_i^t = \frac{\exp(e_i^t)}{\sum_{j=1}^M \exp(e_j^t)}, \tag{11}$$

This type of scoring can be viewed as assigning a probability of being *attended* by the decoder to each context, hence the name of the attention model.

Once the attention weights are computed, we use them to compute the new context *vector* $\mathbf{c}^t$:

$$\mathbf{c}^t = \varphi\left(\{\mathbf{c}_i\}_{i=1}^M, \{\alpha_i^t\}_{i=1}^M\right), \tag{12}$$

where φ returns a vector summarizing the whole context set c according to the attention weights.

A usual choice for φ is a simple weighted sum of the context vectors such that

$$\mathbf{c}^t = \varphi\left(\{\mathbf{c}_i\}_{i=1}^M, \{\alpha_i^t\}_{i=1}^M\right) = \sum_{i=1}^M \alpha_i \mathbf{c}_i. \tag{13}$$

On the other hand, we can also force the attention model to make a hard decision on which context vector to consider by sampling one of the context vectors following a categorical (or multinoulli) distribution:

$$\mathbf{c}^t = \mathbf{c}_{r^t}, \text{ where } r^t \sim \text{Cat}(M, \{\alpha_i^t\}_{i=1}^M). \tag{14}$$

With the newly computed context vector $\mathbf{c}_t$, we can update the hidden state of the decoder, which is a conditional RNN-LM here, by

$$\mathbf{h}_t = \phi_\theta\left(\mathbf{h}_{t-1}, \mathbf{x}_t, \mathbf{c}_t\right). \tag{15}$$

This way of computing a context vector at each time step t of the decoder frees the encoder from compressing any variable-length input into a single fixed-dimensional vector. By spatially or temporally dividing the input[4], the encoder can

---

[2] Note that in the case of machine translation and video description generation, the size of the input varies.

[3] We use $\mathbf{z}_t$ to denote the hidden state of the decoder to distinguish it from the encoder's hidden state for which we used $\mathbf{h}_t$ in Eq. (1).

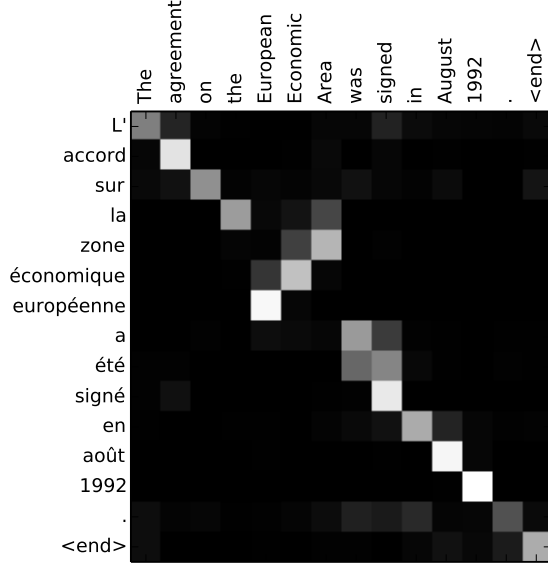[4] Note that it is possible, or even desirable to use overlapping regions.

Fig. 2. Visualization of the attention weights α_j^t of the attention-based neural machine translation model [32]. Each row corresponds to the output symbol, and each column the input symbol. Brighter the higher α_j^t.

represent the input into a set of vectors of which each needs to encode a fixed amount of information focused around a particular region of the input. In other words, the introduction of the attention mechanism bypasses the issue of limited capacity of a fixed-dimensional context vectors.

Furthermore, this attention mechanism allows us to directly inspect the internal working of the whole encoder–decoder model. The magnitude of the attention weight α_j^t, which is positive by construction in Eq. (11), highly correlates with how predictive the spatial, temporal or spatio-temporal region of the input, to which the j-th context vector corresponds, is for the prediction associated with the t-th output variable y_t. This can be easily done by visualizing the attention matrix $\left[\alpha_j^t\right]_{t,j} \in \mathbb{R}^{T' \times M}$, as in Fig. 2.

This attention-based approach with the weighted sum of the context vectors (see Eq. (13)) was originally proposed in [32] in the context of machine translation, however, with a simplified (content-based) scoring function:

$$e_i^t = f_{\text{ATT}}(\mathbf{z}_{t-1}, \mathbf{c}_i). \qquad (16)$$

See the missing $\{\alpha_j^{t-1}\}_{j=1}^M$ from Eq. (10). In [22], it was further extended with the hard attention using Eq. (14). In [33] this attention mechanism was extended to be by taking intou account the past values of the attention weights as the general scoring function from Eq. (10), following an approach based purely on those weights introduced by [34]. We will discuss more in detail these three applications/approaches in the later sections.

C. Learning

As usual with many machine learning models, the attention-based encoder–decoder model is also trained to maximize the log-likelihood of a given training set with respect to the

parameters, where the log-likelihood is defined as

$$\mathcal{L}\left(D = \{(x^n, y^n)\}_{n=1}^N, \Theta\right) = \frac{1}{N} \sum_{n=1}^N \log p(y^n \mid x^n, \Theta),$$
$$(17)$$

where Θ is a set of all the trainable parameters of the model.

