Generating Rich Image Descriptions from Localized Attention

School of Computer Science & Applied Mathematics
University of the Witwatersrand

David Poulton
1662476

Supervised by Dr Richard Klein

July 26, 2023

A dissertation submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Computer Science

Declaration

I, David Poulton, hereby declare the contents of this research report to be my own work. This thesis is submitted for the degree of Master of Science in Computer Science at the University of the Witwatersrand. This work has not been submitted to any other university, or for any other degree.

Signature

July 26, 2023

Date

Acknowledgements

This research would not have been possible without extensive support from various institutions and persons.

For training and evaluation of our models we make extensive use of the University of the Witwatersrand's Mathematical Sciences Cluster as well as the PRIME computational hardware. Funding for these clusters and devices was supported by the National Research Foundation of South Africa (Grant Numbers: 118075 and 117808). The authors also acknowledge the Centre for High Performance Computing (CHPC), South Africa, for providing computational resources to this research project under CSCI1340.

Alongside this, I am grateful for the extensive personal support of my supervisor, family, friends, peers, and colleagues, which aided my progress through this research substantially.

Lastly, I owe great thanks to the willingness of previous researchers to share openly their work upon which mine was based, and to the many, many institutions who make available high quality tools which were in some way used to conduct this research.

Abstract

The field of image captioning is constantly growing with swathes of new methodologies, performance leaps, datasets, and challenges. One new challenge is the task of long-text image description. While the vast majority of research has focused on short captions for images with only short phrases or sentences, new research and the recently released Localized Narratives dataset have pushed this to rich, paragraph length descriptions. In this work we perform additional research to grow the sub-field of long-text image descriptions and determine the viability of our new methods. We experiment with a variety of progressively more complex LSTM and Transformer-based approaches, utilising human-generated localised attention traces and image data to generate suitable captions, and evaluate these methods on a suite of common language evaluation metrics. We find that LSTM-based approaches are not well suited to the task, and under-perform Transformer-based implementations on our metric suite while also proving substantially more demanding to train. On the other hand, we find that our Transformer-based methods are well capable of generating captions with rich focus over all regions of the image and in a grammatically sound manner, with our most complex model outperforming existing approaches on our metric suite.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Image captioning is a sub-field of machine learning that sits between Computer Vision (CV) and Natural Language Processing (NLP), where the primary task is to generate a description or caption of the prominent features of an image. These captions can range from short phrases of several words that identify only prominent aspects, to longer paragraph length descriptions that capture a range of detail from the image. There are a variety of use-cases for such captioning systems. Models capable of connecting visual and linguistic understanding can be useful for purposes such as knowledge transfer, or in downstream tasks such as cycle-consistency in image generation systems – where a loss can be defined for images that yield different captions to those used to generated them. Image indexing and searching methods can also benefit from these models, as images can be captioned by an automated system allowing users to easily find them through descriptions. More human-centric use cases also exist such as accessibility improvements for visually-impaired users, or as a learning aid for new languages through visual learning.

However, the captioning task is particularly challenging for several reasons. From the architectural perspective, models need to develop both the ability to identify visual features in an image, as well as an understanding of language in order to convert those visual features into a coherent sentence or paragraph. As such, systems must handle multi-modal sequences and captioning models tend to contain large numbers of parameters. Additionally, the computational and data requirements to train these large captioning models is substantial, and the effort to produce high-quality datasets equally so. Lastly, salient features of an image and the words we use to describe them are heavily subjective, and both training the model to generate "good" captions and evaluating those captions quantitatively becomes difficult. Several evaluation metrics exist and each introduces rules to lessen the impact of subjectivity. However, subjectivity still poses a substantial challenge.

The effect of subjectivity can be reduced by guiding the captioning system to specific regions of the image through methods such as saliency maps or visual attention traces, rather than relying on the model to determine important aspects of the image. This allows the model to target certain regions of the image and provide more specific wording which eases the difficulty of evaluating the captions fairly. At the same time, this also

introduces additional complexity into the models as it introduces another modality, and further limits the choice of available datasets.

Thus, most captioning models are only targeted at short phrases determined using only the image. The earliest models utilise hand-crafted features and fixed template outputs which made generalisation difficult and generated captions rigid [Sharma *et al.* 2020]. With the advent of deep learning, systems adapted to used learned visual features from Convolutional Neural Networks (CNNs) and were capable of dynamic length output captions owing to Recurrent Neural Networks (RNNs) [Hossain *et al.* 2019]. The addition of the attention mechanism then improved this combination, allowing models to develop their own understanding of salient regions, and further experimentation began with human based attention as well. This led to more performant models, however RNNs were slow and difficult to train and captions were still limited to relatively short phrases.

The advent of the Transformer model eased these limitations greatly. Transformer-based models by nature were substantially faster to train, and were capable of handling longer captions and more complex visual-linguistic relations. As such, more focus has gone towards models generating longer descriptions and utilising richer forms of visual attention, and additional datasets have been created to fulfill these requirements.

In this dissertation, we present our work expanding upon existing long-text image captioning models through both RNN and Transformer-based architectures. Our systems make use of visual regions from images, alongside attention data in the form of temporally aligned caption-trace sequences generated by humans, in order to produce captions. Our final novel Transformer-based model is capable of generating full paragraph length descriptions that are coherent and highly descriptive of the image. We provide the series of Transformer-based models leading to our final architecture, and discuss how each attempted modification impacted the performance of the model over its previous form. We additionally show that RNN based methods may not be suitable for this task after various attempts to develop a suitable model, and this belief is supported by a lack of any existing RNN based architectures for this task.

A brief overview of this dissertation is as follows. In Chapter 2 we provide a more detailed background for the task, discussing sequence-to-sequence systems and their extension into captioning approaches, alongside the various architectures and methods that exist for the problem. In chapter 3 we then discuss related work for the task of long-text image description, especially relying on human attention, alongside research into visual grounding and cross-modal tasks. We then introduce our research questions and hypotheses in Chapter 4, and provide details of our data requirements and processing, as well as evaluation metrics we will be using. We provide finer details of our LSTM-based and Transformer-based architectures in Chapter 5, alongside any processes specific to each format. The results of these models are then covered in Chapter 6, both quantitatively and qualitatively, alongside an ablative analysis of our Cross-Modal Encoder mechanism. Finally, in Chapter 7 we provide a concluding discussion and summary, and provide several topics for possible future work in this task.

Chapter 2

Background

Image captioning literature is extensive owing to the age of the task and the broad range of methods available to researchers. In this chapter we attempt to provide a high level overview of the field and some related topics. We first cover the primary concepts of sequence systems, such as encoder-decoders, attention, and teacher forcing, which are largely relevant to image captioning amongst other use cases. The RNN and LSTM architectures which are used extensively in NLP tasks are also covered, alongside their advantages and limitations, followed by details of the more recent Transformer architecture and how it improves over the RNN format. We then provide further background on the task of image captioning itself with focus on existing literature and widely used methodologies, and finally provide a summary of the background.

2.1 Sequence Systems

Sequence system tasks are any tasks involving the conversion of one sequence of tokens, or document, to another. A vast number of different challenges exist in this category, such as translation of sentences from one language to another, sentiment classification of a paragraph, consuming share price histories in order to generate future estimates, and others. A visualization of some of the forms of sequence tasks can be seen in Figure 2.1.

For the case of guided image captioning we are primarily interested in systems that consume the entire input sequence before producing the relevant output sequence. As such, we will first be covering the concept of Encoder-Decoders, or Seq2Seq, systems. We will then discuss the concept of attention and how it is used in sequence tasks, and finally the concept of teacher forcing and how it improves training characteristics of Seq2Seq models.

2.1.1 Encoder-Decoders

Encoder-Decoder systems, also called Seq2Seq systems, were introduced by Sutskever *et al.* [2014]. Seq2Seq systems are designed to consume an input sequence of tokens

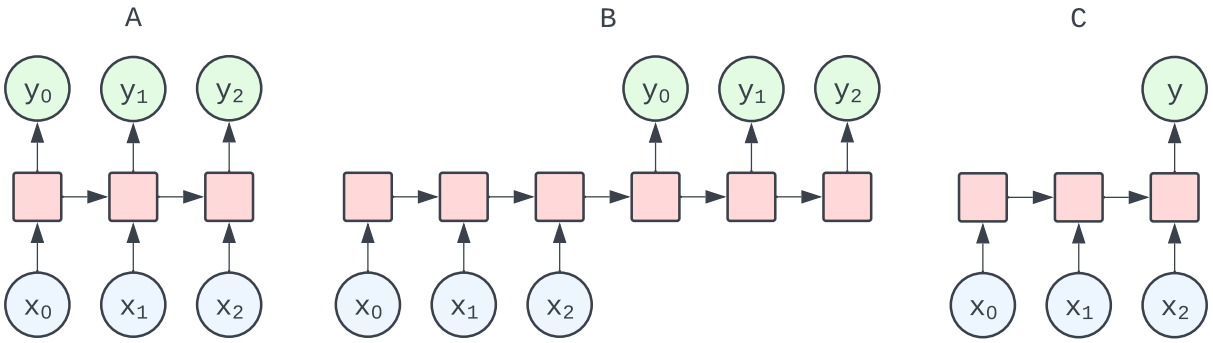


Figure 2.1: A visualisation of how sequence based tasks can vary. From left to right, systems can either consume and produce tokens concurrently, consume all input tokens and then produce all output tokens, or consume all input tokens and produce a single classification token, amongst other approaches. Inspired by Karpathy [2015].

using an encoder segment, and produce an output sequence using an independent decoder segment. The encoder segment is responsible for converting the input sequence into a latent memory tensor representing the context of the sequence, which it does by processing a single token at a time and updating its current memory state. This memory tensor is then provided to the decoder segment in order to progressively generate the output sequence. The decoder will also iteratively modify the memory state as it outputs tokens, and generally will also consume the previous output token in order to generate the subsequent one. The use of the previous tokens allows the system to finely guide the generation of each token to ensure it is cohesive with the immediately preceding tokens, while still maintaining the global state in its memory.

The advantage of utilising independent encoder and decoder segments is that each segment is now able to learn weights specific to its sub-task. Initially both of these segments were LSTM-based, discussed later, however newer systems can be composed of any number of suitable architectures. If separate encoder and decoder segments were not used, a single system would need to learn how to consume both input and prediction tokens, generate those prediction tokens, and build up a latent representation of the input before modifying it progressively throughout the output sequence. Thus, breaking the process into two stages avoids overloading any particular segment. Additionally, it would allow for different encoder-decoder pairings to be made from pre-trained weights without having to retrain the combined system for each possible input and output pairing – provided all encoders and decoders are conditioned to use the same latent embedding space for the memory tensors.

2.1.2 Sequence Attention

An encoder system provides the accumulated memory representation of the input sequence in a single tensor to the decoder, which proved effective for many initial Seq2Seq tasks. However, as systems became more complex and sequences became longer, having a fixed-size memory representation to generate from became a limitation. Fixed-size representations could become overloaded when sequences became too long, and

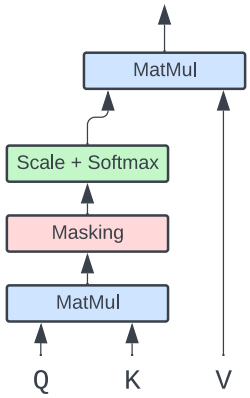


Figure 2.2: A diagram of the Scaled Dot-Product attention mechanism. Q is the query matrix, K the key matrix, and V the value matrix.

additionally forced the decoder into utilising the same combined latent space as the encoder.

Thus, Bahdanau *et al.* [2014] introduces the concept of the Attention Mechanism. Attention mechanisms allow a system to take a variable length sequence and attend to each element of the sequence individually, assigning each element with a specific weight – called a soft weight. Next the sequence is combined into a fixed-sized tensor by taking the sum over each element – or an embedding thereof – which is weighted by the soft weights generated in the attending step. In encoder-decoders, this now allows the decoder to determine which elements of the input sequence are most relevant to the current state of the system, and use that information in order to generate the next token.

While many variants of attention mechanism exist, the most relevant form of attention for our work is Scaled Dot-Product Attention, popularised by the work of Vaswani *et al.* [2017]. We refer to this simply as attention for the purpose of brevity in the remainder of this report. The attention system requires three inputs matrices, Q, K, and V. Q represents one or more query elements, and K the sequence of keys, which are used together to generate the soft weights. V is the sequence of values, which is summed using the generated soft weights in order to produce the final fixed-size tensor. A diagram of the Scaled Dot-Product Attention mechanism can be seen in Figure 2.2, and the formulation is given as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

Where $\sqrt{d_k}$ is the hidden size of the system. Not included in the formulation is an additional masking procedure. The masking procedure involves forcibly setting certain soft weights to zero where attending to certain elements should not be allowed. This is useful, for example, if we do not want our model to attend to padding or masking tokens, or if we want to enforce limitations of which parts of the sequence our decoder can attend to. One common application of this is the masking of "future" tokens in

generative tasks, where the model is only allowed to consume tokens up to the current time step to avoid information leakage.

Additionally, the query, key, and activation matrices are often embedded before being provided to the attention mechanism, using a linear feed-forward layer for each embedding. This allows the attention mechanism to produce an output guided from the value matrix rather than simply as a weighted composition of it. This also means the query and key elements do not need to be similar to each other in the latent space of the decoder, only in the embedding space of the attention mechanism [Vaswani *et al.* 2017]. In encoder-decoder systems, the query for each time step will most often be from the last output token, and both the key and value matrices will be composed of the hidden states after each time step from the decoder. Thus the decoder now has easier access to more information from the encoder as well as the ability to accumulate it as needed, and the encoder is no longer forced to accumulate all the information into any single fixed-sized tensor.

2.1.3 Teacher Forcing

In the decoder segment of a Seq2Seq system, the decoder will often consume the output tokens it generates at each time step in order to generate the subsequent token, which improves the quality of predictions. This approach is known as sampling. However, utilising this approach during the training process can lead to volatile training paths, poor convergence, and even divergence due to exploding gradients. Initially while training, the model will produce inaccurate predictions for each token as the model weights would still be largely randomized. The model will then have difficulty in generating subsequent tokens, as it is being provided with non-sensible tokens to condition onto. Additionally the output tokens will change as the model trains, creating a moving target for the learning process to tend towards and introducing further instability.

A common solution to this challenge is to make use of teacher forcing. In teacher forcing, rather than utilising the previously generated token for each subsequent prediction, the actual ground truth token is provided to the generator. This now stabilises the training process as the generator will instead receive sensible input to condition on, and eliminates the moving target phenomenon. [Bengio *et al.* 2015] Additionally, it greatly accelerates training through parallelisation, as the system can now generate the entire training prediction sequence in a single pass rather than iteratively. This greatly improves the model's training characteristics, however introduces another complication. During the generation of new sequences, where ground truth data is not available, the model will now be required to make use of sampling rather than teacher forcing. In cases where the model can generate the correct output token this is not a problem, however if the model generates an incorrect token for any time step it will impact the remainder of the sequence as well. Because the model was not trained with sampling it will likely be sensitive to this anomaly, and degrade the quality of the subsequent predictions. One method of minimizing this is to utilise scheduled sampling, where the model will either receive sampled input or ground truth input randomly with certain probabilities. This can make the model more robust to perturbations, however also reintroduces the problem of training instability. Professor Forcing [Lamb *et al.* 2016]

is another method that also attempts to solve this problem by using a discriminator RNN during training to eliminate bad predictions impacting the subsequent training steps. However, this introduces an additional component to train and still does not fully eliminate the problem.

2.1.4 Beam Search

As discussed, a common feature of inference in decoders is the reliance on the previous token generated by the decoder, which is used to condition the next token to be generated. As such, the model greedily explores a single path in the tree of possible generated token sequences – once a token is generated, we are limited to the paths containing that token. This can lead to scenarios where the model chooses the "best" token at each time step, however generates a sub-optimal sequence by greedily selecting early tokens that lead to lower probability sequences. This is as opposed to selecting early tokens – with lower greedy probabilities – that lead to the final sequence with the highest global probability.

Thus, the concept of beam search was introduced, and later utilised in machine learning approaches by Graves [2012]. Beam search involves generating B different sequences in parallel, and retaining the set of sequences with the highest global probability thus far in the inference, where B is the so-called beam width. At each time step, the model generates the probabilities for the next token for each of B sequences, and determines the next B sequences to keep by taking those with the highest probabilities – analogous to casting a narrow beam of light through the solution space. This sequence probability is calculated by multiplying the probability of the sequence thus far by the probability of the next token from that sequence. Thus, while beam search is still a greedy algorithm, it increases the chance of finding a more globally optimal sequence than pure greedy inference through wider exploration.

The downside of this approach is the proportional increase in computational demand. Increasing the size of the beam width also increases the number of sequences the model must retain and generate for, requiring more memory and more compute per time step. Additionally, increasing the beam width will have diminishing returns for higher values of B. Thus, generally B is chosen to maximise the accuracy of the model while limiting the introduced computational overhead.

2.2 Recurrent Networks

Most conventional neural networks are simply Feed-Forward Neural Networks, where tensors are passed linearly from one layer to the next until the model outputs a final value. This format is useful for fixed-sized inputs and outputs, however, does not function as well when the input or output size is unknown or varying – such as many NLP cases where phrases or sentences are of different lengths. RNNs solve this problem by allowing tensors to repeatedly pass through the same layer of a model several times, allowing consumption and generation of sequences of arbitrary lengths. This is done by maintaining a hidden state, representing the memory of the model. Each recurrent

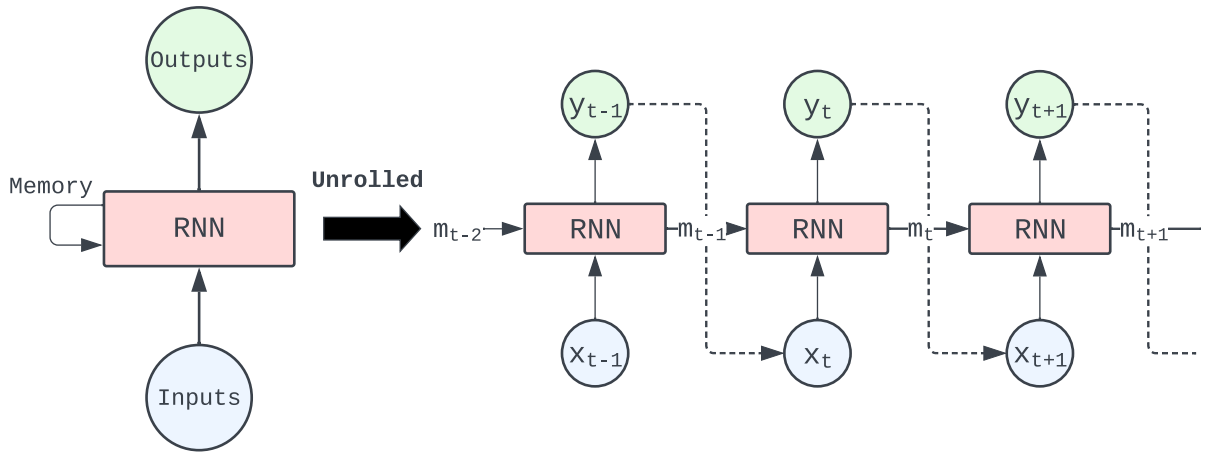


Figure 2.3: A diagram of a traditional RNN architecture. The left of the image displays the compressed structure, where the right displays the same structure however now unrolled through time. At each time step the layer is provided a new token as well as the memory of the previous step to generate an output token, and update the memory state. Inspired by Olah [2015].

unit has its own hidden state, such that repeated passes iteratively change the state of the model and allow it to produce varying results after each repetition – even using the same weights. This can be improved by using LSTMs (Long Short-term Memory), extensions of RNNs. LSTMs introduce additional gates for the retention and updating of memory, which improves the training and evaluation characteristics of models.

2.2.1 RNNs

The vanilla RNN architecture was introduced by Rumelhart *et al.* [1986]. As mentioned prior, it relies upon recurrent layers, or units, that are passed through multiple times during training and evaluation, using hidden state to store information of the current state of the model. One can view each of the repetitions of the recurrent layers as a single step in time, with an associated hidden state, input, and output. As such, during training model weights are updated using backpropagation through time (BPTT) rather than conventional backpropagation. A diagram of a traditional recurrent architecture can be seen in Figure 2.3

For each time step t, each unit receives an input tensor x^t and the accumulated hidden state tensor h^{t-1} from the previous step, and produces an updated representation of the hidden state h^t, alongside an output tensor y^t. To produce the output tensor the RNN will first combine the previous hidden state with the new input tensor to generate the new hidden state. This new hidden state is then used to generate the output, and continues on to the next layer. A traditional RNN recurrent unit can be formalised as:

$$h^t = a_h(W_{hh}h^{t-1} + W_{hx}x^t + b_h), \tag{2.2}$$
$$y^t = a_y(W_{yh}h^t + b_y), \tag{2.3}$$

where a_h and a_y are the activation functions of the hidden state and output – frequently tanh, Ws represent learned weights, and bs represent bias terms. For the initial time

step the hidden state can either be generated with a default value – commonly a zero tensor – or through learned weights. Allowing a learned state initialisation can act as a form of bias for the sequence, and can provide a global context from the training data. These learned initialisation can also be conditioned by parameters specific to the sample to provide guidance to the model.

Yet there are still three substantial problems with the traditional RNN architecture – namely "forgetting" earlier tokens, vanishing gradients, and exploding gradients [Pascanu *et al.* 2013; Hochreiter and Schmidhuber 1997]. The problem of exploding gradients is in no way limited to RNNs, however the repeated use of the same weights and long unrolled sequences make RNNs particularly sensitive to it. Forgetting memory and vanishing gradients are, however, more relevant limitations of the architecture. The problem of forgetting memory appears when long sequences are provided to the model. While the RNN hidden state is able to retain previously seen information, it is modified on every recurrence and will begin to degrade as the number of repetitions grows, and important information is overwritten. Additionally, there is only a limited amount of information that can viably be retained in the fixed-sized state tensor, and these combined effects can result in information from earlier in the sequence being lost. Similarly, vanishing gradients will begin to appear in long sequences as well. The high number of repetitions through the recurrent layers, as with any very deep network, causes gradients to approach zero as they propagate backwards. Naturally this will hinder the model from updating effectively from the earliest tokens in the sequence and lead to slow or sub-optimal convergence.

2.2.2 LSTMs

LSTMs (Long Short-term Memory) are a variant of recurrent networks, and an improvement over the traditional RNN architecture, introduced by Hochreiter and Schmidhuber [1997]. Traditional RNN units contain only straightforward weight multiplications and activations on the inputs and hidden states in order to produce the output and next hidden state. As discussed prior, this results in problems such as vanishing gradients and "forgetting" of tokens over long sequences. LSTMs mitigate these problems with the introduction of a series of gates, and an additional form of memory called the cell state, that control the flow of information through the model alongside the existing hidden state. These three gates – namely the input gate, output gate, and forget gate – are each able to restrict certain information from propagating onward past the unit to the rest of the sequence. They do so by each acting upon the inputs and previous hidden state of the unit in order to selectively modify the previous cell state. This new cell state is used in turn to modify the hidden state. Unlike traditional RNNs, there is also no specific output tensor, and instead the hidden state acts as both the output at each time step as well as a form of memory passed to the next unit. A diagram of an LSTM recurrent unit can be seen in Figure 2.4.

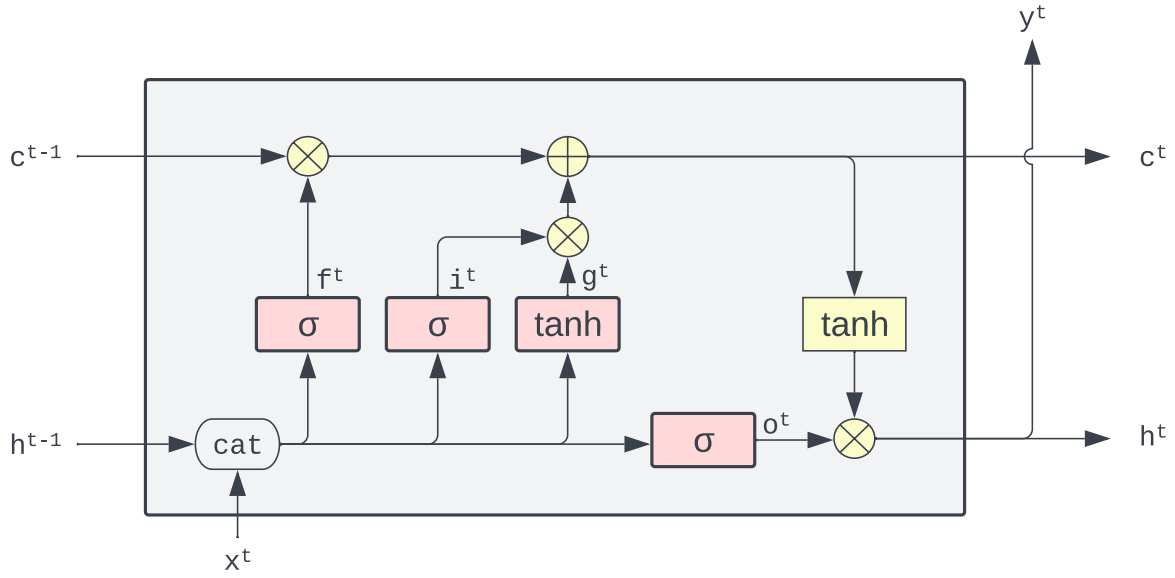


Figure 2.4: A diagram of a typical LSTM recurrent unit. Pink cells indicate fully connected non-linear layers, yellow cells indicate plain activations or numeric operations, cat is the concatenation operation.

The formulation of an LSTM recurrent layer is given as:

$$f^t = \sigma(W_{fx}x^t + W_{fh}h^{t-1} + b_f) \tag{2.4}$$
$$i^t = \sigma(W_{ix}x^t + W_{ih}h^{t-1} + b_i) \tag{2.5}$$
$$g^t = \tanh(W_{gx}x^t + W_{gh}h^{t-1} + b_g) \tag{2.6}$$
$$o^t = \sigma(W_{ox}x^t + W_{oh}h^{t-1} + b_o) \tag{2.7}$$
$$c^t = f^t \odot c^{t-1} + i^t \odot g^t \tag{2.8}$$
$$h^t = o^t \odot \tanh(c^t), \tag{2.9}$$

where f^t, i^t, o^t represent the forget, input, and output activations, g^t represents the cell gate, and c^t and h^t represent the cell state and hidden state, all at time step t. Ws and bs represent the associated weights and biases of each gate. σ represents the sigmoid activation, and $\odot$ represents the Hadamard product.

By "gating" several paths through the unit, it allows the LSTM to remove information from the cell and hidden states that it no longer needs, and also avoid overwriting information that it does. First, the forget gate determines what information to remove from the cell state. The input gate then generates additional information to be placed into the cell state, which is then filtered by the cell gate before actually being added to the cell state. The updated cell state is then used with the output gate activation to modify the new hidden state. The cell state and hidden state are then propagated to the next time step, and the hidden state is utilised as the output for the time step.

By allowing the model to purposefully remove unnecessary information from the cell state, and only add relevant information at each time step, LSTMs largely mitigate the impact of long sequences on memory loss. This ability to retain more relevant data

also makes LSTMs better than traditional RNNs at handling delays between important tokens in the sequence, and makes them less sensitive to the distance between relevant tokens. Finally, because information can now flow unmodified through certain paths of the network, primarily via the cell state, the vanishing gradient concern is reduced. Modifying the values along these paths less results in gradients back-propagating relatively unchanged through the model, allowing them to reach earlier time steps with larger magnitudes, and improve the convergence characteristics during training. Another common recurrent architecture called GRU (Gated Recurrent Units) [Cho *et al.* 2014] has been proposed more recently with a similar style to LSTMs that retains these advantages. GRUs utilise only two gates rather than three as with LSTMs, and differ slightly in how they propagate information. As such they are less complex than LSTMs, albeit at the cost of a slight performance reduction.

However, there are still issues that remain with all of these recurrent architectures. Certain paths through the LSTMs, as with traditional RNNs and even GRUs, can still cause gradient explosions due to repeated multiplications, making the architectures more sensitive to training conditions. Additionally, the need to pass through every token in a recurrent manner limits the opportunity for parallelisation, and results in long training and inference times. Training also requires a non-trivial amount of computational resources, especially RAM, as in some cases the gradient information for each token in the entire sequence must be retained before back-propagation can occur.

2.3 Transformers

The Transformer architecture [Vaswani *et al.* 2017] is a Seq2Seq system that improves on conventional RNN-based systems, and as such has both an encoder and decoder segment, which we later refer to as the Transformer Encoder and Transformer Decoder. It relies extensively upon the concept of attention mechanisms (discussed in Section 2.1.2) in both its encoder and decoder segments, and introduces new forms of sequence attention known as self-attention, and multi-headed attention. This self-attention replaces recurrence in the encoder entirely by utilising attention from each token in the sequence over the rest of the sequence, allowing processing of all tokens in parallel. It also replaces recurrence in the decoder in a similar manner during training, however during inference decoding is often still recurrent by necessity. This modification makes them substantially faster and less computationally demanding for both training and inference in the majority of cases. However, because the encoding and decoding happens simultaneously for all tokens in the sequence the model loses the temporal ordering that the RNN architectures had. Naturally this is an important characteristic, and so positional encoding is applied to the sequences as well, before they enter the encoder or decoder blocks. This positional encoding initially served to replace the lost temporal information, however can also be used to provide any form of order to the sequence, or excluded where not suitable. A diagram of the original Transformer architecture can be seen in Figure 2.5.

We will first discuss the new self-attention and multi-head attention systems introduced by Vaswani *et al.* [2017] and the motivations for their use, and provide further detail

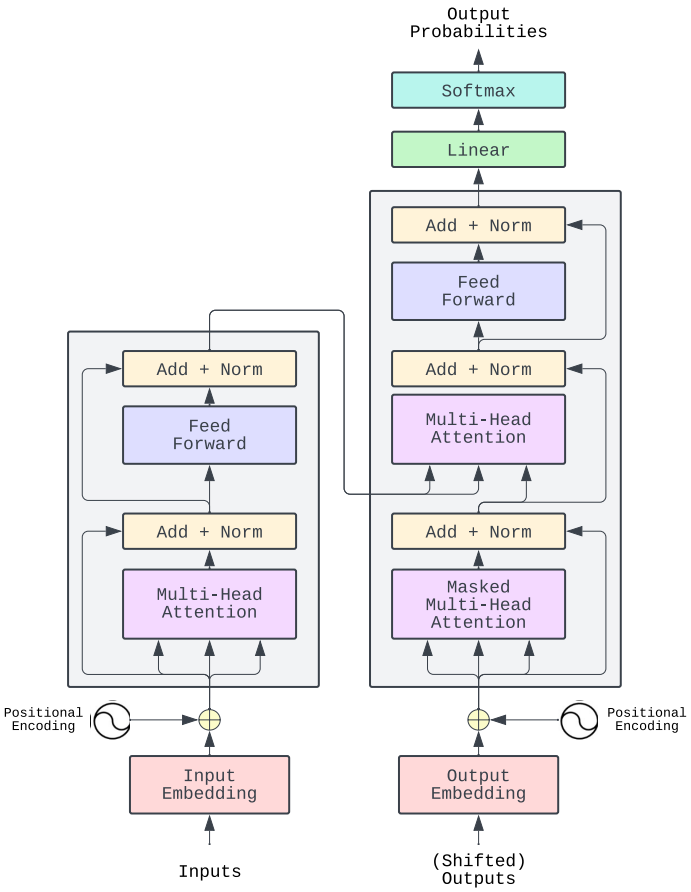


Figure 2.5: A diagram of the original Transformer architecture, based on that found in Vaswani *et al.* [2017]. The left block acts as the encoder, and the right block as the decoder. Both blocks can be repeated an arbitrary number of times.

on the positional encoding process utilised by the Transformer system. With this understanding we then provide a more detailed overview of the Transformer architecture, beginning with a breakdown of the Transformer Encoder and its encoding process, before moving on to the closely related Transformer Decoder.

2.3.1 Transformer Attention

The Transformer architecture introduces two new forms of attention, self-attention and multi-head attention. Self-attention is functionally the same as regular attention, however in this case the query tokens and the key/value tokens are from the same sequence. This is as opposed to the more conventional cross-attention, where the query tokens and the key/value tokens are sourced from different sequences.

Multi-Head Attention, or MHA, is however a functional modification of the attention mechanism. The previously discussed attention mechanisms in equation (2.1) act on the query, key, and value matrices in their entirety. However, this limits the ability of the attention mechanism to learn different representations or features from the hidden representation space of the model. Thus, Multi-Head Attention splits each of the query,

key, and value matrices into h evenly sized segments along the dimension of the hidden space, otherwise referred to as the dimension of the model, or d_{model}. This is formulated as:

$$\text{MHA}(Q, K, V) = \text{concat}(\text{head}_0, \text{head}_1, ..., \text{head}_h)W^O \tag{2.10}$$

$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.11}$$

Where $W^Q, W^K \in \mathbb{R}^{d_{model} \times d_k}$, $W^V \in \mathbb{R}^{d_{model} \times d_v}$, and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$. Additionally, h is the number of heads, or subspaces, $d_k = d_{model}/h$, and in the case of the original Transformer, and the remainder of this work $d_k = d_v$. Each of the W matrices are projections from the hidden spaces of each input to a subspace of the attention mechanism, or in the case of W^O from the attention subspaces back to the hidden subspace of the model. Utilising multiple heads in the attention mechanism is conceptually similar to utilising multiple kernels in the layers of a Convolutional Neural Network (CNN). Each head can now act within its own subspace, further removing the coupling between the hidden space of the model and the attention mechanism, and allowing each head to specialize to certain features. This multi-headed approach does introduce some extra computation due to the projections to and from subspaces, however due to each head having a proportionally smaller dimensionality, the overall complexity remains similar.

2.3.2 Positional Encoding

In RNNs the tokens of a sequence are provided to the encoder or decoder one per time step, allowing the model to build a temporal ordering of the sequence by modifying its current memory. However, as alluded to previously, Transformer systems do not share this trait and in most cases require a predefined positional encoding in order to function effectively. This is because attention mechanisms – and by extension the Transformer system – do not have an inherent ordering of their inputs, and so this information must be instilled in the sequence itself. To achieve this, a positional encoding is applied to the embedded sequences. In the original paper, the authors apply a fixed sinusoidal encoding and state that a learned encoding does not perform better, however a number of different positional encodings can be used depending on the requirements. Each dimension in the hidden space of the model is encoded using a different frequency sine wave, modified for each position in the sequence. The sinusoidal positional encoding is defined as:

$$\text{PE}_{pos,2i} = sin(\frac{pos}{10000^{2i/d_{model}}}) \tag{2.12}$$

$$\text{PE}_{pos,2i+1} = cos(\frac{pos}{10000^{2i/d_{model}}}) \tag{2.13}$$

where pos represents the position of the token in the sequence, and i the index of the embedding dimension. As such, even indexed dimensions are encoded using the sin function, and odd dimensions using the cos function. The advantage of such an encoding is that relative distances within the sequence can easily be determined by difference of the positional encodings for any two indexes, i.e. PE_{pos+k} can be represented by a linear function of PE_{pos}. This allows the model to more easily attend to tokens in the sequence at relative positions from the current token. Additionally, this should

make the model more robust to sequences of lengths not seen while training during inference.

2.3.3 Transformer Encoder

As implied by its name, the Transformer Encoder is responsible for taking in an input sequence and converting it to a format useable by the decoder segment. As the Transformer system uses attention extensively, the encoder does not simply accumulate the entire sequence into a fixed-size memory tensor, but instead converts it to a memory sequence of encodings of the same size as the original input sequence. First the input sequence is embedded to the hidden size of the model, d_{model} as before, using a learned embedding layer. The sinusoidal positional encoding discussed previously is then generated for the sequence and element-wise summed with the initial embedding. The embedding is then provided to the Transformer encoder in this state.

The Transformer Encoder is composed of layers, or blocks, which can be repeated an arbitrary number of times, enabling larger and more complex models if necessary. Each encoder layer is composed of two sub-layers, namely a multi-head attention layer and a two layer feed-forward network (FFN). Each sub-layer is also surrounded by a skip connection followed by a Layer Normalization (LN) [Ba *et al.* 2016]. The skip connection allows the model to retain information after modifications in each layer, and also allows gradients to flow more easily through the network during back-propagation, while the Layer Normalization improves the training stability of the model and reduces training times. The feed-forward network is defined as:

$$\text{FFN}(x) = \max(0, xW_0 + b_0)W_1 + b_1 \tag{2.14}$$

In the original architecture the internal size of the FFN is $d_{ff} = 2048$, however this can be arbitrarily chosen, while the input and output sizes are both that of the model, d_{model}. The formulation of the entire encoder layer is:

$$x_{att} = \text{LN}(\text{MHA}(x, x, x) + x) \tag{2.15}$$

$$h = \text{LN}(\text{FFN}(x_{att}) + x_{att}) \tag{2.16}$$

where x is the sequence embedding provided to the layer, and h is the encoder memory, or generated encoding. This memory is then provided to the next layer in the encoder if one exists, or to the decoder segment for decoding.

2.3.4 Transformer Decoder

The Transformer Decoder is largely the same as its partner encoder, however differs slightly from the decoder concept in the original Seq2Seq format. Whereas the original Seq2Seq decoders directly modify the memory received from the encoder in order to generate a sequence, the Transformer Decoder only utilises the encoder memory to guide the decoding process, and instead modifies the shifted output sequence provided to it. Additionally, because the architecture is not recurrent, the decoder applies the decoding process to all tokens simultaneously, and as such the generated sequence will always be of the same length as the shifted output sequence provided to the decoder.

We note the possible confusion in the "output" sequence being provided to the decoder as input. This is because of the sampling concept discussed prior in Section 2.2.1, where each generated token is provided back to the decoder in order to condition the subsequent token. Thus, during training, and if using teacher forcing (discussed in Section 2.1.3), we provide the entire ground truth sequence to the model such that each ground truth token will condition the token that should follow it. Alternatively, if not using teacher forcing, or during inference, we provide the sequence generated so far and repeat the process until a stop token is generated. As such, we refer to the output sequence provided to the decoder as a "right shifted" output sequence, as each output token conditions the token generated to the right, or after it.

With this in mind, we can now detail the decoding process itself. As with the encoder, first we take the provided shifted output sequence and pass it through a learned embedding layer, and provide the same positional encoding process to yield the initial embedded features. This embedded sequence is then provided to the first Transformer Decoder layer, alongside the memory from the final encoder layer. And as with the encoder, the decoder can be comprised of an arbitrary number of layers. Each layer of the decoder is, however, now comprised of three sub-layers instead of two. The first sub-layer is a masked multi-headed attention layer for self-attention, and the second another multi-headed attention layer, however now for cross-attention over the encoder memory. The final sub-layer is a two layer FFN constructed exactly as in the decoder, and each sub-layer is again surrounded by a skip connection and Layer Normalization. It should also be noted that all decoder layers receive the same memory sequence, and only from the final encoder layer.

We also pause to stress that the first sub-layer has masked attention rather than full attention. This is because we provide the shifted output sequence to the decoder for conditioning. If the decoder were to perform self-attention unmasked it would simply learn to always produce the token immediately following the conditioning token when it trains, rather than relying on the encoded information or preceding tokens. Therefore the shifted output sequence is masked such that the self-attention layers may only attend to the conditioning token and tokens prior to it.

Continuing on, once the initial embedding and encoder memory are provided to the decoder, the embedding passes through the self-attention layer, which feeds into the cross-attention layer. The cross-attention layer then uses the embeddings as the query, and the encoder memory as the keys and values. After this the embeddings are fed to the FFN, and either passed to the next layer if one exists, or move to the prediction segment. In the prediction segment the processed embeddings are passed to a linear layer to project from the hidden size of the model to the same dimensionality as the target vocabulary. The softmax activation is then applied to this target tensor in order to provide the final word probabilities.

We formulate the entire process for each decoder layer as:

$$o^s_{att} = \text{LN}(\text{M-MHA}(o, o, o) + o) \tag{2.17}$$

$$o^c_{att} = \text{LN}(\text{MHA}(o^s_{att}, h, h) + o^s_{att}) \tag{2.18}$$

$$o' = \text{LN}(\text{FFN}(o^c_{att}) + o^c_{att}) \tag{2.19}$$

where o is the shifted output embedding provided to the decoder layer, M-MHA is masked-MHA, and o' is the decoded embedding. The process for the prediction segment is formulated as:

$$y' = \text{softmax}(W_o o' + b_o) \tag{2.20}$$

where W_o and b_o are the projection weights and bias, and y' is the word probability sequence.

2.4 Image Captioning

With the above information we can now directly address the task of image captioning. Image captioning involves the conversion of visual features into a sequence of words describing those features. As such, we can view image captioning as a form of the general sequence-to-sequence task, where we utilise an encoder to convert visual features to a latent representation, and a decoder to take that representation and generate the desired words.

The generation phase of the output sequences is largely the same between modern approaches – generally relying on an LSTM, or more recently, Transformer decoder. However, there is a wider variety in the visual encoding phase. As such, we will first discuss the broader formats that approaches utilise in their visual encoding, primarily the use of fixed-size and region based encoders. We will then specifically discuss the Bottom-Up Attention approach [Anderson *et al.* 2018] which has become widely used in state of the art captioning systems. We leave further detail for our upcoming Related Work, Chapter 3.

2.4.1 Visual Encoding

There are two primary approaches in the encoding process for visual features. The first approach involves consuming an image and converting it into a fixed-size tensor that can be provided to a recurrent decoder. The second and more recent approach involves converting the features of an image into a variable length sequence, which is provided as input into the encoder of a sequence-to-sequence model. Visualisations of the different approaches can be seen in Figure 2.6.

2.4.2 Fixed-Size Encoding

In this approach, the encoder often takes the form of a Fully Convolutional Neural Network which generates fixed-size outputs for any image, representing the most salient visual features. This approach is commonly referred to as one-step encoding, as the system takes a single, holistic pass over the entire image to generate the caption. The most common visual backbones for one-step captioning are the VGG [Simonyan and Zisserman 2014], and ResNet [He *et al.* 2016] family of architectures, due to their high accuracy. While these architectures were designed to generate image classifications, when used as visual backbones for captioning they are modified to remove the classification layer and instead output the image features for use in the decoders.

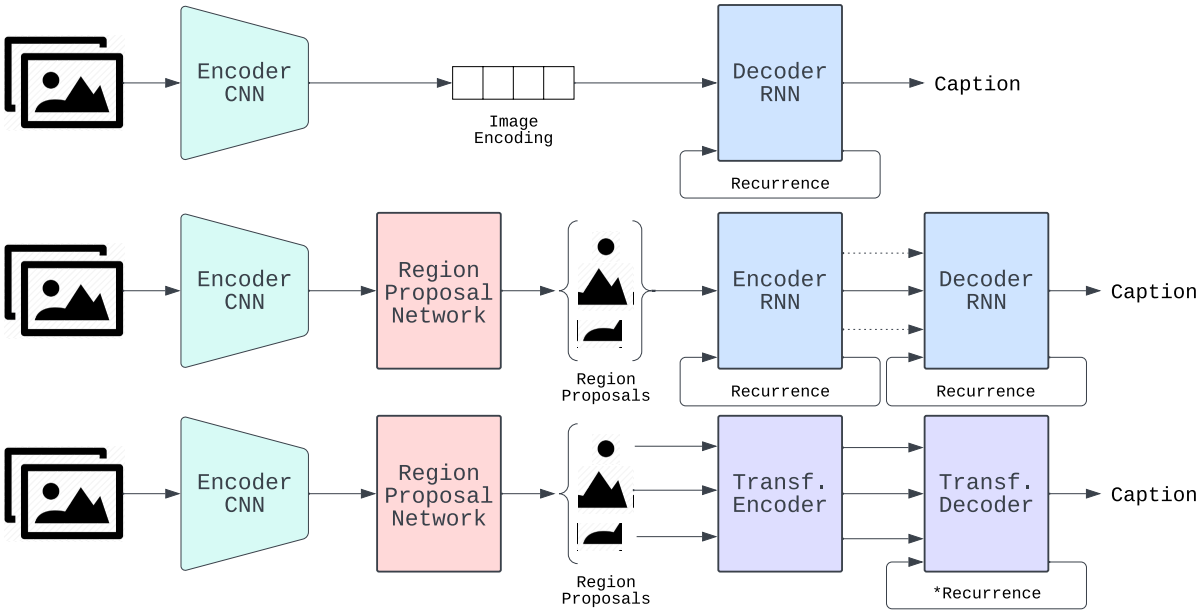


Figure 2.6: A visualisation of the most common visual encoding approaches for image captioning. From top to bottom; one-step image encoding using CNN with LSTM decoder, region proposal based recurrent encoder-decoder using RNN formats, region proposal partially-recurrent Transformer encoder-decoder. * *Transformer Decoders can be trained without recurrence using teacher forcing*.

One of, if not the earliest deep learning based image captioning systems is the Neural Image Caption (NIC) network [Vinyals *et al.* 2015b]. The authors make use of a straightforward CNN based encoding approach to generate the latent image encoding. This encoding was then provided to an unmodified LSTM to generate the output caption. While this approach is relatively basic by modern standards, it substantially outperformed SOTA methods at the time and introduced important concepts to the field. CNNs at the time were showing significant gains in vision based tasks, and similarly LSTMs were becoming SOTA in machine translation tasks. Vinyals *et al.* [2015b] took advantage of this, and set the precedent of a unified deep learning approach combining these two breakthroughs in deep learning.

New approaches quickly developed on the NIC architecture. Jia *et al.* [2015] makes use of a similar design, where a generic CNN converts the image into a latent encoding provided to a modified LSTM. They refer to their modified architecture as gLSTM, where a regular LSTM is modified to also utilise a constant global context tensor in the recurrent units to avoid the issue of memory loss over long sequences. This modification provides improvement over typical LSTMs at the cost of relatively little additional complexity, with captions becoming better structured by qualitative evaluation, and closer to ground truth in quantitative metrics. While they experiment with various guidance and hyper-parameter adjustments to the decoder, they do not experiment further with the visual encoding backbone.