1) Maximum Likelihood Learning: When the weighted sum is used to compute the context vector, as in Eq. (13), the whole attention-based encoder–decoder model becomes one large differentiable function. This allows us to compute the gradient of the log-likelihood in Eq. (17) using backpropagation [35]. With the computed gradient, we can use, for instance, the stochastic gradient descent (SGD) algorithm to iteratively update the parameters Θ to maximize the log-likelihood.

2) Variational Learning for Hard Attention Model: When the attention model makes a hard decision each time as in Eq. (14), the derivatives through the stochastic decision are zero, because those decisions are discrete. Hence, the information about how to improve the way to take those focus-of-attention decisions is not available from back-propagation, while it is needed to train the attention mechanism. The question of training neural networks with stochastic discrete-valued hidden units has a long history, starting with Boltzmann machines [36], with recent work studying how to deal with such units in a system trained using back-propagated gradients [37], [38], [39], [40]. Here we briefly describe the variational learning approach from [39], [22].

With stochastic variables $\mathbf{r}$ involved in the computation from inputs to outputs, the log-likelihood in Eq. (17) is re-written into

$$\mathcal{L}\left(D = \{(x^n, y^n)\}_{n=1}^N, \Theta\right) = \frac{1}{N} \sum_{n=1}^N l(y^n, x^n, \Theta),$$

where

$$l(y, x, \Theta) = \log \sum_{\mathbf{r}} p(y, \mathbf{r} | x, \Theta)$$

and $\mathbf{r} = (r_1, r_2, \ldots, r_T')$. We derive a lowerbound of l as

$$l(y, x) = \log \sum_{\mathbf{r}} p(y|\mathbf{r}, x) p(\mathbf{r}|x)$$
$$\geq \sum_{\mathbf{r}} p(\mathbf{r}|x) \log p(y|\mathbf{r}, x). \qquad (18)$$

Note that we omitted Θ to make the equation less cluttered.

The gradient of l with respect to Θ is then

$$\nabla l(y, x) = \sum_{\mathbf{r}} p(\mathbf{r}|x) [\nabla \log p(y|\mathbf{r}, x)$$
$$+ \log p(y|\mathbf{r}, x) \nabla \log p(\mathbf{r}|x)] \qquad (19)$$

which is often approximated by Monte Carlo sampling:

$$\nabla l(y, x) \approx \frac{1}{M} \sum_{m=1}^M \nabla \log p(y|\mathbf{r}^m, x)$$
$$+ \log p(y|\mathbf{r}^m, x) \nabla \log p(\mathbf{r}^m|x). \qquad (20)$$

As the variance of this estimator is high, a number of variance reduction techniques, such as baselines and variance normalization, are often used in practice [41], [39].

Once the gradient is estimated, any usual gradient-based iterative optimization algorithm can be used to approximately maximize the log-likelihood.

IV. Applications

In this section, we introduce some of the recent work in which the attention-based encoder–decoder model was applied to various multimedia description generation tasks.

A. Neural Machine Translation

Machine translation is a task in which a sentence in one language (source) is translated into a corresponding sentence in another language (target). *Neural* machine translation aims at solving it with a single neural network based model, jointly trained end-to-end. The encoder–decoder framework described in Sec. III-A was proposed for neural machine translation recently in [24], [3], [25]. Based on these works, in [32], the attention-based model was proposed to make neural machine translation systems more robust to long sentences. Here, we briefly describe the model from [32].

1) Model Description: The attention-based neural machine translation in [32] uses a bidirectional recurrent neural network (BiRNN) as an encoder. The forward network reads the input sentence $x = (x_1, \ldots, x_T)$ from the first word to the last, resulting in a sequence of state vectors

$$\left\{ \overrightarrow{\mathbf{h}}_1, \overrightarrow{\mathbf{h}}_2, \ldots, \overrightarrow{\mathbf{h}}_T \right\}.$$

The backward network, on the other hand, reads the input sentence in the reverse order, resulting in

$$\left\{ \overleftarrow{\mathbf{h}}_T, \overleftarrow{\mathbf{h}}_{T-1}, \ldots, \overleftarrow{\mathbf{h}}_1 \right\}.$$

These vectors are concatenated per step to form a context set (see Sec. III-B2) such that $\mathbf{c}_t = \left[\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t \right]$.
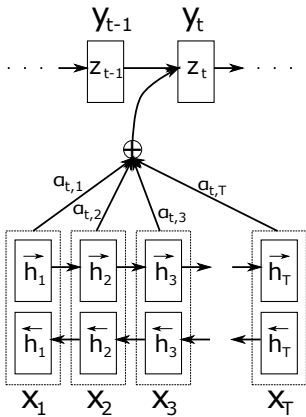

Fig. 3. Illustration of a single step of decoding in attention-based neural machine translation [32].

The use of the BiRNN is crucial if the content-based attention mechanism is used. The content-based attention mechanism in Eqs. (16) and (11) relies solely on a so-called *content-based scoring*, and without the context information from the whole sentence, words that appear multiple times in a source sentence cannot be distinguished by the attention model.

The decoder is a conditional RNN-LM that models the target language given the context set from above. See Fig. 3 for

the graphical illustration of the attention-based neural machine translation model.

TABLE I
The translation performances and the relative improvements over the simple encoder-decoder model on an English-to-French translation task, measured by BLEU [32], [42]. ⋆: an ensemble of multiple attention-based models. ∘: the state-of-the-art phrase-based statistical machine translation system [43].

Model	BLEU	Rel. Improvement
Simple Enc–Dec	17.82	–
Attention-based Enc–Dec	28.45	+59.7%
Attention-based Enc–Dec (LV)	34.11	+90.7%
Attention-based Enc–Dec (LV)⋆	**37.19**	**+106.0%**
State-of-the-art SMT∘	37.03	–

2) Experimental Result: Given a fixed model size, the attention-based model proposed in [32] was able to achieve a relative improvement of more than 50% in the case of the English-to-French translation task, as shown in Table I. When the very same model was extended with a very large target vocabulary [42], the relative improvement over the baseline without the attention mechanism was 90%. Additionally, the very same model was recently tested on a number of European language pairs at the WMT'15 Translation Task.[5]. See Table II for the results.

The authors of [44] recently proposed a method for incorporating a monolingual language model into the attention-based neural machine translation system. With this method, the attention-based model was shown to outperform the existing statistical machine translation systems on Chinese-to-English (restricted domains) and Turkish-to-English translation tasks as well as other European languages they tested.

B. Image Caption Generation

Image caption generation is a task in which a model looks at an input image and generates a corresponding natural language description. The encoder–decoder framework fits well with this task. The encoder will extract the continuous-space representation, or the context, of an input image, for instance, with a deep convolutional network (see Sec. II-C,) and from this representation the conditional RNN-LM based decoder generates a natural language description of the image. Very recently (Dec 2014), a number of research groups independently proposed to use the simple encoder–decoder model to solve the image caption generation [18], [17], [19], [20].

[5]http://www.statmt.org/wmt15/

TABLE II
The performance of the attention-based neural machine translation models with the very large target vocabulary in the WMT'15 Translation Track [42]. We show the results on two representative language pairs. For the complete result, see http://matrix.statmt.org/.

Language Pair	Model	BLEU	Note
En->De	NMT	24.8	
	Best Non-NMT	24.0	Syntactic SMT (Edinburgh)
En->Cz	NMT	18.3	
	Best Non-NMT	18.2	Phrase SMT (JHU)

Instead, here we describe a more recently proposed approach based on the attention-based encoder–decoder framework in [22].
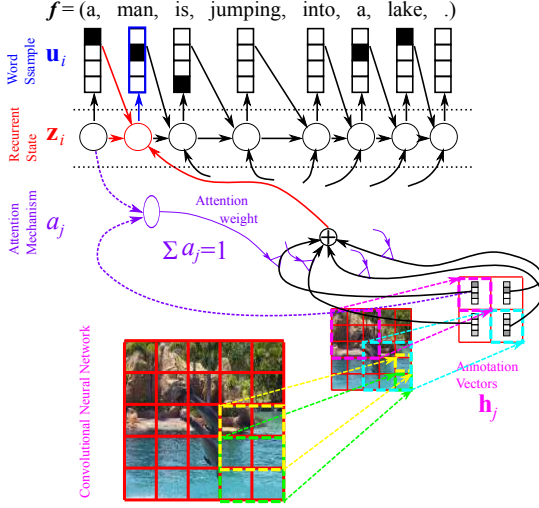

Fig. 4. Graphical illustration of the attention-based encoder–decoder model for image caption generation.

1) Model Description: The usual encoder–decoder based image caption generation models use the activation of the last fully-connected hidden layer as the continuous-space representation, or the context vector, of the input image (see Sec. II-D.) The authors of [22] however proposed to use the activation from the last *convolutional* layer of the pre-trained convolutional network, as in the bottom half of Fig. 4.

Unlike the fully-connected layer, in this case, the context set consists of multiple vectors that correspond to different spatial regions of the input image on which the attention mechanism can be applied. Furthermore, due to convolution and pooling, the spatial locations in pixel space represented by each context vector overlaps substantially with those represented by the neighbouring context vectors, which helps the attention mechanism distinguish similar objects in an image using its context information with respect to the whole image, or the neighbouring pixels.

Similarly to the attention-based neural machine translation in Sec. IV-A, the decoder is implemented as a conditional RNN-LM. In [22], the content-based attention mechanism (see Eq. (16)) with either the weighted sum (see Eq. (13)) or hard decision (see Eq. (14) was tested by training a model with the maximum likelihood estimator from Sec. III-C1 and the variational learning from Sec. III-C2, respectively. The authors of [22] reported the similar performances with these two approaches on a number of benchmark datasets.

2) Experimental Result: In [22], the attention-based image caption generator was evaluated on three datasets; Flickr 8K [47], Flickr 30K [48] and MS CoCo [49]. In addition to the self-evaluation, an ensemble of multiple attention-based models was submitted to Microsoft COCO Image Captioning Challenge[6] and evaluated with multiple automatic evaluation metrics[7] as well as by human evaluators.

[6]https://www.codalab.org/competitions/3221
[7] BLEU [50], METEOR [51], ROUGE-L [52] and CIDEr [53].

TABLE III
THE PERFORMANCES OF THE IMAGE CAPTION GENERATION MODELS IN THE MICROSOFT COCO IMAGE CAPTIONING CHALLENGE. (⋆) [20], (•) [18], (○) [45], (◇) [46] AND (∗) [22]. THE ROWS ARE SORTED ACCORDING TO M1.