Some approaches also go beyond still images for captioning, and extend this breakthrough format to video captioning. Donahue *et al.* [2015] introduce their Long-term

Recurrent Convolutional Network (LRCN) architecture which utilises a CNN based visual backbone that feeds into two layers of LSTMs rather than one. Each frame of the video is passed through the CNN, and the visual features are provided to the first LSTM layer. This first layer serves to accumulate the visual information that the model has been exposed to thus far in the video. The output of the first layer is then, at each time step, provided to the second LSTM layer, which then serves to accumulate the language information and generate words. This approach allows the system to retain visual information separately from language information, with the simultaneous encoding and decoding reducing the diminishing effect of long output sequences on the video information.

Vast quantities of further variations exist in the one-step encoding format – well beyond what we could reasonably address – and we do not commit too much time to this format as these approaches have largely become outdated. The primary cause of this outdating is that all of these approaches suffer from the same flaw, in that their encoded format is fixed-size and accumulated into a single tensor. As discussed in Section 2.1.2, sequence based attention substantially improves the ability of many sequence-to-sequence models by allowing the model to determine important information independently for each output token. By having the entire image encoded into a single tensor with a fixed size, the model is incapable of independent attention to different features of the image. As such, we progress onto the more relevant format of image sequence based captioning systems.

2.4.3 Sequence-Based Encoding

Image sequence based systems generally fall into one of two categories. The first category involves passing the output of the CNN backbone into a Region Proposal Network (RPN), which we refer to as region-based models. RPNs consume visual features and generate a variable length output of region proposals, where each proposal contains the coordinates it came from, the visual features specific to the region, and the confidence that the region is salient. The most frequently used region-based visual backbones are the R-CNN family of models [Girshick *et al.* 2014], and especially the Faster R-CNN variant Ren *et al.* [2015]. These regions are then provided to an encoder, which encodes the entire sequence of regions and passes it on to the decoder. The second and more recent category involves chunking the raw image into a series of patches, and passing them into a vision encoder, which we refer to as patch-based models. These patches can then either be passed through CNN encoders followed by a sequence encoder, or fully attention based encoders that act directly on the pixel-level information of each patch.

Region-Based Approaches

One of the earliest region-based approaches is the work of Karpathy and Fei-Fei [2015]. Their approach utilises a two stage process of region detection and alignment, followed by the description generation, which represent the encoder and decoder phase. First the image is provided to the model, and it is segmented using an R-CNN into the 19 most salient regions. During training each of these regions has an associated segment

of the description that must be aligned to it in some latent space, such that the model learns to associate the visual and language concepts. This process allows the model to compose a larger description by first focusing on the simpler sub-task of visually understanding each region, rather than attempting to represent the entire image in a single shot. The model then passes these latent region representations to an RNN, which uses them to generate a series of hidden states for each time step. The RNN is also provided the entire image as an initial conditioning in the first time step, and from there proceeds to generate the caption word-by-word similarly to other captioning systems. This combination of the holistic image representation and refined individual regions enables the model to capture both high-level and low-level details of the image, and allows it to build up to more complex descriptions in a stepped process. Their approach outperformed existing work on benchmarks at the time, and was the first to report on the MS COCO dataset.

The region-based DenseCap network [Johnson *et al.* 2016] improved upon this format by modifying the region processing segment. It utilises a VGG network as a visual backbone and passes the encoded image through a region proposal network similar to the Faster R-CNN format, using bi-linear interpolation instead of RoI pooling to allow gradient propagation into their coordinate system. These region proposals are then combined into a fixed-size region feature tensor and passed into an LSTM based decoder to generate the output caption. The use of bi-linear interpolation now means the region proposal segment can be trained and fine-tuned alongside the captioning system itself, improving accuracy over systems that utilise only pre-trained region proposals. However, this additional process naturally introduces a non-trivial overhead in the training cycle, and both of the above methods have yet to utilise attention over their regions.

The introduction of attention mechanisms for region-based captioning systems started with Xu *et al.* [2015]. Their work introduces two forms of visual attention, deterministic "soft" and stochastic "hard" attention. As with other models the system passes the image through a CNN backbone to generate a series of visual feature vectors, synonymous here with our usage of regions. The deterministic attention mechanism works similarly to the attention mechanisms discussed in Section 2.1.2. A soft weighting is applied over the different regions of the image, and the sum taken to provide the aggregated context vector for each time step in the caption generation phase. The stochastic attention mechanism instead selects a single region of the series to attend to for each time step during caption generation, rather than aggregating the series. Because it is stochastic it requires training through the REINFORCE [Williams 1992] algorithm rather than straightforward back-propagation, making it somewhat more complex to develop and train. Both approaches yielded similar improvements over existing SOTA approaches, however over time the deterministic approach would become the de facto standard for attention mechanisms.

However, some words are not always well suited to visual grounding. Words such as "the", "of", or "and" have no visual representation at all, and so trying to ground them to a region of the image often does not make sense. Additionally many words can be predicted based on solely the language representation so far, such as "phone" following

"talking on a cell" or "sign" after "at the red stop". As such, Lu *et al.* [2017] introduces a so-called visual sentinel to their language generation segment. This sentinel uses the current state of the LSTM language decoder to determine if the model needs to attend to the image or not. This removes the burden of having to visually ground abstract or unnecessary words, and lead to an improvement on all common captioning metrics over SOTA approaches at the time.

Patch-Based Approaches

However, the reliance of all of these models on region proposals from the images introduces an extra layer of dependency, and we now explore the more recent category of patch-based models. Patch-based approaches have been made substantially more viable by the advent of the Transformer architecture and its great flexibility. While Transformers were initially presented for language translation and transduction, they were then applied to captioning systems with visual backbones, and have also recently proven capable of directly acting on pixel values from raw images. We are currently unaware of any patch-based approaches in the image captioning task, however we cover some approaches in the related topic of image classification.

The first patch-based system, composed purely of Transformer components, is the Vision Transformer (ViT) model [Dosovitskiy *et al.* 2020]. The ViT system segments an input image into a variable number of 16×16 pixel patches, and each patch is projected individually to the hidden size of the model. The patches are then positionally encoded to provide locality, and passed into a conventional Transformer identical to the original format. The use of patches is necessitated by the $\Theta(n^2)$ complexity of the transformer encoder over the number of input tokens, as using every individual pixel as a token would be untenable with the resolution desired. This approach modestly improved on SOTA classification performance at the time, substantially reduced the amount of time required to train the model, and additionally reduced the impact of locality bias which is present by nature in CNN-based systems. This straightforward application of a Transformer to images without the need for an independent visual backbone highlights the strength of the architecture.

The next evolution of this was the CrossViT architecture [Chen *et al.* 2021]. The CrossViT approach is based on the original ViT architecture, however the encoding process involves two branches of different dimensionalities – one processing smaller patches and one processing larger patches. As the information flows through each layer of the encoder, an information sharing process is carried out by utilising a cross-encoding token from each branch. At each layer, the context token from one branch is swapped into the other and applies attention over the current embedding of the other branch, before being swapped back. In the final layer the context tokens are then taken from their sequences and combined, and used to generate the classification for the image. This swapping process allows each branch to attain information from the other branch through the context token, while still operating at its own latent size. Additionally, having two branches allows the model to focus on finer details in the smaller patches, and larger details from the larger patches. This provides a more holistic representation to the decoder, and these modifications further improve upon the ViT results.

The final approach we discuss here is the Swin Transformer design [Liu *et al.* 2021]. The Swin – or Shifted Windows – architecture takes in an image and segments it into regular patches. These patches are passed through multiple layers, where the patches are segmented into fixed-sized non-overlapping windows, and local attention is applied on the patches within each window. At the next layer, the windows will then be offset by half their size in each direction, and the same process is applied. Deeper into the model, neighbouring patches are also merged together, synonymous to down-scaling of the image resolution. The shifting process allows information sharing between windows as there is otherwise no overlap between them, and the down-scaling effect allows for the capture of larger scale details or objects. Both of these benefit the model while still limiting the local-attention computational demands. This approach to image encoding was then applied on various downstream tasks, where it showed improvements over conventional CNN-based backbones.

2.4.4 Bottom-Up Attention

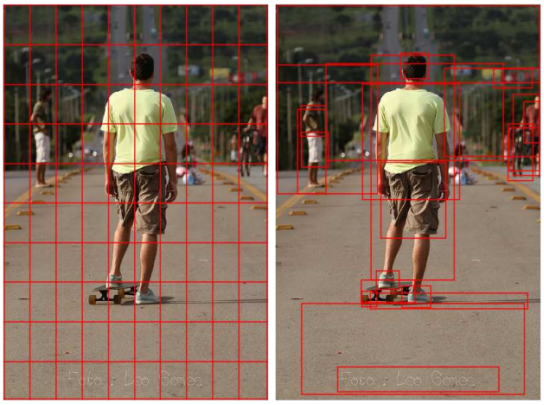
The Bottom-Up and Top-Down approach [Anderson *et al.* 2018] has become one of the most extensively used visual backbones for image captioning, visual question answering (VQA), and classification. It is composed of a Faster-RCNN [Ren *et al.* 2015] Bottom-Up segment for region proposal, and a dual-LSTM Top-Down segment for holistically aggregating these proposals and generating the captions. The motivation for this naming is that the Bottom-Up component provides regions to the system based only on the visual features of the image, without any information of the task at hand and in a sense being "low-level". The Top-Down component then takes these low-level regions and queries them using high-level pieces of information such as the state of the visual LSTM accumulator, the current question for VQA, or the state of the language generation LSTM. Example visuals for the system can be seen in Figure 2.7.

The reliance of the Bottom-Up portion on only the image itself – without any dependency on the task – means that the system is easily applicable to downstream tasks requiring visual backbones. As evidence of this the architecture yielded improvements on SOTA results at the time for the MS COCO image captioning challenge, and additionally was the best performing model in the 2017 VQA Challenge.

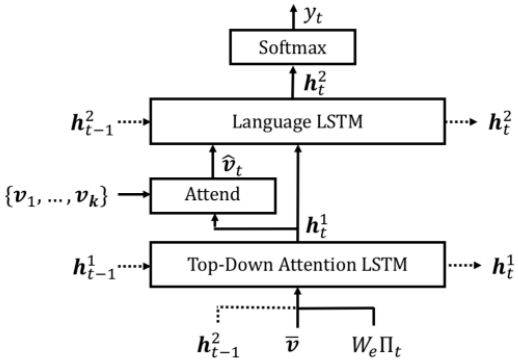
The primary strength of the Bottom-Up encoder is its use of variable size region proposals, as opposed to regions taken from a uniform grid. Dividing an image into a uniform grid will often split objects into less useful segments that are harder for the attention systems to account for, as it would need to learn to re-merge these segments into a meaningful object. On the other hand, variable sized regions allow the system to capture entire objects into a single region from the start, making it easier for the attention system to capture whole concepts within the image.

2.5 Summary

In this chapter we have provided a high level overview of the field of image captioning, and the supporting concepts building up to it. We have discussed the highly relevant



(a) An example image showcasing the Faster-RCNN region selection over naive grid-based patches.



(b) A visualisation of the Top-Down architecture used to accumulate visual information and generate captions from the region proposals.

Figure 2.7: Visuals for the Bottom-Up Top-Down system taken from Anderson *et al.* [2018].

topics of sequence-to-sequence systems and the common design of encoder-decoders. We also cover the implementation of attention within these systems and discuss its benefits, alongside the advantages and shortfalls of widely used teacher forcing.

We also introduce the commonly used recurrent architectures within image captioning, namely RNNs and LSTMs, and discuss their initial strengths and current weaknesses. This leads us to the widely used Transformer architecture, which we discuss in detail in its use of attention, positional encoding, and the overall structure, and how these concepts together largely eliminate many limitations of recurrent systems.

Finally we give a brief overview of image captioning systems and related concepts. Of primary interest is the diversity of image encoding systems, which we cover by discussing the initial use of fixed-size image encodings, before moving onto the more modern and performant use of region- and patch-based encoding systems. We also discuss the Bottom-Up Top-Down system specifically due to its pervasiveness in shared vision and NLP tasks, our work included.

Chapter 3

Related Work

In the discussion up to this point, captioning approaches have been limited to consuming only an image in order to generate the caption. While this approach has proven to work effectively for short captions covering the most salient regions, it has several shortfalls. The first shortfall is that these approaches are qualitatively limited in the length of output captions. As the captions become longer, they begin to lose grammatical correctness of their structure, and their semantic continuity. This is partly due to the design of the networks, and partly due to historic data scarcity, as a very limited number of datasets include long image descriptions. Additionally, many of these datasets are only recently released, which made training of previous long description models non-viable. The second, closely related issue, is that these longer captions become more susceptible to drift in the regions of the image they describe, the choice of words used to describe them, and the order in which they do so. While describing different regions and using different words does not necessarily imply incorrectness, it does make it substantially harder to objectively score these generated captions based on their quality.

As such, we now introduce the more closely related work for our research. We begin with an overview of the work that has been done in visual language localisation, covering some of the various localisation datasets, and some modern approaches towards it. We then cover the usage of human attention – a more directed form of localisation – both in broader fields and image captioning specifically. Given the importance of the Localized Narratives [Pont-Tuset *et al.* 2020] research to this work and its contribution towards long caption generation, we discuss their newly released dataset, as well as their first model capable of long text descriptions. Following this is the MITR work [Meng *et al.* 2021] which developed upon the Localized Narratives model with an improved architecture, and introduced new sub-tasks for the caption generation problem.

3.1 Visual Grounding

Visual grounding, or language localisation, is the process of grounding segments of a sentence or phrase to specific regions or visual features within an image. For example,

Figure 3.1: Example images and their annotations from the MS COCO dataset [Lin *et al.* 2014].

a human seeing the word “dog” appearing in a sentence could easily connect the word to a part of an associated image, and inversely could see a dog in an image and understand which segment of a phrase or sentence associates to it. Modern systems tasked with developing connections between visual and linguistic spaces often develop these connections without an explicit mechanism in the model by correlating the appearance of objects with the phrases that represent them.

3.1.1 Datasets

Various multi-modal datasets exist for approaches to use in developing visual and linguistic connections.

MS COCO

The MS COCO [Lin *et al.* 2014] dataset is one of the most widely used datasets for various tasks in Computer Vision and related fields, notably image captioning. The dataset is composed of approximately 120k labelled training and validation images, and an additional 40k test images with hidden labels. Importantly, most images are annotated with 5 short captions and object segmentation data, with some also containing human pose data and visual keypoint information. Some annotated example images from the dataset can be seen in Figure 3.1, note the short length of the captions.

SALICON

Extending on the MS COCO annotations, the SALICON dataset [Jiang *et al.* 2015] adds additional saliency annotations to 20k images from the COCO dataset, with 10k training, 5k validation, and 5k hidden testing samples. For data collection mouse traces were recorded from users as they viewed each image, and modified to best represent human viewing patterns. Each of the images is annotated with an aggregated saliency map representing the regions of highest importance based on the traces of several users, alongside the individual timed mouse trajectories from the users.

Figure 3.2: Example images overlaid with saliency maps taken from the SALICON dataset [Jiang *et al.* 2015].

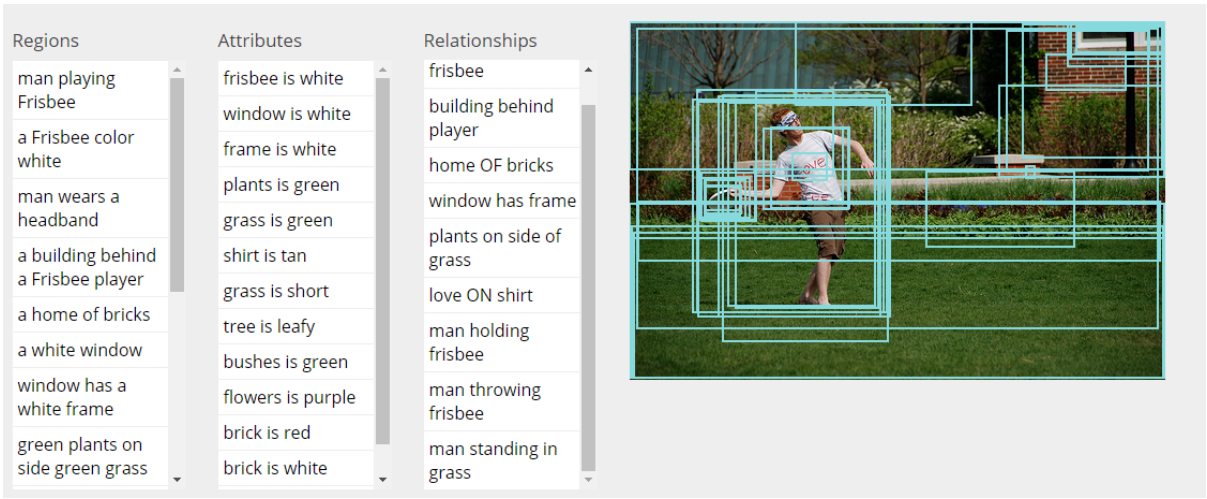


Figure 3.3: Example image alongside the many annotations taken from the Visual Genomes dataset explorer [Krishna *et al.* 2017].

Some example images annotated with saliency maps taken from the SALICON dataset explorer can be seen in Figure 3.2. Notably, while the individual trajectories are provided with timings, they are not synchronised or associated with any particular caption from the COCO annotations.

Visual Genomes

Visual Genomes [Krishna *et al.* 2017] is a dataset of approximately 108k images annotated with millions of region instances, descriptions, and relationships. Each image is segmented into various regions containing objects and prominent features. Regions can have several attributes recorded as well, such as colour, size, or other descriptive traits, as well as relationships to other objects in the image based on various predicates. For example, a man and a house could be regions in an image, with an attribute for the house being red, and relationship for the man being in front of the house. Additionally, the images are annotated with various possible question and answer pairs, alongside scene and region graphs. While the dataset itself does not contain captions, it makes it possible for models to directly learn visual groundings for keywords. An example image from the dataset explorer can be seen in Figure 3.3.

DIDEC

Relatively small compared to other datasets, DIDEC [van Miltenburg *et al.* 2018] is a Dutch dataset composed of 307 images that appear in the COCO, SALICON, and Visual Genomes datasets. Each image is annotated with a caption that was spoken by a user, as well as the gaze tracking data as the user was describing the image. The captions were transcribed and time-stamped semi-automatically, and synchronised to the timings of the gaze tracking data. This synchronisation means that models can now directly learn which regions of an image the phrases are associated to by utilising the gaze position at the time the phrase was uttered. It should be noted that the DIDEC captions are still relatively short, composed primarily of several word sentences.

Localized Narratives

The most relevant and recent dataset, Localized Narratives [Pont-Tuset *et al.* 2020], is composed of approximately 800k images taken from the MS COCO [Lin *et al.* 2014], Flickr30k [Plummer *et al.* 2015], ADE20k [Zhou *et al.* 2017], and OpenImages [Kuznetsova *et al.* 2020] datasets. Similarly to DIDEC [van Miltenburg *et al.* 2018], each image is annotated with a transcribed, spoken caption synchronised to a timed mouse trace which was recorded as the image was described by the annotator. Notably, the Localized Narratives captions are substantially longer than other existing datasets, with captions extending over 100 words in length. An example image from the dataset annotated with the mouse trace can be seen in Figure 3.4. More discussion on the associated work can be found in Section 3.4.

3.1.2 Transfer Learning

The ability for models to develop transferable knowledge has become increasingly of interest, particularly in the field of computer vision where many ensemble models rely on a visual backbone which can be fine-tuned on many related tasks. Recently, models utilising visual grounding have shown to be especially useful for this, likely due to their more structured learning of visual features. This would enable them to generalise better to a variety of tasks by extracting finer yet still important features, whereas previous models may only provide information about larger or more general features.

A powerful trait of recent models is their ability to learn specifically from unsupervised image and text pairs for transfer to downstream tasks. One such model, CLIP [Radford *et al.* 2021] is a general purpose system that learns to match images and associated text batches. The system has two variants, either using a CNN-based image encoder, or a more performant fully Transformer-based ViT system [Dosovitskiy *et al.* 2020]. In both cases the language decoder is a traditional Transformer architecture. The system is designed primarily for knowledge transfer to related tasks, and is largely successful, achieving similar performance to SOTA on image classification tasks even with zero-shot learning, showcasing the Transformer architectures strong ability to generalise. However the model under-performs in more specific tasks using data such as medical imagery or satellite data.

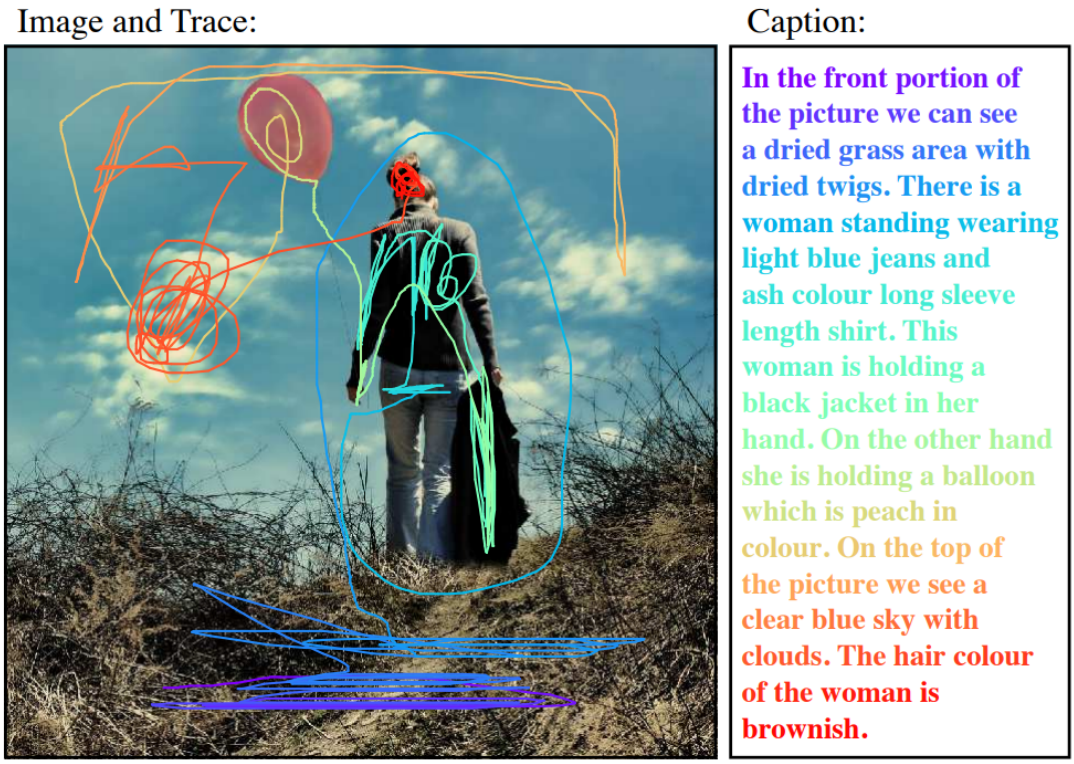


Figure 3.4: Example image annotated with the timed mouse trace taken from the Localized Narratives dataset explorer [Pont-Tuset *et al.* 2020].

As such, other models aim to transfer knowledge in more domain specific fields, such as ConVIRT [Zhang *et al.* 2020]. ConVIRT is a Transformer-based system aimed at downstream medical image tasks. The system is pre-trained on a large number of unsupervised paragraph-image pairs for medical images, with the intent of the model building its own understanding between the visual and linguistic features. The model can then be applied with minimal fine-tuning to various one-shot tasks, and achieves similar results as SOTA medical systems on image retrieval or classification tasks, with substantially lesser data requirements.