Model	Human		Automatic	
	M1	M2	BLEU	CIDEr
Human	0.638	0.675	0.471	0.91
Google⋆	0.273	0.317	0.587	0.946
MSR•	0.268	0.322	0.567	0.925
Attention-based∗	0.262	0.272	0.523	0.878
Captivator○	0.250	0.301	0.601	0.937
Berkeley LRCN◇	0.246	0.268	0.534	0.891

In this Challenge, the attention-based approach ranked third based on the percentage of captions that are evaluated as better or equal to human caption (M1) and the percentage of captions that pass the Turing Test (M2). Interestingly, the same model was ranked eighth according to the most recently proposed metric of CIDEr and ninth according to the most widely used metric of BLEU.[8] It means that this model has better relative performance in terms of human evaluation than in terms of the automatic metrics, which only look at matching subsequences of words, not directly at the meaning of the generated sentence. The performance of the top-ranked systems, including the attention-based model from [22], are listed in Table III.

The attention-based model was further found to be highly interpretable, especially, compared to the simple encoder–decoder models. See Fig. 5 for some examples.

C. Video Description Generation

Soon after the neural machine translation based on the simple encoder–decoder framework was proposed in [25], [3], it was further applied to video description generation, which amounts to translating a (short) video clip to its natural language description [28]. The authors of [28] used a pre-trained convolutional network (see Sec. II-D) to extract a feature vector from each frame of the video clip and average all the frame-specific vectors to obtain a single fixed-dimensional context vector of the whole video. A conditional RNN-LM from Sec. II-B was used to generate a description based on this context vector.

Since any video clip clearly has both temporal and spatial structures, it is possible to exploit them by using the attention mechanism described throughout this paper. In [23], the authors proposed an approach based on the attention mechanism to exploit the global and local temporal structures of the video clips. Here we briefly describe their approach.

1) Model Description: In [23], two different types of encoders are tested. The first one is a simple frame-wise application of the pre-trained convolutional network. However, they did not pool those per-frame context vectors as was done in [28], but simply form a context set consisting of all the per-frame feature vectors. The attention mechanism will work to select one of those per-frame vectors for each output symbol being decoded. In this way, the authors claimed that the overall model captures the *global* temporal structure (the structure across many frames, potentially across the whole video clip.)
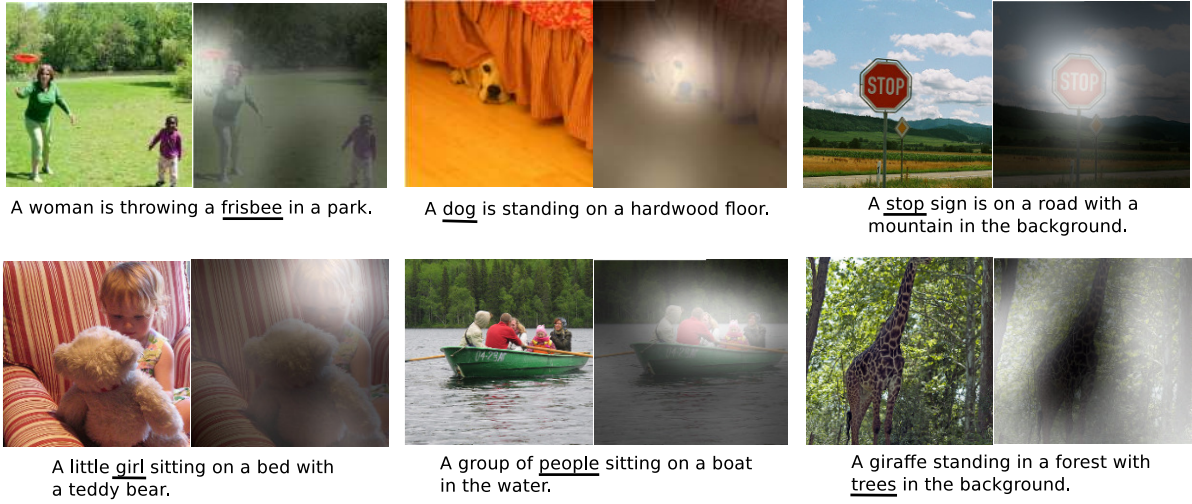
[8]http://mscoco.org/dataset/#leaderboard-cap

A woman is throwing a <u>frisbee</u> in a park.

A <u>dog</u> is standing on a hardwood floor.

A <u>stop</u> sign is on a road with a mountain in the background.

A little <u>girl</u> sitting on a bed with a teddy bear.

A group of <u>people</u> sitting on a boat in the water.

A giraffe standing in a forest with <u>trees</u> in the background.

Fig. 5. Examples of the attention-based model attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word) [22]
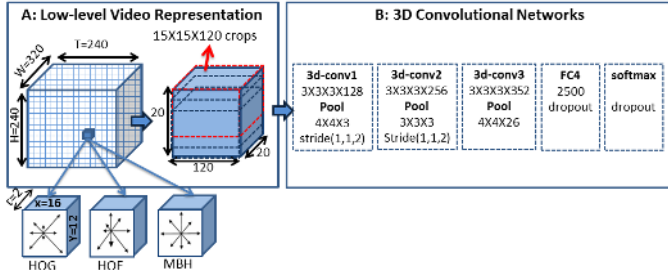

Fig. 6. The 3-D convolutional network for motion from [23].