Other models take a supervised approach to knowledge transfer, such as VirTex [Desai and Johnson 2021]. The system is composed of a ResNet-based encoder and Transformer-based captioning decoder. The model is trained on the MS COCO [Lin *et al.* 2014] dataset using the image and caption pairs for the image captioning task. Once the system is trained, the image encoder can be detached and applied to downstream tasks, where it performs similarly to SOTA domain specific methods, such as in image segmentation and classification. Once again the system achieves this performance with minimal fine-tuning and data required.

3.2 Human Attention

Many deep learning approaches can take some inspiration from human-centric concepts, and as such the use of human or human-like attention may well suit these ap-

proaches. Human attention can be utilised in several ways, such as visual saliency maps, fixation points, timed attention traces, and others. In many fields the inclusion of human or human-like attention has proven to yield both quantitative and especially qualitative improvements over purely machine attention based systems.

3.2.1 General Applications

Human attention has received focus in various tasks based around visual reasoning and language. Sood *et al.* [2020] make use of human attention by recording saliency over written text sentences to determine important words. They use this data to develop a saliency model composed of both an LSTM and Transformer component which is aimed at predicting salient words in text, and when trained on question answering tasks their system achieves SOTA results. This saliency predictor can also be used in conjunction with other task specific question answering systems by integrating it via an attention layer, and they show this can assist in downstream tasks with minimal fine-tuning required.

Similarly, Qiao *et al.* [2018] introduce their so-called HAN module, designed to predict human-like attention when provided an image and related question. Their system utilises a series of GRUs to store information over the response sequences and update their attention predictions over the image at each time step. Their approach improves over a baseline system on the VQA task, and qualitatively their attention generations are shown to closely match genuine human data, and can be used to augment existing datasets with mimicked human attention.

Min and Corso [2021] also focus on the simulation of human-like attention data, however in this case for video data. Their model attempts to learn human-like attention based on real human gaze information over the course of a video, and in turn apply this mechanism in a larger model for actions classification in videos. They also attempt to account for timing and sensor inaccuracies in gaze measurements which can confound other models, and overall their approach improves on existing SOTA results for egocentric activity recognition in videos.

Also in the field of video activity systems, DirecFormer [Truong *et al.* 2022] is a novel Transformer-based system used to determine the order of a sequence of actions in video data. They make use of a spatial attention system augmented with human gaze data, alongside a similar augmented temporal attention mechanism, and their approach yields state of the art performance on action recognition tasks.

3.2.2 Image Captioning

Human and human-like attention in image captioning tasks is more widely utilised. The majority of approaches make use of static saliency maps and human-like attention mechanisms to better attend to important regions of the images, however newer approaches and datasets allow for temporal attention mechanisms.

One of the first human-attention enriched captioning systems was the work of Sugano and Bulling [2016]. Their approach makes use of gaze saliency maps generated for

images based on the SALICON dataset, dynamically combining fixation points for important regions with non-fixated eye movements. First the image features are extracted from the image into a fixed size visual feature lists using a VGG backbone. An attention LSTM then takes this data alongside the gaze map features in order to give a weighted context vector to the language decoder at each time step. This system was shown to improve over conventional bottom-up models that used only the image as input to determine the attended regions, especially for cluttered images where there was no clear central object.

However, relying on the saliency maps during inference introduced an additional dependency to the system. Chen and Zhao [2018] instead introduce an additional saliency prediction system, also trained on the SALICON dataset, in order to generate human-like saliency maps for the images. The saliency prediction is then incorporated into their attention mechanisms, which they refer to as "boosted attention". Their LSTM-based language model can then use this boosted attention over the region features from their visual CNN backbone in order to generate captions, which improves on both their own baseline system and SOTA at the time.

Several other methods follow this same approach, such as He *et al.* [2019] whose model utilises saliency map predictions in order to better attend to features extracted from an image, which again improves over their own baseline and achieves similar performance to SOTA. Additionally, they introduce their own new dataset containing human attention annotations based on eye-motion tracking. Sun *et al.* [2020] uses a dual branch attention mimicking system called OpenSalicon [Thomas 2016] to generate their saliency and fixation maps. These saliency and fixation maps are then used with the image features to generate the caption or answer visual questions. However, unlike previously mentioned methods, their attention system makes use of multiple saliency sub-modules in a similar vein to multi-head attention when attending to visual regions. This multi-module approach is intended to improve the diversity of regions attended to by the system, and does in fact show improved results on complex images with many objects.

Until this stage many of these systems are limited to using raw visual feature lists extracted directly by a CNN backbone. Alahmadi and Hahn [2022] make use of an R-CNN region-proposal system instead for their attention modifications. As with other works they include a gaze prediction module, however, here based on the integrated pointer network architecture [Vinyals *et al.* 2015a]. This module generates their saliency maps, which are passed alongside the region proposals to an LSTM-based decoder as with other approaches, and when used as a pre-trained encoder backbone in downstream tasks their system achieves performance similar to SOTA with minimal fine-tuning.

Instead of directly generating saliency maps, systems can also utilise human-like attention in other ways. Al-Malla *et al.* [2022] use a combination of an Xception CNN and YOLOv4 model to generate their visual encodings. The Xception network extracts the visual features from the image, and the YOLOv4 model generates object features which are weighted by visual importance based on prediction confidence and visual size, intended to mimic what humans find important in an image. This visual information is then passed to a GRU-based decoder to generate image captions. They show that

this approach improves over their own baseline, however they do not provide a clear comparison to existing SOTA.

These human-like systems can also apply to video data. Yu *et al.* [2017] make use of a saliency prediction module that acts on each frame of a video rather than a static image. Their system utilises the visual features and inter-frame motions and foveations to determine temporal saliency. This saliency data is then passed on to a two-layer GRU decoder to generate captions for the video, and their model achieves SOTA performance on both the captioning task and the saliency prediction task.

However all of these systems have so far used static saliency maps which are not aligned to their captions. Takmaz *et al.* [2020] present the first model that utilises temporal gaze data rather than just static saliency maps, although their system does additionally utilise aggregated saliency maps. Their approach makes use of their own DIDEC dataset [van Miltenburg *et al.* 2018], containing gaze data temporally aligned to utterances in the caption they are trying to generate. They utilise a two-layer LSTM-based decoder as with many other approaches, alongside an additional gaze LSTM which accumulates the sequential gaze maps over time, with the addition of temporal gaze showing improvements over existing methods. The Localized Narratives approach [Pont-Tuset *et al.* 2020] and MITR system [Meng *et al.* 2021] also make use of temporally aligned attention traces, however with substantially longer captions and using the Transformer architecture. These approaches will be discussed more thoroughly in Section 3.4 and Section 3.5 respectively.

3.3 Cross-Modal Encoding Systems

The vast majority of the human attention enriched captioning systems mentioned thus far are LSTM- or GRU-based, however Transformers have also shown capability of handling multi-modal data. Importantly, Transformers have been utilised effectively with encoding processes that combine multiple modalities of data to yield improved performance.

LXMERT [Tan and Bansal 2019] is a dual-branch Transformed-based system used to generate visual encodings of images alongside linguistic and meta representations of captions or questions. During the encoding phase, the system performs cross-attention between the language and vision branches to yield a more holistic encoding of each one. This cross-encoded data is then used to generate visual features from the first branch, and a caption output from the second, alongside a task-specific cross-modal output such as a response to a visual question or image match confidence. Their system is pre-trained on masked sentence-image pairs and then fine-tuned for various down-stream tasks, and achieves SOTA performance on VQA challenges.

While not strictly cross-modal, Pan *et al.* [2020] introduce their X-LAN networks, LSTM- or Transformer-based systems that utilise so-called X-Linear attention modules. Their X-Linear attention introduces additional interactions between the query embedding and value embedding, as well as a so-called "squeeze excitation" block which modifies the final weighted sum of the attention. This allows the attention mechanism to model

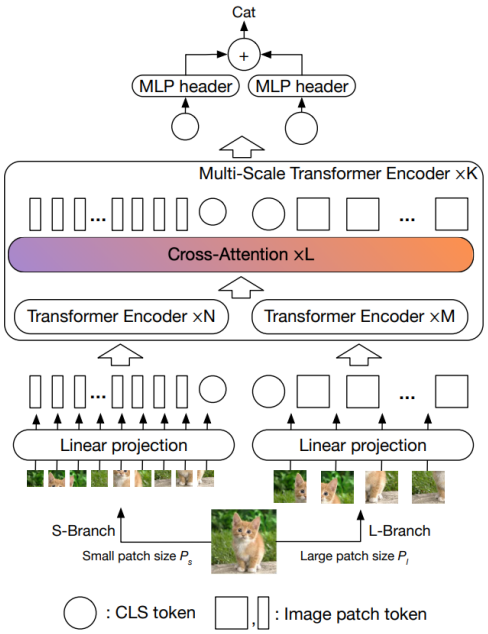


Figure 3.5: Visualisation of the CrossViT architecture taken from their paper [Chen *et al.* 2021].

higher order interactions between the different modalities in the system without explicit per-modality branches. This addition yielded SOTA results on the COCO captioning task.

Also not strictly a cross-modal encoding system, the CrossViT [Chen *et al.* 2021] network does perform some sense of cross-encoding in its use of multi-resolution image data, and our work here draws strong inspiration from it. Their cross encoding system shares information between two branches of different image resolutions to yield more holistic encodings and improve on existing image classification systems. The network is covered with more detail in section 2.4.3, and a visualisation of their approach can be seen in Figure 3.5.

3.4 Localized Narratives

The Localized Narratives [Pont-Tuset *et al.* 2020] approach is, to the best of our knowledge, the first to attempt long-text descriptions of images. Their recently released dataset is discussed separately in section 3.1.1. Their approach at the captioning task utilises a straightforward two-layer Transformer architecture with no modifications from the original configuration [Vaswani *et al.* 2017]. Their implementation is primarily meant as a baseline for future development, and is not a main focus of their research.

In their implementation, they first pre-process the mouse traces from the dataset by cutting them according to their timestamps into segments of 0.4s each. These segments are converted into bounding boxes by taking the maximum and minimum co-ordinates of each segment, and the relative area of the box is appended before the values are

embedded to the model dimensions. They also pass the image through the Bottom-Up encoder [Anderson *et al.* 2018] to yield bounded image regions, and embed the region features as well as the region bounding boxes to the size of the model. The region features and their respective boxes are combined, which is intended to provide positional encoding to the encoder. These visual and trace embeddings are then concatenated and provided to the Transformer decoder as is.

The captions are then tokenised and each token embedded to the size of the model, with positional encoding added using the sinusoidal approach discussed in Section 2.3.2. While not explicitly discussed, it appears that the tokenisation of the captions retains punctuation and uncommon words based on the provided samples. The decoder then utilises cross-attention over the multi-modal encoded sequence, and masked self-attention over the input caption, to generate the output captions.

To investigate the improvement gained from the trace box sequence, they perform a series of additional experiments. First, they entirely exclude the trace boxes from the system, and find that the model performs poorly as expected, lacking any form of guidance on which features to describe or how long the caption should be. They then include the trace boxes – however with their areas all set to cover the entire image – essentially guiding the model with only the desired length of the caption and discarding spatial information. This improves the model's performance over having no trace at all, but still does not provide information on which features to describe. The final approach of providing the true trace boxes performs substantially better than either of these approaches, both in terms of quantitative metrics and qualitative results, indicating the usefulness of the trace modality.

3.5 MITR

The MITR approach [Meng *et al.* 2021] expanded upon the Localized Narratives model by segmenting the Transformer system into three branches – one for each of the vision, trace, and caption modalities. Their work also introduced two new sub-tasks, namely attention trace generation, and joint caption and trace generation. A visualisation of the architecture taken from the original paper can be seen in Figure 3.6.

In their design, the vision encoder consumes the region proposals embedded in the same format as the Localized Narratives [Pont-Tuset *et al.* 2020] implementation and produces the encoded visual sequence. The trace and caption branches can then each swap between encoding and decoding configurations depending on the task at hand, with the caption input embeddings once again produced in the same manner as in Localized Narratives. The trace embeddings are produced similarly, however trace segments are split by word according to the time synchronisation, rather than into uniform 0.4s intervals. This is because different narrators tend to speak at different rates, and will often change pace throughout the caption, making time intervals and unreliable estimate of caption length. Thus, splitting by word rather than time interval allows a more accurate guidance to the model of what the output caption length should be.

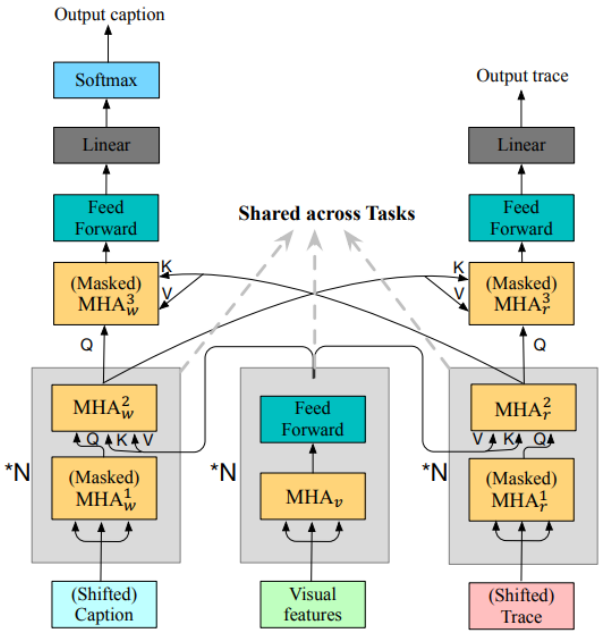


Figure 3.6: Visualisation of the MITR architecture taken from their paper [Meng *et al.* 2021].

As for the other modalities, when a branch is acting as an encoder it is provided the with the relevant modality embedding and performs unmasked self-attention, and cross-attention over the visual encodings. The output encodings are then provided to the branch of the remaining modality. When acting as a decoder, the self-attention in the branch is right-masked to avoid information leakage, and a final cross-attention is performed with the decoded sequence. This cross-attention is over the encodings generated by the other modality mentioned previously. There is also the case where both branches act as decoders in order to generate captions and trace boxes simultaneously, in which case both branches receive the encodings from the other modality and generate their own decoding.

The advantage of this dual-functionality approach is two-fold. First, and most importantly, the model can be trained a single time while being able to generate either single modality. Additionally, the model need only be re-trained – without architectural modification – to generate predictions for both modalities jointly using only the visual features. Secondly, generating predictions of the both modalities allows the model to utilise cycle-consistency loss and improve output quality. By using the generated modality outputs as input to the system in a second pass, a loss can be applied to first pass outputs that do not retain enough necessary information to generate predictions in the second pass.

While this cycle consistency loss introduces additional overhead to the training process – as there are now two passes required through the entire model – the authors do show that it improves the model's performance during inference. On top of this, the structure of their architecture alone already outperforms the Localized Narratives approach [Pont-Tuset *et al.* 2020] without cycle consistency loss. However, the authors

themselves note that the different forms of trace embedding used to that of the Localized Narratives work – splitting by word instead of by time – make the comparison incomplete, as it affects the ground truth input provided to the model for generation.

3.6 Summary

In this chapter we introduced the most relevant research to the work we introduce in this report. We discuss the topic of grounding language into vision through various methods, and the datasets that exist to enable this alongside some approaches in the field. This leads to the topic of human attention and its applications both in broader vision and NLP contexts, and specifically in image captioning where it shows improvements over plain image input models. We then briefly discuss cross-modal encoding systems and how they provide more holistic encodings, and cover in detail the very relevant Localized Narratives [Pont-Tuset *et al.* 2020] and MITR [Meng *et al.* 2021] works which inspire much of the work in this research.

Chapter 4

Research Methodology

The focus of this research is to expand the existing work on long-text description of images, in this instance through the use of localised attention. We investigate a series of LSTM and Transformer-based approaches to determine important contributions of different methods toward the final results, and compare our implementations to existing methods in the field.

For this chapter, we first introduce our hypotheses and related research questions that we seek to address, followed by the motivation and applications for this research. We then discuss the practical aspects of our approach, namely the data we make use of, and the metrics we will be using to evaluate our results. Finally, we discuss the common preparation steps for the data, with more detail of architecture implementations and their specific data preparation steps in the following chapters.

We note that the detailed discussions of our implementations, architectures, and motivation of our decisions are presented in Chapter 5 for our LSTM- and Transformer-based approaches. This chapter serves instead as a holistic overview of the research.

4.1 Research Questions and Hypothesis

We present the series of research questions this work seeks to answer, as well as our initial hypotheses relevant to each one and how we intend to address the questions in practice. Our questions and hypotheses are broken down between our LSTM approaches and Transformer approaches, however it should be noted that our initial findings from our LSTM approaches motivate our choice of questions and hypotheses for the Transformer section.

4.1.1 LSTMs

To the best of our knowledge no literature currently exists on the task of long-text image description using LSTM based approaches, and as such we expect LSTM approaches to under-perform as compared to Transformer approaches. Nevertheless we aim to see if LSTM-based approaches are at all viable, or if they fail to capture the complexity of

the solution space. The detailed discussion of the LSTM implementations and relevant decisions can be found in Section 5.1.

1. Are LSTM-based systems powerful enough to handle descriptions of the desired length?

   Hypothesis: *While they suffer from degradation over long sequences, an LSTM-based system should be able to generate captions hundreds of words long with at least some success.*

   Approach: *Develop an LSTM encoder-decoder and train it to embed and reconstruct captions in the dataset, and use common metrics to judge reconstruction accuracy.*

2. Can an unguided CNN-LSTM system generate reasonable captions for an image?

   Hypothesis: *Generating a description of the desired length from only the image features is not viable due to the lack of guidance in the attended regions and their ordering.*

   Approach: *Train a ResNet+LSTM captioning system that utilises only the raw image without any trace data and evaluate it using common metrics.*

3. Does providing a saliency/attention heatmap per time step improve the performance of the captioning model?

   Hypothesis: *Converting the traces for each caption into a series of heatmaps should help guide the model in the regions it describes and their order, and lead to an improvement over the unguided model.*

   Approach: *Modify the previous model to now consume the image and heatmap per time step, and accumulate this information into a vision encoder LSTM.*

4. Do Faster R-CNN regions with attention boxes perform better than ResNet features with heatmaps?

   Hypothesis: *Faster R-CNN region proposals have become prevalent over fixed-size CNN feature maps in many vision tasks, especially captioning, and as such we expect that using them will provide richer information to our model and improve its performance.*

   Approach: *Modify the previous model to now consume a series of Bottom-Up [Anderson et al. 2018] region proposals and trace boxes per time step, and accumulate this into the vision encoder LSTM.*

5. Does attention from the language LSTM over the vision LSTM outputs improve performance?

   Hypothesis: *Integrating attention has been shown to largely improve performance in many fields, and we expect there should be no difference here.*

   Approach: *Modify the previous model to perform attention over the visual features during the caption generation.*

4.1.2 Transformers

There already exists literature utilising Transformer-based approaches in the long-text description task, and so we neglect questions focused only on the viability of such systems and instead focus on modifications to these systems. The detailed discussion of the Transformer implementations and relevant decisions can be found in Section 5.2.

1. How does an un-modified Transformer architecture perform on caption generation when utilising region proposals and trace boxes?

   Hypothesis: *The original Transformer approach [Vaswani et al. 2017] has shown to be flexible to many different tasks with minimal adjustments, and as such we expect it to perform suitably here.*

   Approach: *Combine our region proposals and trace boxes into a single sequence and train a basic Transformer architecture on this data.*

2. Does splitting the Transformer into two branches – one per modality – improve over the straightforward combined approach?

   Hypothesis: *While having a single encoder and decoder handle both modalities allows for cross-modal attention, it may overload the weights, and as such separating the system into two branches should allow the branches to specialise and improve performance.*

   Approach: *Modify the Transformer architecture to have two parallel encoder branches and two consecutive decoder branches – one per modality.*

3. Can cross-modal encoding through the use of context tokens as in Chen *et al.* [2021] improve the encoding process?

   Hypothesis: *Splitting the architecture into two encoder branches improves specialisation of the weights, but means cross-modal attention cannot happen. As such we expect re-introducing cross-modal information sharing to improve the encoding and as such model performance.*

   Approach: *Modify the two-branch system to utilise a cross-modal context token system during the encoding process.*

4. Does alternating modality decoding outperform one-at-a-time decoding in multi-layer decoders?

   Hypothesis: *Alternating modalities at each decoding layer should allow the model to better retain information from each modality, as opposed to going through all the decoder layers of each modality in a single shot.*

   Approach: *Modify the cross-encoding system to alternate modalities at each layer during the decoding process.*

5. Can newer, fully Transformer patch-based vision encoders outperform the Bottom Up proposals for image description?

Hypothesis: *Some modern vision encoding systems utilise Transformer encoders over raw image patches instead of R-CNN systems, and as such a patch-based model may improve over Bottom Up proposals.*

Approach: *Modify the cross-encoding system to utilise the Swin [Liu et al. 2021] image encodings instead of Bottom Up region proposals.*

4.2 Motivation

Image captioning in general has a variety of applications, such as accessibility for visually impaired users, image searching by feature description, image sorting based on common descriptors, and others. Additionally, image captioning systems will develop visual-linguistic understanding that can be used in down-stream tasks such as visual question answering, or more generalised challenges that involve visual understanding. Rich, long-text descriptions compound on these advantages firstly by improving on the level of detail in their description of the image, and secondly by developing a more robust connection between vision and language.

Thus this work aims to understand the viability of new image description approaches, and expand upon the existing approaches in modern literature. This is in the hopes of providing an improved, SOTA system in the field of long-text image description which benefits the above applications.

4.3 Datasets

We make use of the MS COCO [Lin *et al.* 2014] sub-set of the Localized Narratives [Pont-Tuset *et al.* 2020] dataset for our model training and evaluations, as in the original paper and the MITR approach [Meng *et al.* 2021]. The splits provided by Localized Narratives utilise the same split as the regular MS COCO , which includes only a training set and validation set. As such, we follow the widely used Karpathy split [Karpathy and Fei-Fei 2015] which segments the training and validation sets into mutually exclusive training, validation, and test splits of $110,000$, $5,000$, and $5,000$ images respectively.

4.3.1 Limitations.

While the Localized Narratives dataset is fundamental to this task, it carries with it a variety of challenges due to the nature of subjective image captioning.

The provided word-trace sequences are not always perfectly aligned when qualitatively analysing samples from the dataset. In many instances the delay between the narrator describing the image and moving the mouse results in words being grounded to regions of the image where they are not suitable. Additionally, a substantial amount of captions begin with phrases such as “In this image, I can see..” or “This image is...”, which are neither visually groundable nor descriptive of the image itself.

Errors in the voice-to-text transcriptions where certain words are mistaken for similar sounding ones that do not fit the context of the image also occur, although it is difficult to determine how frequently this is the case. Additionally, narrators will sometimes use vague or generalised wording when describing objects within an image, e.g. “instrument“ instead of “drum”, or incorrect grammar in their sentence structure.

This compounds with the natural difficulty of objectively scoring a generated description. Even though captions are guided by traces, there are still a large variety of suitable captions to describe the path through the image, and sometimes generated captions will be scored poorly even in instances where the generated caption can appear qualitatively better than the provided ground truth.

4.4 Metrics

Various metrics exist for evaluation of machine translation or image captioning models. These metrics are primarily based on n-gram matching, or more recently, scene graph matching, and each have various strengths and weaknesses which make them subject to “gameability”. Thus, we utilise a series of the most common evaluation metrics to provide a holistic view of the performance of our models, which we discuss below.

4.4.1 BLEU

The Bi-Lingual Evaluation Understudy (BLEU) metric [Papineni *et al.* 2002] was initially developed to evaluate machine translation systems. It rates target or generated documents on a scale of 0 to 1, with 0 representing no correlation between the target and reference documents, and 1 being perfect correlation. The system relies on the concept of n-grams and precision, where an n-gram is a series of n consecutive tokens in a document, and precision the rate at which n-grams in the target appear in the references, with values of n ranging from 1 to 4 being most common. The system uses clipped precision rather than plain precision to penalise excessive repetition. In clipped precision, n-grams in the target only contribute towards the precision score up to the maximum number of times they appear in any reference document.

Target documents are evaluated by first calculating the clipped precision against the reference documents for each of the desired n-gram sizes. A brevity score from 0 to 1 is also calculated that exponentially penalises target documents shorter than their reference documents. Each BLEU-N score is then calculated by taking the geometric average precision of each n from 1 to N and multiplying it by this brevity score.

We can represent the whole evaluation process as:

$$\text{BLEU-N} = \min(0, 1 - \frac{r}{c}) + \sum_{n=1}^{N} w_n \log(p_n) \tag{4.1}$$

Where r is the reference length, c is the target length, p_n is the clipped precision, and w_n is the weighting of each n-gram (summing to 1), most commonly with $w_n = \frac{1}{n}$.

While BLEU has been shown to match human ratings of translations relatively well, it has some shortfalls. The metric has no consideration of synonyms, and words like "person" and "human" will be considered entirely different even if contextually synonymous. Above this, the metric does not utilise word stemming and even the same word in a different form, for example "rain" and "raining", will be considered different. Additionally, there is no sense of word importance in documents, with common filler words such as "the" or "a" being ranked equally with conceptually important words in the sentence.

4.4.2 ROUGE

The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) package [Lin 2004] is a series of metrics based on machine translation, with varying calculations and use-cases. The primary metric in the ROUGE package for our interest is ROUGE_L, which is based on the longest common sub-sequence (LCS) between the target and reference documents. As with BLEU, ROUGE_L ranks target and reference documents between 0 and 1, with 0 being no correlation and 1 being perfect correlation. First the recall is calculated as the LCS length over the reference sentence length, followed by the precision as the LCS length over the target length. The F score is then calculated and returned as the final metric value, which we can formalise as:

$$R_{LCS} = \frac{\text{LCS}(t, r)}{l_r} \tag{4.2}$$

$$P_{LCS} = \frac{\text{LCS}(t, r)}{l_t} \tag{4.3}$$

$$\text{ROUGE}_L = (1 + \beta^2)\frac{R_{LCS}P_{LCS}}{R_{LCS} + \beta^2 P_{LCS}} \tag{4.4}$$

Where t is the target document, r the reference, l_t and l_r their respective lengths, and β the recall importance weighting, commonly $\beta = P_{LCS}/R_{LCS}$.

ROUGE_L shares many of the downfalls of BLEU in its lack of synonym handling, word stemming, and handling of relative importance of words.

4.4.3 METEOR

The Metric for Evaluation of Translation with Explicit Ordering (METEOR) metric [Banerjee and Lavie 2005] was developed to make up for some shortfalls of the BLEU metric. As with BLEU and ROUGE, METEOR rates sentences on the same scale of 0 to 1. However, unlike the previous metrics, METEOR utilises word stemming and synonym replacement during evaluation, and is targeted at evaluating single documents rather than full corpuses (although it can be used for both).

First, the target and reference documents are aligned by pairing any matching unigrams, accounting for stemming and synonyms. The recall and precision are then calculated using the alignment, and the F-score is calculated prioritising the recall value. Lastly a penalty is calculated by dividing the target into the smallest possible number

of so-called chunks, where each chunk is a matching n-gram, and penalising sentences with a higher number of chunks – representing the fragmentation of unigram matches. The process can be formalised as:

$$R = \frac{m}{l_r} \tag{4.5}$$

$$P = \frac{m}{l_t} \tag{4.6}$$

$$F = \frac{10PR}{R + 9P} \tag{4.7}$$

$$p = 0.5 \left(\frac{c}{m} \right)^3 \tag{4.8}$$

$$\text{METEOR} = F(1 - p) \tag{4.9}$$

where m is the number of matching unigrams in the target and reference, l_t and l_r their respective lengths, and c the number of chunks. The only remaining shortfall of METEOR also found in BLEU and ROUGE is the lack of word importance in document ratings, however METEOR also introduces another possible flaw in that synonyms can often be subjective. While some correct cases can still be missed, in others unigrams may be too leniently matched where synonyms are invalid for the context. Additionally, allowing word stemming can be too lenient on incorrect grammar and tense usage.

4.4.4 CIDEr

The Consensus-based Image Description Evaluation (CIDEr) metric [Vedantam *et al.* 2015] utilises similar concepts as BLEU and METEOR and was designed for image captioning evaluation. It uses n-gram matching with n commonly ranging from 1 to 4 with word stemming – although notably does not use synonym matching – and also introduces a Term Frequency Inverse Document Frequency (TF-IDF) weighting of n-grams, and rates documents on a scale of 0 to 10. The TF-IDF weights n-grams higher according to how frequently they appear in the reference sentences, and lower according to how frequently they appear in the entire reference corpus. The process can be formalised as:

$$g_k(r_{ij}) = \frac{h_k(r_{ij})}{\sum_{w_t \in \Omega} h_t(r_{ij})} \log \left(\frac{|I|}{\sum_{I_p \in I} \min(1, \sum_q h_k(r_{pq}))} \right) \tag{4.10}$$

where r_{ij} is the j-th reference document of image i, t_i the target document, h_k the number of times n-gram k appears in a document, Ω the vocabulary of the corpus, I the set of all images, and g_k the TF-IDF score of n-gram k. The left fraction represents the TF – or higher weighting of n-grams appearing in the reference sentence – and the right fraction represents the IDF – or lower weighting of terms appearing frequently in the corpus.

As for the CIDEr score itself:

$$\text{CIDEr}_n(t_i, R_i) = \frac{1}{m} \sum_j^m \frac{g^n(t_i) \cdot g^n(r_{ij})}{\left\| g^n(t_i) \right\| \left\| g^n(r_i j) \right\|} \tag{4.11}$$

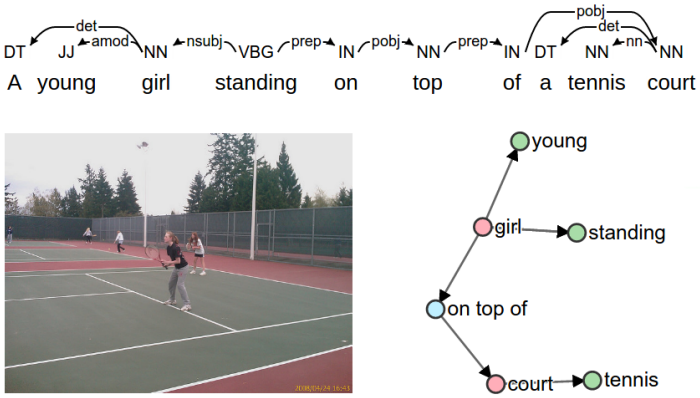


Figure 4.1: Example image, caption, and associated scene graph generated using SPICE taken from their original paper [Anderson *et al.* 2016].

where R_i represents the set of reference documents for image i, m the number of references for i, and g^n the set of g_k values for all n-grams in the document. The final, combined score is calculated as:

$$\text{CIDEr}(t_i, R_i) = \sum_{n=1}^{N} w_n \text{CIDEr}_n(t_i, R_i) \tag{4.12}$$

where w_n is the weighting (summing to 1) of each n-gram score, with the most common N and w_n values being 4 and $\frac{1}{N}$, as in the original paper. The advantage of CIDEr over the previous metrics is that it introduces varying importance of words through the TF-IDF weighting. Words that appear more frequently in the corpus are more likely to be structural or vaguely descriptive and are generally applicable to many images. Conversely, words appearing more frequently in the references for a specific image – but not the corpus overall – are more likely to refer to visual aspects particular to the image. Thus, the TD-IDF weighting helps provide an approximation of how well our model captures unique visual aspects of images.

4.4.5 SPICE

The Semantic Propositional Image Caption Evaluation (SPICE) metric [Anderson *et al.* 2016] differs from the previously discussed metrics in that it uses a scene graph rather than n-gram matching to evaluate targets. First, the reference captions are provided to a scene parser in order to generate a scene graph similar to that of the Visual Genomes dataset [Krishna *et al.* 2016]. The graph for each image contains a series of objects alongside their attributes and relations to each other extracted from the reference captions, and utilises stemming for each word. An example scene graph can be seen in Figure 4.1.

The target caption is then parsed as well and matched against the reference scene graph based on the objects first followed by their related attributes and relationships. Their approach also utilises synonym matching similar to METEOR [Banerjee and Lavie

2005]. The metric better correlates with qualitative human judgement than others, and is more flexible in how it evaluates divergent, but high-quality targets that correctly encapsulate the image, however it does so at the cost of structure. The graph representation is unable to effectively capture grammatical structure, and additionally only scores object and property matching, meaning systems can score well by generating captions that include objects with correct attributes and properties while neglecting sentence structure and fluency.

4.5 Data Preparation

While our data preparation varied depending on the architecture utilised at each stage, some portions are shared between some LSTM and Transformer approaches, namely the caption and trace box pre-processing, and image region extraction. Methods specific to any architecture are discussed in their relevant chapters below.

4.5.1 Caption Pre-processing

The pre-processing approach for the majority of our models involves converting the caption to an embedded sequence in the latent space of the model. First the caption is lower-cased and provided to a tokeniser, in our case the *word_tokenize* method from the NLTK library [Bird *et al.* 2009]. This splits the caption into a sequence of separate language tokens for each word and punctuation symbol, accounting for hyphenated words and repeated punctuation using regular expressions and a pre-generated word bank.

This process is first run over all reference captions, and each unique language token is added to a vocabulary mapping from language tokens to unique indices representing them. Additionally, three meta-symbols are included in the vocabulary, namely the *[STR]*, *[EOS]*, and *[NULL]* tokens, representing the start of the string, end of the string, and null padding respectively. When each sequence is tokenised, the start and end tokens are added to the beginning and end of the caption, and the null token is added where needed for captions shorter in length than that of the batch they form. The final vocabulary mapping is then used to convert each language token sequence into the encoded index token sequence to be provided to the model. An example of a caption batch being pre-processed can be seen in Figure 4.2.

4.5.2 Trace Box Pre-processing

The primary method for converting the trace data into a format consumable by our models was by using bounding boxes over segments of the traces, similar to that of Meng *et al.* [2021]. The time stamped mouse traces provided by Localized Narratives [Pont-Tuset *et al.* 2020] are time synchronised to their captions, which themselves are broken into variable length time stamped utterances, where each utterance has one or more words in it with a starting and ending time. As such we can split the trace into one segment per word using the time stamps.

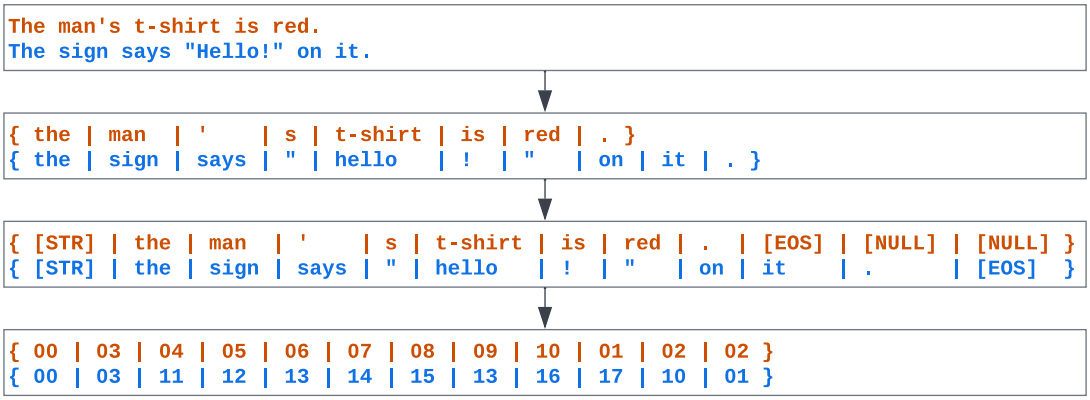


Figure 4.2: An example batch of captions going through the pre-processing flow.

Figure 4.3: Example of the time synchronised caption and trace, as well as the generated trace boxes. Colour change indicates the flow of time.

First we tokenise each utterance using the same *word_tokenize* as above and split the time-span of the utterance evenly into as many segments as tokens. Next we extract the segment of the trace that fits within the sub-span and calculate the bounding box by taking the maximum and minimum normalised x- and y-coordinates of the segment points. We also append the area of the normalised coordinates to the box as in both the original method and the MITR implementation [Pont-Tuset *et al.* 2020; Meng *et al.* 2021], such that each bounding box becomes $(x_{min}, y_{min}, x_{max}, y_{max}, area)$. We also experimented with re-using the same time span for each uttered token, and using only a single trace per utterance instead of per word, however there was no measurably significant impact in our final results.

4.5.3 Image Region Extraction

For our image region extraction we make use of the pre-generated regions provided in the GitHub repository of the Bottom-Up Faster R-CNN system [Anderson *et al.* 2018]. Each image has 36 associated regions provided, composed of their latent features and

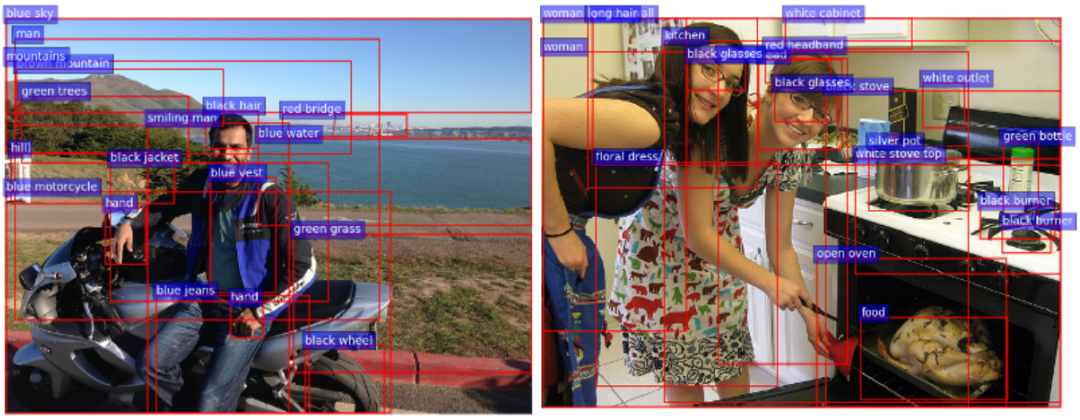


Figure 4.4: Example images annotated with their region proposals and classifications taken from the Bottom-Up attention demo [Anderson *et al.* 2018].

bounding box, which are intended to be the most visually important aspects of the image. Each of the regions have a 2048 dimensional feature representation which we refer to as the region features, as well as a bounding box which we normalise to the image dimensions and refer to as the region boxes. We also attempted producing our own region proposals using the pre-trained weights provided by the authors [Anderson *et al.* 2018], however noticed no significant impact on our final results. Some example region proposals and classifications can be seen in Figure 4.4.

4.6 Summary

In this chapter we have presented the overarching concepts primary to our research. We discuss the questions and hypotheses we seek to address through our experiments, as well as the motivation for carrying out this work. An overview of the Localized Narratives data we utilise for our training and evaluation is provided, alongside the various widely used metrics we report on for our models' quantitative performance. A discussion on each metric, its calculation process, and the strengths and weakness are included. Finally, the data preparation process for each of the caption, trace, and image region modalities is outlined and motivated. The implementation discussions for the LSTM-based and Transformer-based models are provided in the upcoming Chapter 5.

Chapter 5

Experimental Approaches

This chapter now introduces the fine implementation details, architectural decisions, and model-specific data processing. The more traditional LSTM-based approaches are discussed first, followed by the more advanced Transformer-based systems. Results from all of these approaches can be found in Chapter 6.

5.1 LSTMs

Our initial approaches to the long-text description task were based on LSTM architectures. Given the limitations of LSTMs such as information loss over long sequences and extensive time and hardware requirements for training, we expected that LSTM-based approaches would struggle with such long descriptions. Nevertheless we investigated the viability of such models in various stages of complexity.

We discuss in detail each of the architectures that we developed, and the progressively more complex mechanisms we introduced hoping to achieve better performance on our metric suite. The architectures begin with a simple one-step system utilising a ResNet visual backbone, which we then expand to a temporal system utilising a visual accumulator over multiple time steps. We then experiment with the R-CNN module in place of a plain ResNet backbone, and progressively introduce attention at several points of the model. Additionally, for each architecture we discuss any data preparation steps specific to the approach, such as heatmap generation and standard image encoding.

5.1.1 Language Encoder-Decoder

The first architecture is an auto-regressive encoder-decoder system that simply encodes the provided caption into a latent embedding, and then reconstructs the original caption in the decoder. This architecture acts as an ideal baseline for further models – allowing us to determine how effectively an LSTM based system can generate captions of the desired length. We stress that this model is not a captioning system and has no visual components, it simply serves as an indicator of how well an LSTM-based system can retain linguistic information its latent embedding space when provided with per-

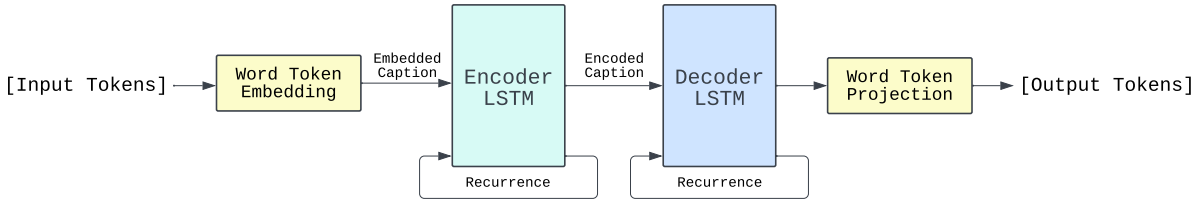


Figure 5.1: Visualisation of the Language Encoder-Decoder baseline model.

fect input. The architecture is composed of two identical LSTMs – one acting as the encoder, and one as the decoder.

First the caption is pre-processed and tokenised as described in Section 4.5. The tokenised caption is then embedded into the latent space of the encoder LSTM using a learned embedding, which acts as a lookup table for each token. The encoder LSTM then consumes the embedded sequence one token at a time, in order to build up the hidden memory tensor. The memory state and cell state start initially as all zeros, and once the sequence is exhausted the memory tensor should contain a representation of the entire caption.

The memory tensor and cell state are then provided to the decoder LSTM. The decoder iteratively generates output tensors which are projected back to the vocabulary space and soft-maxed to provide logits representing each possible token. This iteration continues until the maximum output caption length is reached, or until the *[EOS]* token is generated. During inference or when training without teacher forcing, the generated logits are used to find the most likely token, which is fed back into the decoder for the next time step.

The output sequence is then converted into a readable caption through the vocabulary token mapping. The reconstructed output caption should then match the input caption as closely as possible, and the system can be trained using cross-entropy loss between the ground-truth tokens and the output token logits. A visualisation of the architecture can be seen in Figure 5.1.

5.1.2 One-Step ResNet Captioner

The first actual image captioning architecture is a naive approach where only the visual features are used to generate the caption, without any dependence on the attention trace. Naturally this approach is not expected to be highly effective due to the complexity of the captions, and the limited information about the caption structure that can be discerned from the image. Nevertheless, it provides a very straightforward baseline for our captioning LSTMs and later approaches. The model is composed of a pre-trained ResNet visual encoder backbone, and an untrained LSTM-based decoder.

We utilise a pre-trained ResNet as it allows for transfer learning from related visual tasks trained on larger image sets, however the ResNet is also fine-tuned during training. Because the system requires the raw visual features for the image, the ResNet

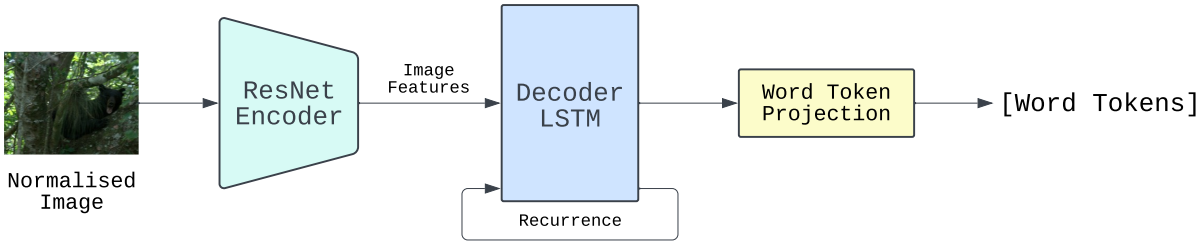


Figure 5.2: Visualisation of the One-Step ResNet captioning model.

classification layer is removed and the encodings are taken from the second last layer instead.

The encoding process is as follows. First the image is colour-space normalised by removing the mean and dividing by the variance of the dataset, resized to 224×224 pixels, and fed into the ResNet encoder. The ResNet encoding, or visual embedding, is intended to mimic the hidden memory that would have been generated by an encoder LSTM on the ground-truth caption – such as in the Language Encoder-Decoder model. As such, the visual embedding passes through a linear projection layer to normalise the dimensions, and is provided to the decoder LSTM as its starting memory tensor.

For the decoding process, the cell state is initialised with zeros, however other initialisation methods could be explored. The decoder then iteratively generates the output tokens in the same manner as the Language Encoder-Decoder, first generating output tensors, and converting them to logits. Cross-entropy loss is also utilised in the same manner as the Language Encoder-Decoder, as is the case for the remainder of our models. A visualisation of the architecture can be seen in Figure 5.2.

5.1.3 Temporal ResNet Captioner

In order to integrate the temporal aspect of the attention traces into our approach we need to modify our encoder to be sequential rather than one-step. As such, we introduce another LSTM which will act as a visual accumulator. This accumulator builds up a visual memory by consuming the information from the visual backbone combined with the trace data at each time step. This is conceptually similar to how the encoder in the Language Encoder-Decoder accumulates information about the caption by iteratively consuming each input token.

In order to combine the visual data with the trace data, we make use of heatmaps which can be integrated with the image. To generate the heatmaps, we start with 224×224 zero-grids, and utilise the attention traces associated with the image to fill them. First the traces are split into segments in the same manner as the trace preprocessing discussed in Section 4.5, however they are not converted into trace boxes. Instead, for each point in the segment we increase the cell values of the heatmap using a 2D Gaussian distribution centred on the mouse position, with a variance of 30. Once all the points have been added, the values in the heatmap are divided by the maximum cell value to normalise them between 0 and 1. This process is repeated for each segment in the trace, yielding a heatmap for each time step. Examples can be seen in Figure 5.3.