The other type of encoder in [23] is a so-called 3-D convolutional network, shown in Fig. 6. Unlike the usual convolutional network which often works only spatially over a two-dimensional image, the 3-D convolutional network applies its (local) filters across the spatial dimensions as well as the temporal dimensions. Furthermore, those filters work not on pixels but on local motion statistics, enabling the model to concentrate on motion rather than appearance. Similarly to the strategy from Sec. II-D, the model was trained on larger video datasets to recognize an action from each video clip, and the activation vectors from the last convolutional layer were used as context. The authors of [23] suggest that this encoder extracts more local temporal structures complementing the global structures extracted from the frame-wise application of a 2-D convolutional network.

The same type of decoder, a conditional RNN-LM, used in [22] was used with the content-based attention mechanism in Eq. (16).

2) Experimental Result: In [23], this approach to video description generation has been tested on two datasets; (1) Youtube2Text [54] and (2) Montreal DVS [55]. They showed that it is beneficial to have both types of encoders together in their attention-based encoder–decoder model, and that the attention-based model outperforms the simple encoder–decoder model. See Table IV for the summary of the evaluation.

TABLE IV
THE PERFORMANCE OF THE VIDEO DESCRIPTION GENERATION MODELS ON YOUTUBE2TEXT AND MONTREAL DVS. ($\star$) HIGHER THE BETTER. ($\circ$) LOWER THE BETTER.

Model	Youtube2Text		Montreal DVS	
	METEOR*	Perplexity$^\circ$	METEOR	Perplexity
Enc-Dec	0.2868	33.09	0.044	88.28
+ 3-D CNN	0.2832	33.42	0.051	84.41
+ Per-frame CNN	0.2900	27.89	.040	66.63
+ Both	0.2960	27.55	0.057	65.44

Similarly to all the other previous applications of the attention-based model, the attention mechanism applied to the task of video description also provides a straightforward way to inspect the inner workings of the model. See Fig. 7 for some examples.
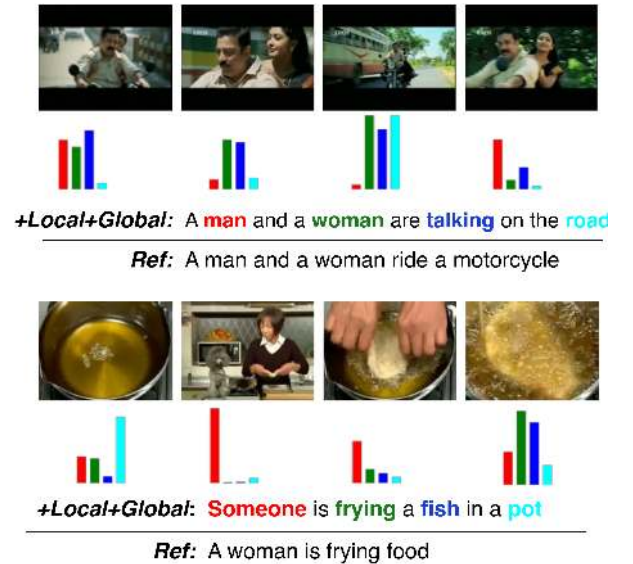

Fig. 7. Two sample videos and their corresponding generated and ground-truth descriptions from Youtube2Text. The bar plot under each frame corresponds to the attention weight α_j^t (see Eq. (11)) for the frame when the corresponding word (color-coded) was generated. Reprinted from [23].

D. End-to-End Neural Speech Recognition

Speech recognition is a task in which a given speech waveform is translated into a corresponding natural language transcription. Deep neural networks have become a standard for the acoustic part of speech recognition systems [56]. Once the input speech (often in the form of spectral filter response) is processed with the deep neural network based acoustic model, another model, almost always a hidden Markov model (HMM), is used to map correctly the much longer sequence of speech into a shorter sequence of phonemes/characters/words. Only recently, in [57], [8], [58], [59], fully neural network based speech recognition models were proposed.

Here, we describe the recently proposed attention-based fully neural speech recognizer from [33]. For more detailed comparison between the attention-based fully speech recognizer and other neural speech recognizers, e.g., from [58], we refer the reader to [33].

1) Model Description–Hybrid Attention Mechanism: The basic architecture of the attention-based model for speech recognition in [33] is similar to the other attention-based models described earlier, especially the attention-based neural machine translation model in Sec. IV-A. The encoder is a stacked bidirectional recurrent neural network (BiRNN) [60] which reads the input sequence of speech frames, where each frame is a 123-dimensional vector consisting of 40 Mel-scale filter-bank response, the energy and first- and second-order temporal differences. The context set of the concatenated hidden states from the top-level BiRNN is used by the decoder based on the conditional RNN-LM to generate the corresponding transcription, which in the case of [33], consists in a sequence of phonemes.

The authors of [33] however noticed the peculiarity of speech recognition compared to, for instance, machine translation. First, the lengths of the input and output differ significantly; thousands of input speech frames against a dozen of words. Second, the alignment between the symbols in the input and output sequences is monotonic, where this is often not true in the case of translation.

These issues, especially the first one, make it difficult for the content-based attention mechanism described in Eqs. (16) and (11) to work well. The authors of [33] investigated these issues more carefully and proposed that the attention mechanism with location awareness are particulary appropriate (see Eq. (10). The location awareness in this case means that the attention mechanism directly takes into account the previous attention weights to compute the next ones.

The proposed location-aware attention mechanism scores each context vector by

$$e_i^t = f_{\text{ATT}}(\mathbf{z}_{t-1}, \mathbf{c}_i, f_{\text{LOC}}^i(\{\alpha_j^{t-1}\}_{j=1}^T),$$

where f_{LOC}^j is a function that extracts information from the previous attention weights $\{\alpha_j^{t-1}\}$ for the i-th context vector. In other words, the location-aware attention mechanism takes into account both the content $\mathbf{c}_i$ and the previous attention weights $\{\alpha_j^{t-1}\}_{j=1}^T$.

In [33], f_{LOC}^j was implemented as

$$f_{\text{LOC}}^j(\{\alpha_j^t\}) = \sum_{k=j-\frac{K}{2}}^{j+\frac{K}{2}} \mathbf{v}_k \alpha_k^{t-1}, \qquad (21)$$

where K is the size of the window, and $\mathbf{v}_k \in \mathbb{R}^d$ is a learned vector.

Furthermore, the authors of [33] proposed additional modifications to the attention mechanism, such as sharpening, windowing and smoothing, which modify Eq. (11). For more details of each of these, we refer the reader to [33].

2) Experimental Result: In [33], this attention-based speech recognizer was evaluated on the widely-used TIMIT corpus [61], closely following the procedure from [62]. As can be seen from Table V, the attention-based speech recognizer with the location-aware attention mechanism can recognize a sequence of phonemes given a speech segment can perform better than the conventional fully neural speech recognition. Also, the location-aware attention mechanism helps the model achieve better generalization error.

TABLE V
PHONEME ERROR RATES (PER). THE BOLD-FACED PER CORRESPONDS TO THE BEST ERROR RATE ACHIEVED WITH A FULLY NEURAL NETWORK BASED MODEL. FROM [33].

Model	Dev	Test
Attention-based Model	15.9%	18.7%
Attention-based Model + Location-Awareness	15.8%	**17.6%**
RNN Transducer [62]	N/A	17.7%
Time/Frequency Convolutional Net+HMM [63]	13.9%	16.7%

Similarly to the previous applications, it is again possible to inspect the model's behaviour by visualizing the attention weights. An example is shown in Fig. 8, where we can clearly see how the model attends to a roughly correct window of speech each time it generates a phoneme.

E. Beyond Multimedia Content Description

We briefly present three recent works which applied the described attention-based mechanism to tasks other than multimedia content description.

1) Parsing–Grammar as a Foreign Language: Parsing a sentence into a parse tree can be considered as a variant of machine translation, where the target is not a sentence but its parse tree. In [64], the authors evaluate the simple encoder–decoder model and the attention-based model on generating the linearized parse tree associated with a natural language sentence. Their experiments revealed that the attention-based parser can match the existing state-of-the-art parsers which are often highly domain-specific.

2) Discrete Optimization–Pointer Network: In [65], the attention mechanism was used to (approximately) solve discrete optimization problems. Unlike the usual use of the described attention mechanism where the decoder generates a sequence of output symbols, in their application to discrete optimization, the decoder predicts which one of the source symbols/nodes should be chosen at each time step. The authors achieve this

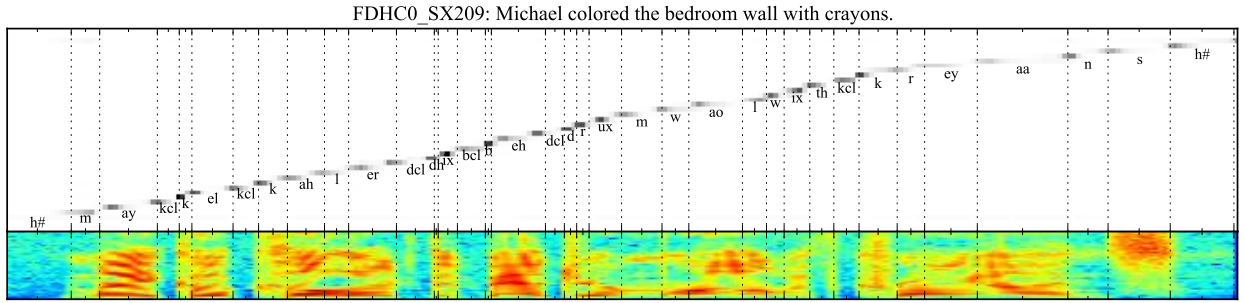FDHC0_SX209: Michael colored the bedroom wall with crayons.

Fig. 8. Attention weights by the attention-based model with location-aware attention mechanism. The vertical bars indicate ground-truth phone location. For more details, see [33].

by considering α_i^t as the probability of choosing the i-th input symbol as the selected one, at each time step t.

For instance, in the case of travelling salesperson problem (TSP), the model needs to generate a sequence of cities/nodes that cover the whole set of input cities so that the sequence will be the shortest possible route in the input map (a graph of the cities) to cover every single city/node. First, the encoder reads the graph of a TSP instance and returns a set of context vectors, each of which corresponds to a city in the input graph. The decoder then returns a sequence of probabilities over the input cities, or equivalently the context vectors, which are computed by the attention mechanism. The model is trained to generate a sequence to cover all the cities by correctly *attending* to each city using the attention mechanism.

As was shown already in [65], this approach can be applied to any discrete optimization problem whose solution is expressed as a subset of the input symbols, such as sorting.

3) Question Answering–Weakly Supervised Memory Network: The authors of [66] applied the attention-based model to a question-answering (QA) task. Each instance of this QA task consists of a set of facts and a question, where each fact and the question are both natural language sentences. Each fact is encoded into a continuous-space representation, forming a context set of fact vectors. The attention mechanism is applied to the context set given the continuous-space representation of the question so that the model can focus on the relevant facts needed to answer the question.