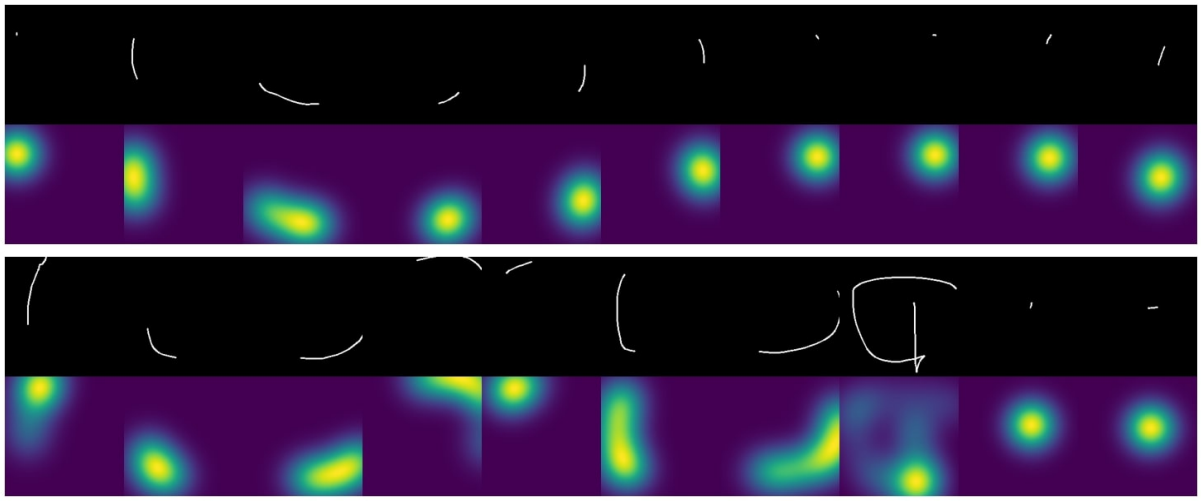


Figure 5.3: Visualisation of traces (top) and their associated heatmaps (bottom) for each time step in the sequence. Only the first 10 time steps are shown.

We experiment with three modifications to the existing One-Step implementation in order to integrate these heatmaps. The first modification focuses on augmenting the heatmaps, and the latter two modifications are variants of the ResNet encoder. The heatmap modification involves utilising the state of the newly added visual accumulator to augment the attention heatmaps before they are provided to either of the two ResNet variants. This is done by reshaping the memory tensor into a 28×28 grid, and utilising progressively larger 2D convolutional upsampling layers up to the same dimensions as the generated heatmaps. The upsampled memory representation is then merged with the heatmap using a 1×1 convolution down to a single layer, which is then passed on to the next module.

As for the ResNet modifications, the first variant involves element-wise multiplying the raw image with the attention heatmap generated for each time step, before providing it to the ResNet module. This allows us to directly control the information provided to the model by outright removing unattended visual data. The second ResNet variant instead altered the number of input channels in the ResNet backbone from three to four, with the final layer now being the attention heatmap which was stacked onto the image. This softer approach allowed the ResNet backbone itself to integrate the attention via the learned weights. However, this approach also meant the pre-trained ResNet weights could no longer be used. This would now require training the ResNet module from scratch, and means the model loses the knowledge transfer benefits. As such, we utilise the first variant for the remainder of our work.

In summary, for each time step of the sequence, the associated trace segment is extracted and converted to a trace heatmap. This trace heatmap is then augmented using the aforementioned mechanism, based on the hidden memory state of the accumulator. The raw image is then multiplied by the augmented heatmap to mimic attention, and is passed through the ResNet to generate visual features. These visual features are then provided to the visual accumulator LSTM, and this process repeats for each trace segment. Once the sequence is exhausted, the hidden state of the visual accumulator

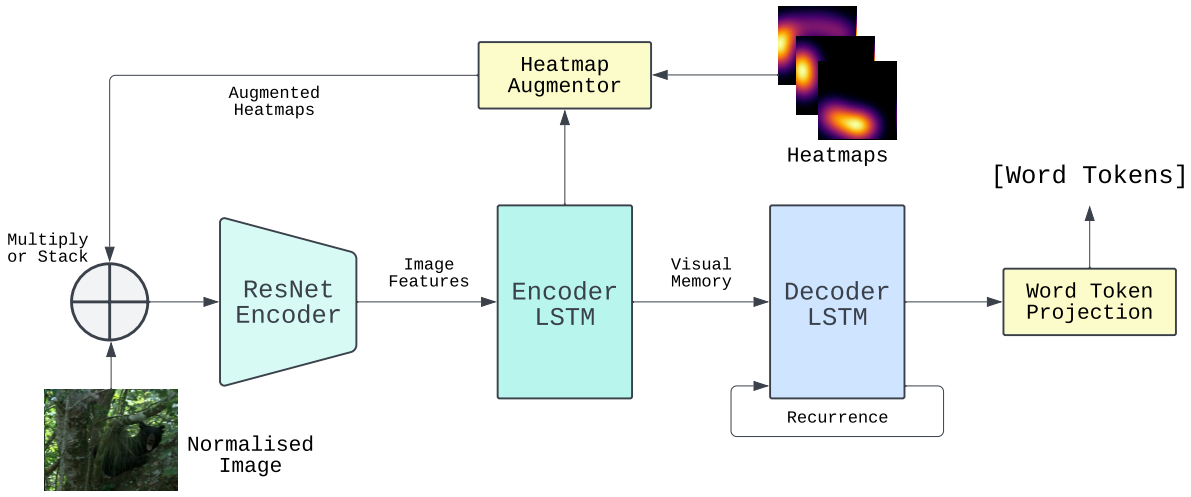


Figure 5.4: Visualisation of the Temporal ResNet captioning model.

is passed to the decoder, and the output caption is generated, in the same manner as previous sections. A visualisation of the architecture can be seen in Figure 5.4.

5.1.4 Temporal R-CNN Captioner

We now modify the Temporal ResNet architecture to use the Bottom-Up [Anderson *et al.* 2018] region proposals and bounding boxes, rather than a plain ResNet backbone with raw image data. Many models show improved performance when using R-CNN region features over raw image data or fixed size representations, as it allows visual components to attend to specific regions of the image more easily. It also allows systems to better utilise already learned knowledge from up-stream tasks such as object detection. The new architecture utilises a dual-LSTM encoder, where one LSTM accumulates the trace data over multiple time steps, and the other accumulates visual features, and we continue to utilise a single LSTM for the language decoding segment.

First, we attain the visual information from our region proposals. The image regions and their bounding boxes are extracted from the images as per Section 4.5, yielding 36 region feature and bounding box pairs. To standardise the size of the features and boxes, they are each linearly embedded to the size of the model using two fully connected layers – one for features and one for boxes. The embedded pairs are then element-wise summed to yield 36 tensors, which now represent the visual features with their associated positional information. We refer to this sequence as the visual embedding sequence.

The trace data must then be integrated into the model. We now make use of trace boxes directly rather than using generated attention heatmaps, extracted using the method discussed in Section 4.5. At each time step, the trace box is embedded using a fully connected layer. The embedding is then provided to the trace accumulator LSTM in order to generate an output tensor, which we refer to as the attention state. We note that the attention state is the output of the LSTM at each time step, and not the hidden memory state as has been used frequently elsewhere.

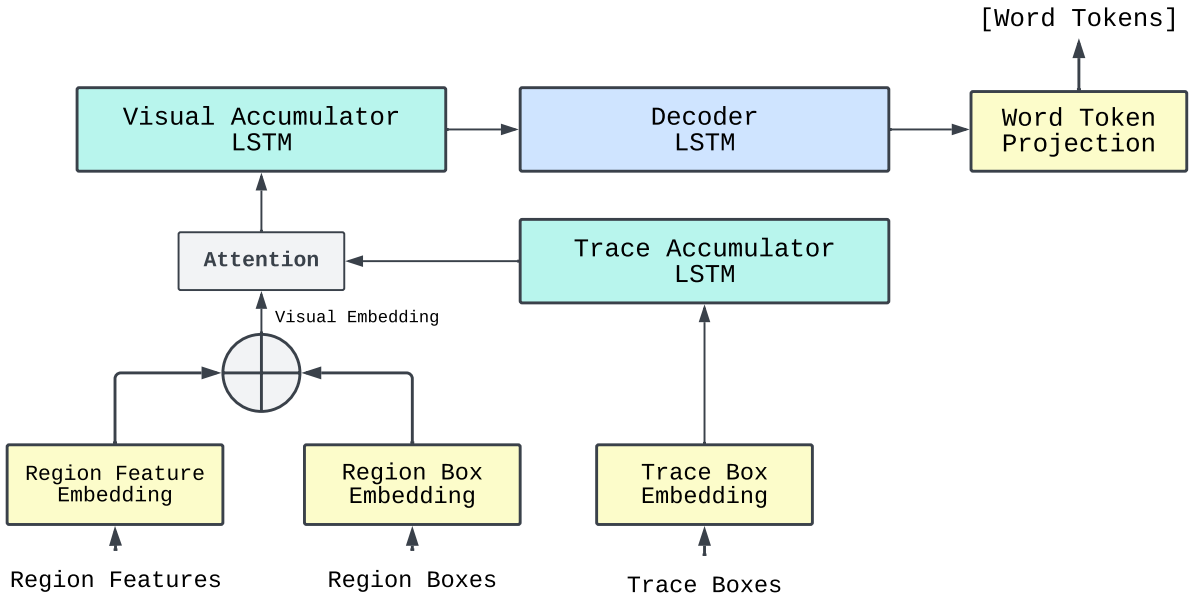


Figure 5.5: Visualisation of the Temporal R-CNN captioning model.

In order to combine the visual embedding sequence and the attention state at each time step, a multi-head attention mechanism is used. The attention state is provided as the query tensor, and the visual embedding sequence is used for both the key and value tensors. The attention mechanism then yields a single tensor representing the visual features attended to at this time step, which we refer to as the attended visual tensor. We make use of the attention state rather than directly using the trace box embedding here as it allows the model to learn long-term patterns in the trace, and apply that information in identifying visual regions of interest through the attention mechanism.

The attended visual tensor produced by the attention mechanism is then provided to the visual accumulator, and this process continues until the trace box sequence is exhausted. Once the trace sequence is completed, the hidden state of the visual accumulator is passed to the decoder, and the output caption is generated as in other models. A visualisation of the architecture can be seen in Figure 5.5.

5.1.5 Attended R-CNN Captioner

Our final modification on the LSTM architectures is the addition of another attention mechanism. Many approaches benefit from the application of attention over visual features, and many captioning models further benefit when using attention in the language decoder. This architecture follows the same design and process as the Temporal R-CNN Captioner above, however with an additional attention mechanism from the language decoder over the visual accumulator.

The encoding process remains the same as in the Temporal R-CNN Captioner, up to the point that the attended visual tensor is generated and provided to the visual accumulator LSTM. In this architecture the hidden state of the visual accumulator is retained

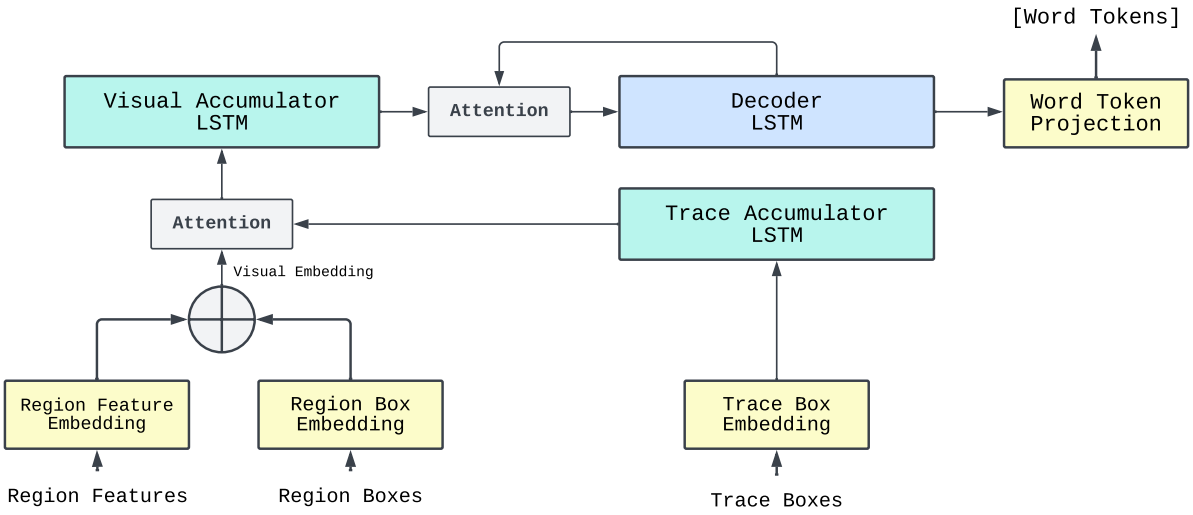


Figure 5.6: Visualisation of the Attended R-CNN captioning model.

at each time step, and added to a visual memory sequence. Once the trace sequence is exhausted, the visual accumulator then passes the memory sequence to the decoder.

The decoder begins with a zero-tensor hidden state and consumes the *[STR]* token to begin decoding. At each subsequent time step the decoder uses the last output as the query in the new attention mechanism, with the visual memory sequence acting as the keys and values. The output of this attention step is then element-wise summed to the current hidden state of the decoder, allowing the decoder to dynamically pull in segments of the visual memory while retaining its own state. The conversion of the decoder outputs to the caption follows the same approach as previous architectures. A visualisation of the Attended R-CNN architecture can be seen in Figure 5.6.

5.2 Transformers

We now progress to the Transformer-based approaches considered in this research. As previous work relying upon Transformer-based systems for this task already exists, it is clear that Transformer architectures are largely capable of handling the complexity of long descriptions, and we need not perform our own viability evaluation [Pont-Tuset *et al.* 2020; Meng *et al.* 2021]. Additionally, we primarily rely on region-based visual data as the Transformer architecture structure is better suited to it, and region-based systems have shown to outperform raw image systems.

5.2.1 Basic Joint Transformer

The first Transformer-based architecture is a straightforward approach utilising a single encoder and decoder for both modalities – visual and trace. As such, we must extract and combine both input modalities into a single sequence. The joint Transformer will be the baseline for the progressively more complex approaches to follow.

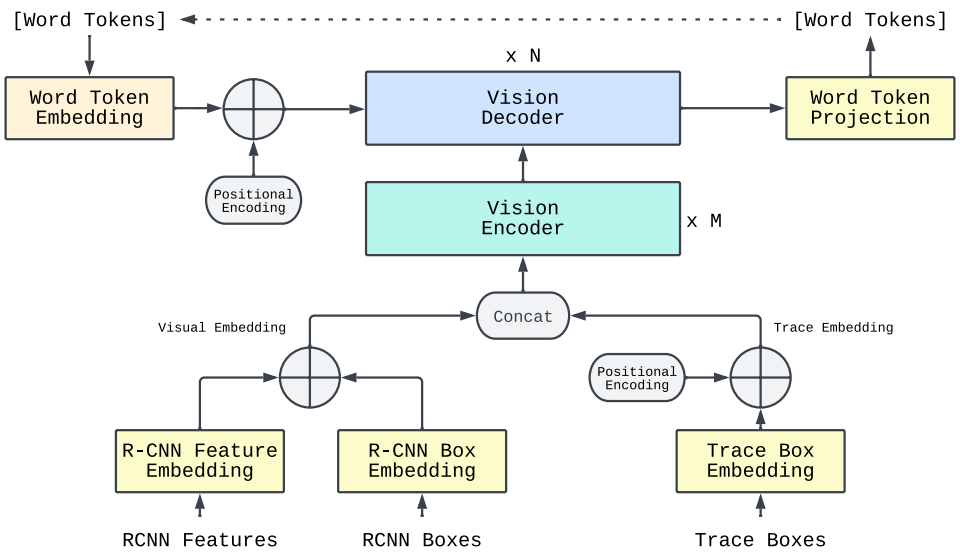


Figure 5.7: Visualisation of the Basic Joint Transformer model.

First the visual features must be attained. The region features and bounding boxes pairs are extracted from the image and processed as in Section 4.5. The pairs are embedded to the hidden size of the model using two fully connected layers – one for region features and one for bounding boxes. The embedded pairs are then element-wise summed together to yield 36 merged tensors. These tensors now contain an embedded form of the region features enriched with positional information from the bounding box, and form the visual embedding sequence.

For the trace information, the trace boxes are extracted from the mouse trace as explained in Section 4.5. As with the visual data, the trace boxes are embedded to the hidden size of the model utilising a fully connected layer. The embeddings are then positionally encoded using the sinusoidal approach discussed in Section 2.3.2, yielding our trace embedding sequence. The visual and trace embedding sequences are then concatenated together into a joint embedding of size $(r + t, h)$, where r is the number of regions, t the number of trace boxes, and h the hidden size of the model. This joint sequence is then provided to a traditional Transformer encoder to yield the joint encoding.

We now move to the decoder segment. First the input caption is pre-processed and tokenised as in Section 4.5. The token sequence is then embedded to the hidden size of the model using a learned embedding, acting as a lookup table for each token in the vocabulary. The embeddings are then positionally encoded using the same sinusoidal manner as above, yielding our caption embedding.

This caption embedding is then provided to a traditional Transformer decoder as input, and the joint encodings discussed prior are provided as memory. The decoder then generates the output encodings which are projected back to the vocabulary dimensions using a fully connected layer, and soft-maxed to yield the token logits. During inference the output caption attained from the logits is provided back to the decoder as input for iterative caption generation. A visualisation of the model can be seen in Figure 5.7.

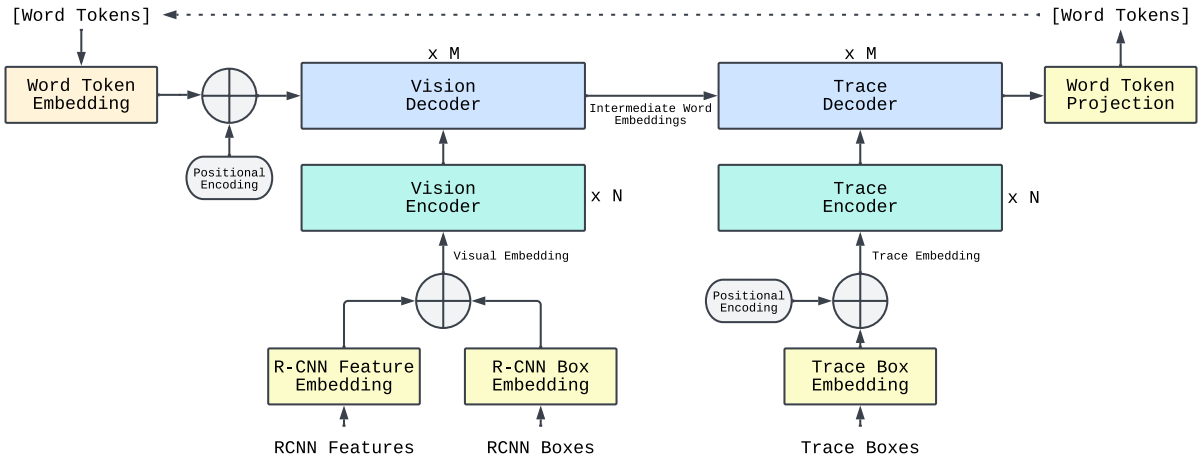


Figure 5.8: Visualisation of the Dual-Branch Transformer model.

5.2.2 Dual-Branch Transformer

One concern of the joint Transformer approach is that a single decoder and encoder are expected to handle two modalities simultaneously. Allowing the model to specialise a branch per modality reduces the complexity during self-attention of the sequence and allows more targeted weights. As such we split the joint Transformer into a dual-branch system, one for vision and one for trace. However, the dual-branch approach is still largely similar to the joint approach.

As in the joint approach the region features and bounding boxes are embedded and merged to give the visual embeddings, and the trace boxes are embedded and positionally encoded to give the trace embeddings. However, instead of being concatenated, now the visual embeddings are provided to the vision Transformer encoder, and the trace embeddings to a separate trace Transformer encoder. These now produce the independent visual and trace encodings respectively.

Following this, as in the joint approach, the input caption is embedded to the hidden size of the model and positionally encoded. The decoding process then involves two steps. In the first step handling the visual modality, the caption embedding is provided as input to the first Transformer decoder, and the visual encodings are provided as memory. The raw output from the first decoder is then provided as input to the second Transformer decoder which handles the trace modality, with the trace encodings now provided as memory. This second decoder then produces the caption output which is projected back to the vocabulary space as in the previous architecture, and fed back into the decoder system if necessary for inference. A visualisation of this architecture can be seen in Figure 5.8.

5.2.3 Cross-Modal Transformer

While the dual-branch approach may improve performance due to the specialisation of weights in each branch, it introduces a possible weakness in the encoding process. Where the joint encoder can share information between tokens of different modalities

through the self-attention step in each encoder layer, this is not possible in the dual-branch system as modalities each have their own encoder and as such their own self-attention. Thus we can re-introduce cross-modal attention in the encoders through the use of a cross-modal context sharing token.

Whereas the dual-branch system utilises two independent, traditional Transformer encoders, the cross-modal system utilises what we refer to as cross-modal encoders, which act as two branches within each encoder layer. In order to facilitate the information sharing, once the visual and trace embeddings have been generated as in the previous approaches they are each appended with an additional context sharing token. The initial value of each context token is generated by a learned embedding acting as a lookup table for each modality. The context tokens will be used to share information between the two encoding branches.

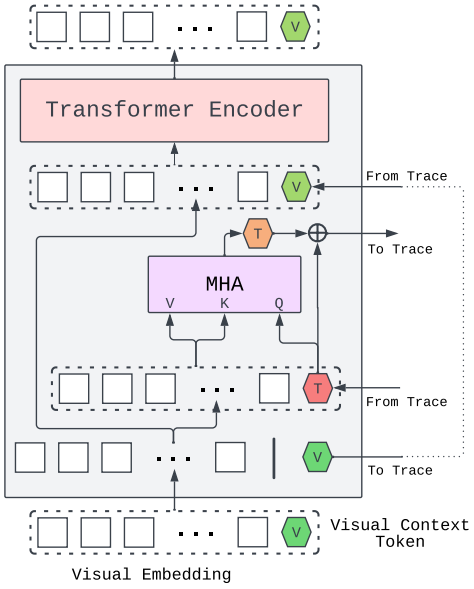


Figure 5.9: Visualisation of the Cross-Modal Encoder mechanism used for sharing contextual information between the different modality branches.

In the encoding process, each layer receives the output encodings of the previous layer – or the initial embeddings in the case of the first layer – as two separate sequences. The first sequence represents the visual modality, and the second the trace modality, each including their respective context token. Both cross-modal encoders within the layer take their respective sequence and split out the context token, which is sent to the other branch. The context token received from the other branch is then concatenated onto the current split sequence, and multi-head attention is performed using the context token as the query, and the concatenated sequence as the keys and values. The output of the attention step is then summed with the original context token, and this tensor will be the new context token for the other branch. Importantly, this new context token will now be carrying information attained from the sequence of the current branch. The new context token is then returned to the other branch, and re-concatenated with the original split sequence, and the entire joined sequence is provided to a traditional Transformer encoder layer.