V. Related Work: Attention-based Neural Networks

The most related, relevant model is a neural network with *location-based* attention mechanism, as opposed to the content-based attention mechanism described in this paper. The content-based attention mechanism computes the relevance of each spatial, temporal or spatio-temporally localized region of the input, while the location-based one directly returns to which region the model needs to attend, often in the form of the coordinate such as the (x, y)-coordinate of an input image or the offset from the current coordinate.

In [34], the location-based attention mechanism was successfully used to model and generate handwritten text. In [39], [67], a neural network is designed to use the location-based attention mechanism to recognize objects in an image. Furthermore, a generative model of images was proposed in

[68], which iteratively reads and writes portions of the whole image using the location-based attention mechanism. Earlier works on utilizing the attention mechanism, both content-based and location-based, for object recognition/tracking can be found in [69], [70], [71].

The attention-based mechanim described in this paper, or its variant, may be applied to something other than multimedia input. For instance, in [72], a neural Turing machine was proposed, which implements a memory controller using both the content-based and location-based attention mechanisms. Similarly, the authors of [73] used the content-based attention mechanism with hard decision (see, e.g., Eq. (14)) to find relevant memory contents, which was futher extended to the weakly supervised memory network in [66] in Sec. IV-E3.

VI. Looking Ahead...

In this paper, we described the recently proposed attention-based encoder–decoder architecture for describing multimedia content. We started by providing background materials on recurrent neural networks (RNN) and convolutional networks (CNN) which form the building blocks of the encoder–decoder architecture. We emphasized the specific variants of those networks that are often used in the encoder–decoder model; a conditional language model based on RNNs (a conditional RNN-LM) and a pre-trained CNN for transfer learning. Then, we introduced the simple encoder–decoder model followed by the attention mechanism, which together form the central topic of this paper, the attention-based encoder–decoder model.

We presented four recent applications of the attention-based encoder–decoder models; machine translation (Sec. IV-A), image caption generation (Sec. IV-B), video description generation (Sec. IV-C) and speech recognition (Sec. IV-D). We gave a concise description of the attention-based model for each of these applications together with the model's performance on benchmark datasets. Furthermore, each description was accompanied with a figure visualizing the behaviour of the attention mechanism.

In the examples discussed above, the attention mechanism was primarily considered as a means to building a model that can describe the input multimedia content in natural language, meaning the ultimate goal of the attention mechanism was to aid the encoder–decoder model for multimedia content description. However, this should not be taken as the only possible application of the attention mechanism. Indeed, as

recent work such as the pointer networks [65] suggests, future applications of attention mechanisms could run the range of AI-related tasks.

Beside superior performance it delivers, an attention mechanism can be used to extract the underlying mapping between two entirely different modalities *without* explicit supervision of the mapping. From Figs. 2, 5, 7 and 8, it is clear that the attention-based models were able to infer – *in an unsuperivsed way* – alignments between different modalities (multimedia and its text description) that agree well with our intuition. This suggests that this type of attention-based model can be used solely to extract these underlying, often complex, mappings from a pair of modalities, where there is not much prior/-domain knowledge. As an example, attention-based models can be used in neuroscience to temporally and spatially map between the neuronal activities and a sequence of stimuli [74].

ACKNOWLEDGMENT

REFERENCES

[1] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. Berg, "Babytalk: Understanding and generating simple image descriptions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 12, pp. 2891–2903, 2013.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, Oct. 2014.

[4] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," NIPS'2014 Deep Learning workshop, arXiv 1412.3555, 2014.

[5] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Nets*, pp. 157–166, 1994.

[6] S. Hochreiter, F. F. Informatik, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," in *Field Guide to Dynamical Recurrent Networks*, J. Kolen and S. Kremer, Eds. IEEE Press, 2000.

[7] T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Proc. 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP 2011)*, 2011.

[8] A. Graves, "Sequence transduction with recurrent neural networks," in *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, 2012.

[9] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Audio chord recognition with recurrent neural networks," in *ISMIR*, 2013.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[11] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25 (NIPS'2012)*, 2012.

[12] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV'14*, 2014.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," arXiv:1409.4842, Tech. Rep., 2014.

[15] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *International Conference on Learning Representations*, 2014.

[16] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*. IEEE, 2014, pp. 512–519.

[17] A. Karpathy and F.-F. Li, "Deep visual-semantic alignments for generating image descriptions," in *CVPR'2015*, 2015, arXiv:1412.2306.

[18] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, C. L. Zitnick, and G. Zweig, "From captions to visual concepts and back," 2015, arXiv:1411.4952.

[19] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. L. Yuille, "Deep captioning with multimodal recurrent neural networks," in *ICLR'2015*, 2015, arXiv:1410.1090.

[20] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: a neural image caption generator," in *CVPR'2015*, 2015, arXiv:1411.4555.

[21] R. Kiros, R. Salakhutdinov, and R. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," *arXiv:1411.2539 [cs.LG]*, Nov. 2014.

[22] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *ICML'2015*, 2015.

[23] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," *arXiv: 1502.08029*, 2015.

[24] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *EMNLP'2013*, 2013.

[25] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NIPS'2014*, 2014.

[26] G. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *ECCV'10*, 2010.