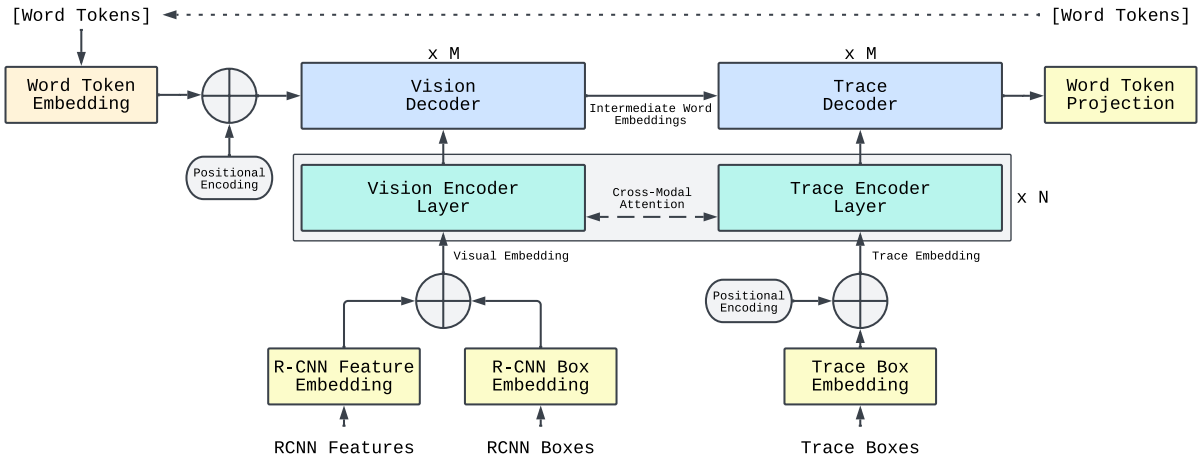


Figure 5.10: Visualisation of the Cross-Modal Transformer model.

We can formalise this process – for the visual branch – as follows:

$$x^c_{t,v} = \text{LN}(\text{cat}(c_v, x_t)) \tag{5.1}$$

$$c'_v = \text{MHA}(c_v, x^c_{t,v}, x^c_{t,v}) + c_v \tag{5.2}$$

$$x^c_v = \text{cat}(c'_v, x_v) \tag{5.3}$$

$$x^{c\prime}_v = \text{TEnc}(x^c_v) \tag{5.4}$$

Where LN is the layer normalisation operation, MHA is multi-head attention, cat is sequence concatenation. c_v is the visual context token and x_t is the split trace sequence, with c_v‘ and x^{c}_{v}‘ being the new context token and newly enriched visual sequence respectively. The process for the trace branch is identical, simply swapping v and t sub- and super-scripts. A visualisation of the cross-modal encoder mechanism can be seen in Figure 5.9.

The motivation for this context token system is two-fold. Firstly, by having the tokens swapped at each layer it allows the self-attention step to partially attend to important information from the branch of the other modality – while still keeping the branches largely independent. Secondly, it does this while avoiding reintroduction of the quadratic complexity that comes with combining the two sequences as in the joint encoder.

The remainder of the cross-modal architecture – namely the caption embedding and projection, as well as the overall decoding process – is identical to that of the dual-branch encoder. It should, however, be noted that the sequences provided to the decoders still contain the context tokens appended during the encoding process as they may still contain important information not present in the rest of the sequence. For completeness, a visualisation of the overall architecture is shown in Figure 5.10.

5.2.4 Alternating Cross-Modal Transformer

Our final dual-branch modification is to the decoder segment. In all of our dual-branch architectures thus far, the decoding process passes through all the visual decoder layers

before passing through the trace decoder layers. In a single layer decoder this is not a problem, however in deeper networks this means the network may lose information from the visual encoder and, even with residual connections, hinder propagation of gradients back to the earlier layers.

To address this concern, we modify the network to alternate modalities during the decoding process. In this network, each layer of the decoder is composed of two branches, as in our Dual-Branch system. In each decoder layer the outputs from the previous layer – or the initial caption embeddings in the case of the first layer – are passed through the first visual decoder branch, which uses the visual encodings as memory. The outputs from this first branch are then provided as input to the trace decoder branch, which uses the trace encodings as memory. These decoded outputs are then passed on to the next layer, or projected to the output caption.

This revised process now regularly exposes the decoded outputs to the encoded sequences from both encoder branches. This shortens the distance the gradients must flow back in order to reach the visual encoder, and also refreshes both input modalities throughout the decoding process, reducing the amount of information the outputs must retain through multiple layers. The network and embeddings remain identical aside from this modification, and a visualisation of the architecture can be seen in Figure 5.11.

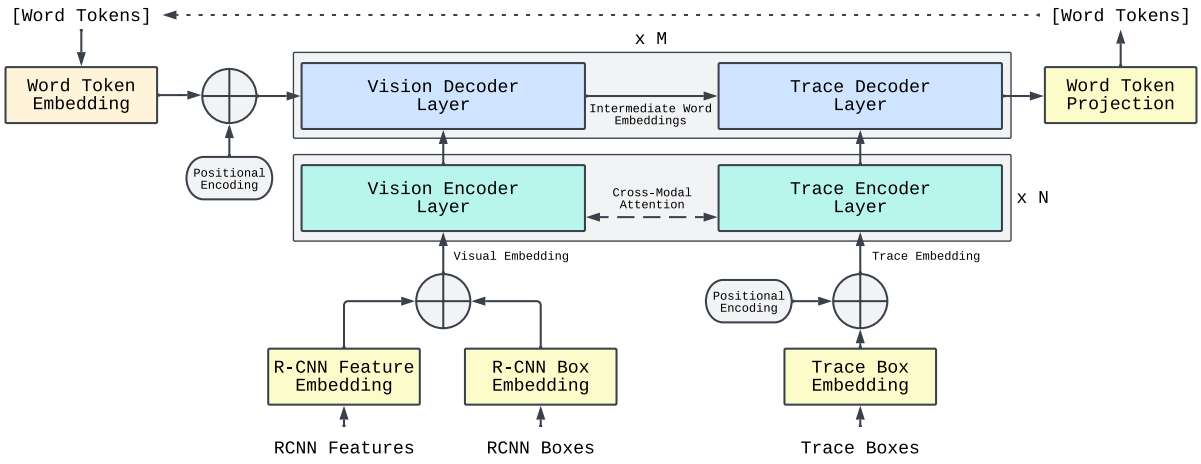


Figure 5.11: Visualisation of the Alternating Cross-Modal Transformer model.

5.2.5 Swin Joint Transformer

As an additional experiment, we construct a model that utilises a patch-based visual backbone rather than R-CNN region proposals. Recent work has shown purely Transformer-based visual systems as competitive with convolutional systems, so this may yield further improvements on our metric suite. For this we utilise the pre-trained

Hugging Face Swin Transformer model[1] as our visual backbone. The encoder and decoder aspects of the architecture and process are the same as that of the Basic Joint Transformer discussed previously.

First, the raw, normalised images are provided to the Swin Transformer, which encodes the image into a sequence of uniform sized patch feature tensors. These patches are then embedded to the hidden size of the model using a fully connected layer, in the same manner as the region features in the prior models. However, as the Swin model does not use strictly bounded regions of the image, we cannot encode position using bounding boxes. Instead, the patch embeddings are more simply positionally encoded using the same sinusoidal manner as the trace and caption tokens. The embedded patches are then concatenated with the embedded trace sequence – generated in the same manner as the Basic Joint Transformer – forming the joint sequence.

The joint sequence is then provided to the single Transformer encoder, and from this point the architecture and process is identical to that of the Basic Joint Transformer. More detail about the Swin architecture itself can be found in Section 2.4.3.

5.3 Summary

This chapter has presented in detail the series of architectures, alongside specific data requirements, upon which this research is based. The more traditional LSTM-based approaches are covered first, composed of the auxiliary language encoder-decoder, and four primary LSTM-based image captioning systems. The basic one-step captioning system is introduced, which acts as a straightforward benchmark for the remaining implementations, followed by the modifications that will be applied to it. The system will first be modified to include temporal attention heatmaps with the same ResNet backbone to allow for more complex relationships, before switching to an R-CNN region-based visual back-bone that utilises trace boxes which has shown better performance over raw image backbones. The final modification then introduces additional attention between the language and vision LSTMs.

Secondly, the primary Transformer-based architectures involved in this research are discussed. The basic joint architecture is covered, followed by the four additional region-based systems that build upon it. Alongside this is discussion on the mechanisms introduced in each architecture, leading to our final alternating decoding and cross-modal encoding model. Additionally, with the introduction of fully Transformer-based visual systems, some attention is dedicated towards a system utilising the Swin visual encoder as a backbone. Visualisations of each model are also provided, as well as descriptions of their encoding and decoding processes. The effectiveness of all these architectures is evaluated in the upcoming Chapter 6.

[1] https://huggingface.co/docs/transformers/model_doc/swin – 2022/12/28

Chapter 6

Results

This chapter now covers the results of the various architectures discussed in the previous chapters, and modifications thereof. Experimental configurations used for the models will be covered, such as hyper-parameters and device configurations, followed by the quantitative results each series of models achieved using these configurations, separated into RNN and Transformer sections. An ablative analysis of the Cross-Modal encoding mechanism is also performed to evaluate the effect it has in the systems. Lastly, a section on the qualitative analysis of the results for the Transformer architectures is presented. Given the poor performance of the RNN models a full qualitative analysis is not included, however discussion is provided on some of the emergent qualitative characteristics within the quantitative results section.

6.1 Experimental Configuration

All models were trained with learning rates of 10^{-4}, which we found to yield the best results, utilising the Adam optimiser [Kingma and Ba 2014]. We additionally experimented with the SGD optimiser, however this resulted in much slower convergence and infeasible training times. Final model weights were taken by storing the weights with the highest validation CIDEr score at any point during training. During inference beam search was utilised to improve output diversity and quality of captions, using a beam width of 5.

RNN models utilised hidden sizes for the LSTM layers of 2048, and ResNet-101 as the visual backbones. Modifying these to be larger did not noticeably affect metric performance, however did consequently slow training and inference. Given their large size and the slow training characteristics of LSTMS, RNN models were allowed to train up to a maximum of 15 epochs with a maximum batch size of 12. While this is less than the allowed epochs for Transformers, RNN validation scores indicated plateauing well before this stage, and the allowed 15 epochs already required 4 days of training time per run, making additional epochs infeasible. This validation score plateau can be seen in Figure 6.1 – notably the LSTM approaches plateau in substantially fewer epochs than the Transformer approaches. Additionally, due to the demanding training

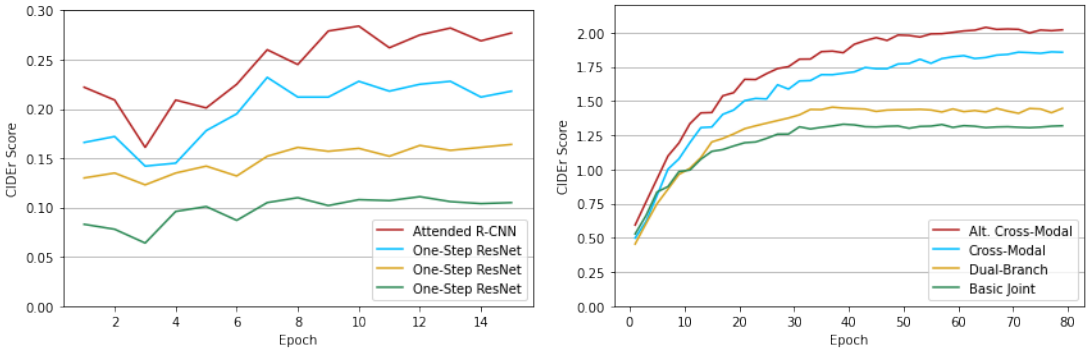


Figure 6.1: Validation CIDEr scores recorded during training for the LSTM-based models (left) and Transformer-based models (right).

characteristics and time, final metrics were averaged over two runs. RNN models were trained using an Nvidia Quadro RTX8000 and required approximately 64GB of RAM for training.

Our Transformer models utilised hidden sizes of 512, feed-forward sizes of 2048, and 8-headed attention for all components of the system. Modifying these generally yielded no improvement on metrics, and in some cases would result in reduced validation and testing performance. Transformer models were allowed to train to a maximum of 80 epochs using a batch size of 30, and final results were averaged over three runs. Each region-based run required approximately 20 hours to train, using an Nvidia RTX3090 and 48GB of RAM. The Swin patch-based approach required approximately 48 hours to train under the same conditions.

6.2 Quantitative Results

For our quantitative results we report on the BLEU-1, BLEU-4, METEOR, ROUGE$_L$, CIDEr, and SPICE metrics where possible, as is standard with captioning models. We also include the original results of the basic Localized Narratives [Pont-Tuset *et al.* 2020] and best MITR [Meng *et al.* 2021] implementations for easy comparison. It should be noted that Localized Narratives splits their trace by fixed time step of 0.4s rather than per-word as in our work and the MITR implementation, and thus is not fully comparable.

6.2.1 RNNs

The final results of our models can be seen in Table 6.1. As can be seen, the RNN captioning implementations fall well below the existing literature on this task.

The basic Encoder-Decoder system achieves reasonable scores on the metric suite, comparative to scores of human captioning annotators and translators given the likely variance in order and word choice. However, it is worth re-stating that the basic Encoder-Decoder is simply an auto-regressive caption-to-caption system and is not image based. While ideally there should be no variance for the Encoder-Decoder task – and as such it

Method	L_E	L_D	BLEU-1	BLEU-4	METEOR	$ROUGE_L$	CIDEr	SPICE
Localized Narr.	2	2	0.522	0.246	N/A	0.483	1.065	0.365
MITR	2	2	0.607	0.292	0.263	0.487	1.485	0.317
Encoder-Decoder	1	1	0.797	0.554	0.523	0.791	4.332	0.563
One-Step ResNet	1	1	0.224	0.041	0.039	0.177	0.112	0.065
Temporal ResNet	1	1	0.312	0.062	0.053	0.213	0.165	0.086
Temporal R-CNN	1	1	0.348	0.079	0.073	0.269	0.235	**0.134**
Attended R-CNN	1	1	**0.374**	**0.100**	**0.092**	**0.289**	**0.288**	0.127

Table 6.1: Results for our RNN models on the test split for the COCO subset of Localized Narratives. L_E and L_D indicate the number of layers in the encoder and decoder respectively. We include the Localized Narratives [Pont-Tuset *et al.* 2020] and best MITR [Meng *et al.* 2021] implementation, as well as our viability encoder-decoder, for ease of comparison. Bolded values represent the best result within our own RNN model series.

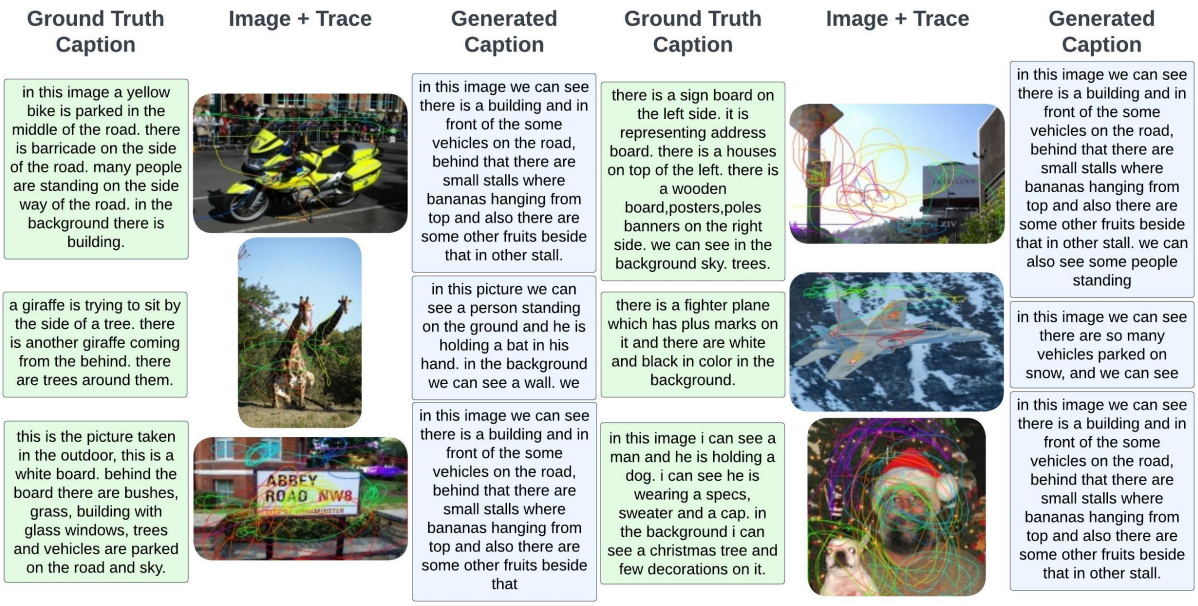


Figure 6.2: Example captions generated by the Attended R-CNN model taken from our test set. While extensive repetition of training captions and severe mismatch between the caption and image content is notable, the model appears slightly capable in identifying large features like buildings and vehicles, and matching paragraph length.

should not be comparable to captioning or translation – it does indicate that an LSTM system is at least capable of generating sensible captions in a perfect environment, and provides a reasonable upper bound for the performance of the remaining LSTM models.

However, the captioning systems themselves do not fare as well. The One-Step captioning system performs the worst of all listed architectures in every metric, and tends to either output an "average case" caption composed of the most common phrases or simply generate a non-sensible caption. This is not unexpected, as the model is very rudimentary and attempts to complete a highly complex visual task in a single leap.

Adding in the temporal nature as in the Temporal ResNet model immediately improves the scores on all metrics fairly substantially, with an approximately 40% improvement on all scores. However, the generated captions still lack relevance to the visual features and again are often simply the average case or slightly more sensible unique captions than the One-Step model. One noticeable characteristic improvement is that the output captions are often closer to the length of the expected caption than in the One-Step model. This is because the system can better discern the number of output tokens by using the number of input trace boxes. This addition is likely the primary component of the metric improvements in the system.

Introducing R-CNN features in place of a straightforward ResNet visual backbone further improved each metric by approximately 15-50%. The largest of these improvements occurred on metrics typically considered better aligned to human judgement or scene understanding, namely CIDEr and SPICE, which correlates with our prediction that region-based systems would better capture important aspects of the images. Another improved characteristic after this change is that generated captions are often closer aligned to their reference captions, and the model more frequently provides uniquely generated captions relevant to the image. Additionally, these unique captions tend to be more coherent and better grammatically structured than previous models. While substantial re-use of a set of training captions is still apparent, the used captions at least tend to align to the visual features of the image rather than being seemingly arbitrary. Some examples can be seen in Figure 6.2.

The final addition of the extra attention mechanism between the language and visual components provided a further moderate improvement of approximately 10-20% over each metric. This is likely due to the language component now being able to utilise richer per-step information from the visual component which before was lost during the compression into a single fixed-size tensor. While there are no apparent characteristic changes from the previous model, it appears the overall quality of the generated captions has improved. However, in general the quality of captions and metric scores fall well below existing methods and are not feasible for real world application.

6.2.2 Transformers

The results of our Transformer-based systems can be seen in Table 6.2. The Transformer systems fared substantially better than the RNN-based implementations, and as can be seen from the results even compete with or outperform existing SOTA methods on

Method	L_E	L_D	BLEU-1	BLEU-4	METEOR	$ROUGE_L$	CIDEr	SPICE
Localized Narr.	2	2	0.522	0.246	N/A	0.483	1.065	**0.365**
MITR	2	2	0.607	0.292	0.263	0.487	1.485	0.317
Attended R-CNN	1	1	0.374	0.100	0.092	0.289	0.288	0.127
Basic Joint	1	1	0.535	0.228	0.237	0.436	1.284	0.309
Dual-Branch	1	1	0.551	0.251	0.244	0.445	1.439	0.313
Cross-Modal	1	1	0.570	0.272	0.255	0.466	1.603	0.332
Cross-Modal	4	3	0.598	0.304	0.270	0.491	1.852	0.350
Alt. Cross-Modal	4	3	**0.609**	**0.318**	**0.278**	**0.508**	**2.023**	0.363
Swin Joint	2	2	0.497	0.212	0.197	0.412	1.231	0.298

Table 6.2: Results for our Transformer models on the test split for the COCO subset of Localized Narratives. L_E and L_D indicate the number of layers in the encoder and decoder respectively. We include the Localized Narratives [Pont-Tuset *et al.* 2020] and best MITR [Meng *et al.* 2021] implementation for ease of comparison.

the long captioning task. Example captions can be found in the upcoming qualitative analysis section in Figure 6.4 and Figure 6.5, alongside a larger sample in Appendix A.

The baseline Basic Joint system composed of a simple one-layer joint-modality Transformer greatly outperforms even the best of our RNN implementations on every metric. The conceptually straightforward approach manages well with the diversity of the caption set and already captures most visual features in the images. However, the model has difficulty with generating correct words consecutively as seen by the relative difference in BLEU-1 and BLEU-4, and at times does not include visually important features specific to the image in the caption, as seen in the relatively low CIDEr score. Nevertheless, the generated captions are often relevant to the image and grammatically sensible, although still below the quality viable for real world applications.

The Dual-Branch modifications to the baseline Basic Joint architecture then provided a slight improvement over most metrics, with increases between 1-3%. However, scores on BLEU-4 and CIDEr – typically more difficult metrics – both increased by approximately 10%. A larger increase in BLEU-4 indicates the system is now better at generating strings of correct words, and CIDEr indicates a better encapsulation of unique visual features in the images. This would match our hypothesis as the dual-branch structure should allow the system to specialise weights per-modality and thus better follow the trace through the image, and pick up finer visual details.

Our primary modification, the Cross-Modal encoding system, then provides similar gains on top of our Dual-Branch system. Over most metrics we see a larger increase of approximately 4-5%, and again note an approximately 10% increase on the CIDEr metric, 8% on BLEU-4, and additionally an additional 6% on SPICE. All of these indicate further refinements in the ability of the model to identify unique visual aspects of the images, and additionally caption them in the correct order with higher quality. This fits our expectations of the cross-modal encoding process, which we envisioned would yield

a stronger relation between the visual features and trace path and holistically improve our captions.

In order to evaluate the alternating decoding process, we also increase the number of layers in our Cross-Modal system. We first increase the layers without modifying the architecture to act as a benchmark for the alternating decoding approach. Performance on our metrics plateaued after $E = 3$ and $D = 4$ layers using a narrowed grid-search – where greater values caused training characteristics and performance to degrade – and as such we selected these as our final layer counts. As expected this straightforward increase in the size of the model yielded fairly substantial improvements alone, with increases on various metrics ranging from 5%-15% over the smaller Cross-Modal model, although at the cost of an approximately $3\times$ increase in the number of parameters.

Modifying the Cross-Modal system to now use the alternating decoding process provides the final improvement on our metric suite. We see an increase ranging from 3%-10% on each metric over the previous model, with the most substantial once again being the CIDEr metric. These increases likely stem from the improved training characteristics of the system – as gradients can now reach both the visual and trace components more easily, as well as the inference characteristics – as the memory of each modality is accessed across a wider time span rather than in one pass. We also note that this improvement comes with no increase in the number of parameters in the system or time requirements during training and inference, and introduces minimal change to the overall process.