[27] A. Mnih and G. E. Hinton, "Three new graphical models for statistical language modelling," 2007, pp. 641–648.

[28] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," arXiv:1412.4729, 2014.

[29] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–Decoder approaches," in *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, Oct. 2014.

[30] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," in *ICLR*, 2015.

[31] M. Denil, A. Demiraj, and N. de Freitas, "Extraction of salient sentences from labelled documents," University of Oxford, Tech. Rep., 2014.

[32] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR'2015, arXiv:1409.0473*, 2015.

[33] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *arXiv preprint arXiv: 1506.07503*, 2015.

[34] A. Graves, "Generating sequences with recurrent neural networks," arXiv:1308.0850, Tech. Rep., 2013.

[35] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[36] G. E. Hinton and T. J. Sejnowski, "Learning and relearning in Boltzmann machines," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986, pp. 282–317.

[37] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," arXiv:1308.3432, 2013.

[38] Y. Tang and R. Salakhutdinov, "Learning stochastic feedforward neural networks," in *NIPS'2013*, 2013.

[39] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," *arXiv:1412.7755*, 2014.

[40] T. Raiko, M. Berglund, G. Alain, and L. Dinh, "Techniques for learning binary stochastic feedforward neural networks," in *ICLR*, 2015.

[41] A. Mnih and K. Gregor, "Neural variational inference and learning in belief networks," *CoRR*, vol. abs/1402.0030, 2014.

[42] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, "On using very large target vocabulary for neural machine translation," in *ACL-IJCNLP'2015*, 2015, arXiv:1412.2007.

[43] N. Durrani, B. Haddow, P. Koehn, and K. Heafield, "Edinburgh's phrase-based machine translation systems for WMT-14," in *Proceedings of the Ninth Workshop on Statistical Machine Translation.* Association for Computational Linguistics Baltimore, MD, USA, 2014, pp. 97–104.

[44] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, and Y. Bengio, "On using monolingual corpora in neural machine translation," *arXiv preprint arXiv:1503.03535*, 2015.

[45] J. Devlin, H. Cheng, H. Fang, S. Gupta, L. Deng, X. He, G. Zweig, and M. Mitchell, "Language models for image captioning: The quirks and what works," *arXiv preprint arXiv:1505.01809*, 2015.

[46] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," arXiv:1411.4389, 2014.

[47] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *Journal of Artificial Intelligence Research*, pp. 853–899, 2013.

[48] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *TACL*, vol. 2, pp. 67–78, 2014.

[49] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *ECCV*, 2014, pp. 740–755.

[50] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics.* Association for Computational Linguistics, 2002, pp. 311–318.

[51] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014.

[52] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out: Proceedings of the ACL-04 workshop*, vol. 8, 2004.

[53] R. Vedantam, C. L. Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," *arXiv preprint arXiv:1411.5726*, 2014.

[54] D. L. Chen and W. B. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Portland, Oregon, USA, June 2011, pp. 190–200.

[55] A. Torabi, C. Pal, H. Larochelle, and A. Courville, "Using descriptive video services to create a large data source for video annotation research," *arXiv preprint arXiv: 1503.01070*, 2015.

[56] G. E. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[57] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *ICML'2006*, Pittsburgh, USA, 2006, pp. 369–376.

[58] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *ICML'2014*, 2014.

[59] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deepspeech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[60] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," in *ICLR*, 2014.

[61] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *NASA STI/Recon Technical Report N*, vol. 93, p. 27403, 1993.

[62] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *ICASSP'2013*, 2013, pp. 6645–6649.

[63] L. Tóth, "Combining time-and frequency-domain convolution in convolutional neural network-based phone recognition," in *ICASSP 2014*, 2014, pp. 190–194.

[64] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton, "Grammar as a foreign language," *arXiv preprint arXiv:1412.7449*, 2014.

[65] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," *arXiv preprint arXiv:1506.03134*, 2015.

[66] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "Weakly supervised memory networks," *arXiv preprint arXiv:1503.08895*, 2015.

[67] V. Mnih, N. Heess, A. Graves, and k. kavukcuoglu, "Recurrent models of visual attention," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2204–2212.

[68] K. Gregor, I. Danihelka, A. Graves, and D. Wierstra, "DRAW: A recurrent neural network for image generation," *arXiv preprint arXiv:1502.04623*, 2015.

[69] H. Larochelle and G. E. Hinton, "Learning to combine foveal glimpses with a third-order Boltzmann machine," in *Advances in Neural Information Processing Systems 23*, 2010, pp. 1243–1251.

[70] M. Denil, L. Bazzani, H. Larochelle, and N. de Freitas, "Learning where to attend with deep architectures for image tracking," *Neural Computation*, vol. 24, no. 8, pp. 2151–2184, 2012.

[71] Y. Zheng, R. S. Zemel, Y.-J. Zhang, and H. Larochelle, "A neural autoregressive approach to attention-based recognition," *International Journal of Computer Vision*, vol. 113, no. 1, pp. 67–79, 2014.

[72] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[73] J. Weston, S. Chopra, and A. Bordes, "Memory networks," *arXiv preprint arXiv:1410.3916*, 2014.

[74] L. Wehbe, B. Murphy, P. Talukdar, A. Fyshe, A. Ramdas, and T. Mitchell, "Simultaneously uncovering the patterns of brain regions involved in different story reading subprocesses," *PLOS ONE*, vol. 9, no. 11, p. e112575, Nov. 2014.