Our last approach, the Swin Joint system, unfortunately under-performs the Basic Joint benchmark system even when utilising more layers. The Swin approach shows decreases ranging from 3%-16% across the metric suite versus the baseline, even with double the number of layers, with the largest decrease in the METEOR score, followed by the BLEU-4 metric. This would imply either that our approaches are not as well suited to the Swin visual encoding as they are to the Faster R-CNN regions, or that the Swin encoding system does not provide as much information – or provides it in a less easily accessible format – reducing the performance ceiling of our systems. Additionally, the Swin approach required more training and inference time than the R-CNN approaches due to the need to generate our own patches and fine-tune the system.

6.2.3 Cross-Modal Encoding

The Cross-Modal encoding is the primary contribution of interest which we introduce in our research, and as such we wish to investigate its effect more thoroughly on our system. From the final results in Table 6.2 we already see that the Cross-Modal encoder outperforms both the Basic Joint benchmark and Dual-Branch architectures on our metric suite, as the first sign that our mechanism provides a tangible improvement to our model. Additionally, it forms part of the system that outperforms the MITR [Meng *et al.* 2021] and Localized Narratives [Pont-Tuset *et al.* 2020] implementations, which represent the current SOTA for long-text image description.

However, we also wish to see the interactions of our shared context tokens with the remainder of the system during the encoding process. We can do this by extracting

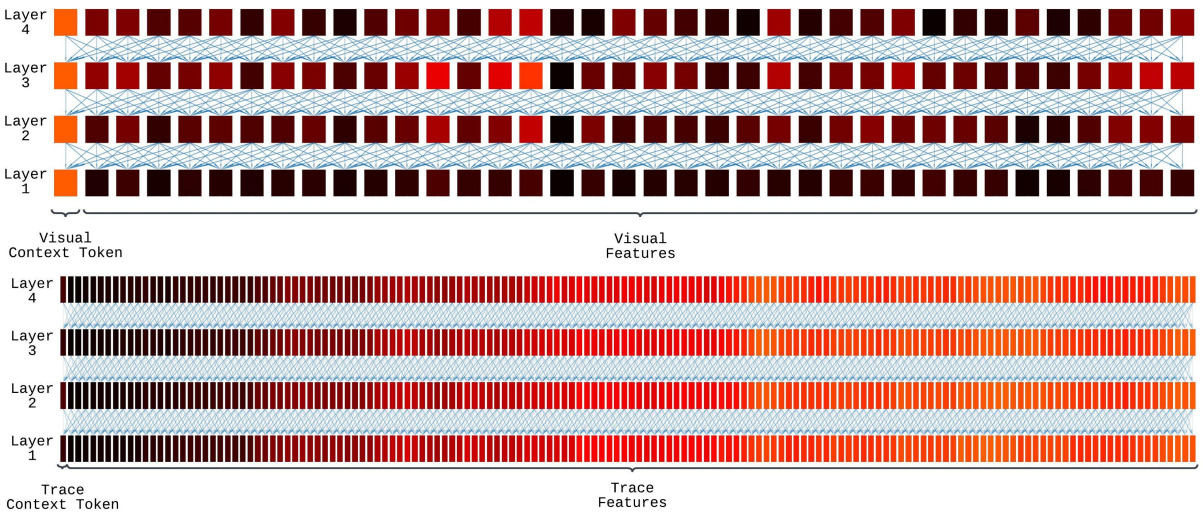


Figure 6.3: Visualisation of the self-attention weights from our Cross-Modal Encoder layers averaged over a batch of 16 images. Brighter colours indicate higher relative weighting by other tokens during self-attention. Top is the visual branch, bottom is the trace branch.

the self-attention weights from each encoding layer of our multi-layer Cross-Modal encoders, allowing us to determine the relative importance given to the context tokens according to the other tokens in the sequence. We do so by taking the self-attention encoding weights for a batch provided to the model, and for each token average the weight assigned to the token from each other token in the sequence, accounting for masking. This weighting is calculated per-sequence before being averaged for the entire batch and normalised. A visualisation of the importance rating can be seen in Figure 6.3.

We note interesting behaviour which varies between each modality. The visual modality encoders assign strong importance to the context token which pulls information from the trace encoder throughout all of the layers. This strong importance may indicate that the visual branch values the trace data in determining which regions are of the most value. The importance of the context appears to be highest during the early layers, before the individual visual tokens become more refined and relatively important during the later layers.

The trace modality on the other hand provides less relative importance to the context token and instead seems to provide more weight to tokens later in the sequence, with hot spots at positions in the sequence covering salient regions. This low importance of the earlier tokens is likely caused by the repeated usage of filler phrases such as “in this image we see” which are not visually grounded, meaning the associated mouse trace segments are less relevant. The relatively low importance of the context token itself indicates that the trace modality may be more focused on aggregating information about the shape or flow of the trace, rather than integrating visual information, allowing the decoders to combine the information instead.

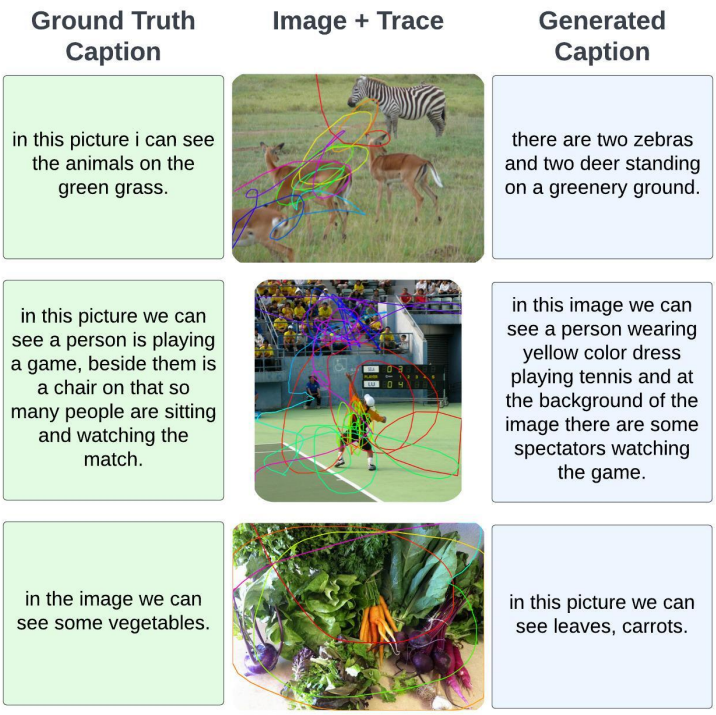


Figure 6.4: Example ground truth captions and predictions from the test set where predictions scored poorly while being qualitatively sound.

6.3 Qualitative Results

While quantitative metrics provide insight and allow straightforward comparison of models, they do not provide the full picture of the performance of the model. As such, we perform a series of qualitative evaluations of our models to provide a deeper understanding of the shortfalls and emergent characteristics.

The first point of interest is examining the well known weakness of existing captioning and translation evaluation metrics. While various metrics exist, each attempting to compensate for shortfalls in others, many still fail to fairly evaluate captions that are valid but do not use the same words or focus on the same objects in the image. A series of such captions can be seen in Figure 6.4, where the predicted captions from our Alternating Cross-Modal model are scored poorly due to the different choice of words to that of the annotator. In some cases the predicted captions could even be considered qualitatively better than the ground truths in terms of word diversity, the specificity of described regions, and the use of grammar.

However, such cases appear to be a minority, and in general the model is capable of matching the wording and region ordering of the narrator. In Figure 6.5 we show a series of example ground truth and predicted captions alongside their associated images. It is clear that the system is able to capture concepts throughout the image and, using the trace, describe them richly and in the order desired by the annotator. The system is capable of expressing the position of an object within the image suitably, and can differentiate between foreground and background, as well as the sides, corners

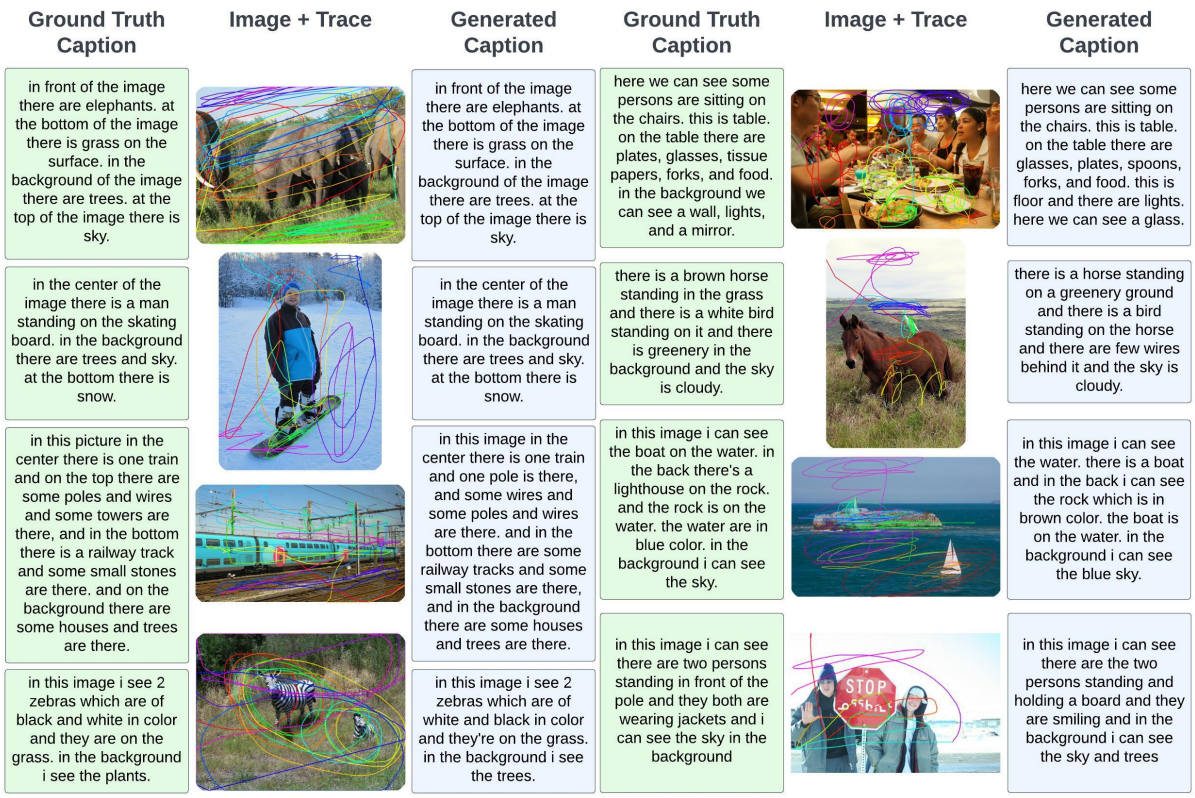


Figure 6.5: A selection of example ground truth and predicted captions with their associated images, taken from the test set.

and centre of the image depending on the context. In the majority of cases it can also identify the object or region of interest from a trace segment even when the segment does not fully overlap it or goes beyond its boundary. The generated captions are also almost always grammatically correct in structure, keeping in mind that the ground truth captions themselves sometimes contain incorrect grammar and can often be clumsily worded. Additional captioning examples can be found in Appendix A.

Additionally, by splitting traces by word it allows us to indirectly control the desired caption length of the model. As such, even in cases where the model may not be able to predict the correct words, it can still generate a caption of a suitable length or with appropriate verbosity, which can be clearly seen in the distribution of caption lengths shown in Figure 6.6. Notably, this phenomenon is present in all of the models, and even while the Alternating Cross-Modal system substantially outperformed the Basic Joint model on our metric suite, both models appear equally capable of generating captions of the desired lengths.

While similarity in the lengths of captions is important, arguably more important is the content itself. Thus, we also perform an analysis of the different words composing the ground truth and predicted captions from our models, the results of which can be seen in Table 6.3. Looking only at the average length of generated captions, all of our models appear close to the ground truth statistics. However, more specific breakdowns of the captions show a clearer differentiation of the approaches. Our best approach,

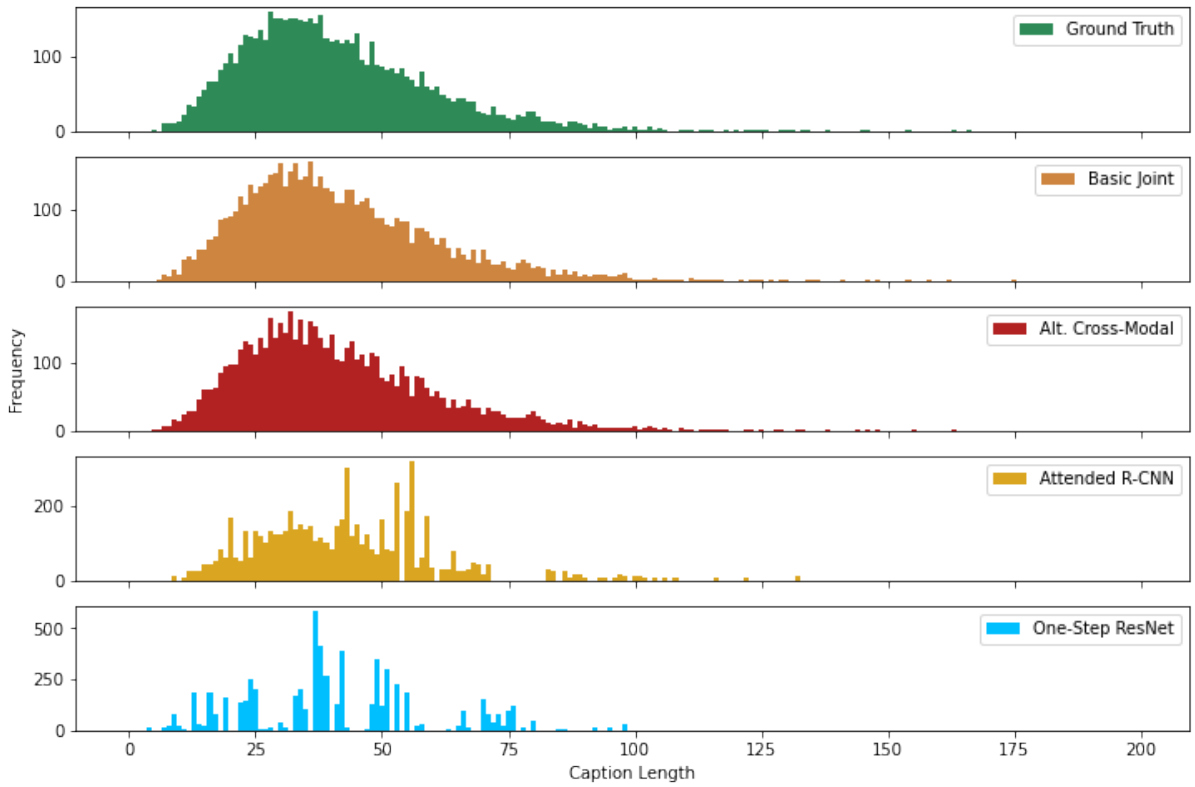


Figure 6.6: Frequency of ground truth and predicted caption lengths in our test set.

the Alternating Cross-Modal system, utilises more unique words on average than our Basic Joint approach, and was closer to the expected ground truth average. And while the Alternating Cross-Modal also used fewer nouns than the Basic Joint approach, it included more verbs and adjectives indicating a better capability of *describing* what it sees in the image, rather than just listing the objects contained within it.

Also of interest are the statistics from our Attended R-CNN approach. While its results on our metric suite are very poor, the breakdown of captions from its predictions would appear to be reasonably comparable to the ground truth data – with many unique words on average and a good composition of verbs and adjectives. This further demonstrates the tendency in the LSTM approaches to resort to using a series of similar "average" captions learned from the training data, instead of attempting to generate a tailored caption for the image. The re-use of learned captions, even for images that are not suitable, results in grammatically sound and rich predictions that are poorly scored by the metrics because they do not match the desired ground truth.

However, even in our best performing approach, there are a number of poor characteristics in the system. At times the generated captions can refer repeatedly to the same object or feature of the image when it should appear only once, such as the poles and wires in the third image of Figure 6.5, or the glasses in the fifth image. The model may also incorrectly count the number of objects in an image, such as in the first image of Figure 6.4 where it lists two zebras instead of one, or misidentify objects, such as referring to a yellow dress instead of an orange shirt in the second image of the same

	Words	Unique	Nouns	Verbs	Adj.
Ground Truth	41.22	26.51	12.35	4.07	2.24
Attended R-CNN	42.57	30.22	9.79	4.46	2.07
Basic Joint	41.70	23.26	12.61	3.87	1.98
Alt. Cross-Modal	41.34	25.07	12.12	3.99	2.07

Table 6.3: Qualitative part of speech metrics on our test set, comparing the average caption part-of-speech composition of the ground truth data alongside our baseline Transformer and final R-CNN and Transformer model predictions.

figure. Nevertheless, the system is in the majority of cases capable of describing images at a near human level of quality, and could be viable for real world applications.

6.4 Summary

In this section we have presented the qualitative and quantitative results of our various approaches in the long-text description task. We first provide the summarised experimental configurations that were used for our evaluation runs as well as the relevant training requirements and limitations of each system. We then presented the quantitative results of both our RNN- and Transformer-based approaches on our metric suite, and showed the Transformer approaches to be substantially more capable of the description task. Within these sections we discussed the progressive modifications made to each architecture and the impact they had both on the metric scores of the model as well as the characteristics of its predictions. We also discuss in detail the impact of our Cross-Modal Encoder mechanism and provide insight into how it is utilised by the model. Finally, we provide a qualitative analysis of our Transformer approaches with an evaluation of their use of language, as well as the limitations and shortcomings of our final implementation.

Chapter 7

Conclusion

7.1 Future Work

The primary limitation of our work is its reliance on the trace data included in the ground truth information, as well as the need to split the trace by word. While the reliance on trace data does allow us to guide the model in what it describes, it can also limits real world applications. A key piece of future work would be alleviating this limitation, and there are various possible approaches to this. The first could be the exclusion of trace data entirely, instead utilising control codes to guide the model in the verbosity and length of its descriptions, and the regions it is required to describe. Alternatively, an additional mechanism could be developed to generate attention traces or boxes with high enough quality so as to replace the ground truth trace data without having to modify the current architectures.

Another region of interest would be introducing new metrics tailored specifically to long-text image descriptions which account for the diversity in regions described, the order thereof, and the choice of words for each region. Naturally this would be a challenging task for the same reasons current metrics have difficulty, however recent metrics are more robust than older metrics to these difficulties and new research into the field could make use of advancements in related vision and language systems as part of their evaluations.

The final region of future work could be the implementation of knowledge transfer in the language domain. Currently the visual domain already utilises knowledge transfer via visual backbones, and in some cases pre-trained encoders, however there is no such activity in the decoder or language components. Utilising large, pre-trained systems such as BERT to instil language understanding into the model beyond what the dataset itself contains could improve the predictive abilities of the model. Additionally, embedding systems such as BERT utilise semantic embedding spaces, which could lead to models capable of zero-shot captioning.

7.2 Contributions

This research provides several contributions to the existing academic literature on long-text image captioning. The first contribution is the exploration of a series of LSTM-based captioning models, where we find that LSTM-based approaches are not well suited to this problem. Our second contribution is the introduction of a series of Transformer-based captioning systems, with our final model achieving SOTA performance on the task. As part of this contribution, we introduce a new cross-modal encoding context sharing mechanism based on the work of Chen *et al.* [2021], as well as a novel alternating decoding technique for multi-modal Transformer inference. Both the cross-modal encoding and alternating decoding are shown to improve the performance of our models on the common metric suite.

7.3 Discussion

In this research we have presented a variety of approaches to the challenging task of long-text image captioning, and introduced our novel Cross-Modal Encoder system for use in multi-modal Transformer-based systems. The task of long-text image captioning has only recently received attention due to the limitations of previous approaches and historical lack of openly available training data, and only a very limited number of viable implementations exist at the time of this work. As such, the primary goal of this work was to research the viability of several captioning approaches, and develop a new deep learning architecture capable of generating rich descriptions of images comparable to SOTA.

We develop two sets of architectures which combine visual and attention trace features, the first set being LSTM-based and the second being Transformer-based. Our approaches make use of the COCO subset of the newly released Localized Narratives [Pont-Tuset *et al.* 2020] dataset, the first to provide long-text captions for images, as well as associated mouse traces time synchronised to the caption. For our visual features we make use of the Bottom-Up region proposal system [Anderson *et al.* 2018] which segments images into a series of regions with associated region features and bounding boxes.

Our LSTM-based implementations were based upon a ResNet101 backbone feeding into one or more LSTMs. The initial One-Step ResNet model attempted to caption the image with no additional attention or temporal information in the simplest manner possible. This architecture was then augmented to include temporal attention heatmaps for image encoding, alongside an additional LSTM to accumulate visual information, in the Temporal ResNet model. We also experimented with replacing the plain ResNet extracted visual features with the Bottom-Up R-CNN region data, and introducing visual attention in our Temporal R-CNN approach, and as well as an additional attention mechanism from the language LSTM to the visual LSTM in the Attended R-CNN approach.

Our Transformer implementations primarily used the Bottom-Up region proposals. Our initial baseline Transformer architecture was the Basic Joint approach, which took

inspiration from the Localized Narratives benchmark implementation. The approach consumed the combined region and trace sequences in a single Transformer encoder and decoder. We then experimented with breaking the Transformer into one branch per modality instead of a combined format in our Dual-Branch implementation. To re-introduce attention between the modalities, we implement our novel Cross-Modal Encoder mechanism for each branch in our Cross-Modal implementation, which takes inspiration from the Cross-ViT architecture [Chen *et al.* 2021]. This cross-modal mechanism enables the different branches of our encoder to share contextual information between modalities while reducing computational demands. We also experiment with a novel alternating decoding process in our Alternating Cross-Modal approach which improves the flow of information through our models without incurring any computational overhead. We later show that these novel additions contribute to the SOTA performance of our final model. As an additional exercise we also experiment with replacing the R-CNN region proposals with Swin extracted image patches in our Swin Joint architecture.

We evaluate our implementations on a suite of commonly used translation and captioning evaluation metrics, namely BLEU, ROUGE, METEOR, CIDEr, and SPICE, and provide comparisons against the Localized Narratives benchmark architecture, as well as the more advanced MITR implementation of Meng *et al.* [2021]. We find that while the progressive modifications to our initial LSTM approach do in fact improve its performance on our metric suite, the final architecture still falls well below existing work, and is not viable for realistic applications.

On the other hand, the Transformer-based approaches fare substantially better, and each of our modifications – bar the Swin encoding – progressively increases performance on our metric suite, with our final implementation outperforming both existing works which represent the SOTA at this time. We also analyse the impact of our novel Cross-Modal Encoder mechanism, which improves the metric performance of our model and has a clear importance in the pre-existing attention mechanisms of the encoding process. Examples showing the capability of our models to generate rich, grammatically correct descriptions are also included, as well as a discussion on some of the emergent strengths and shortfalls in test predictions. Additionally, we perform a qualitative analysis of our Transformer implementations and show they utilise a diverse set of words, with average caption lengths, unique words, and numbers of nouns, adjectives, and verbs closely matching the ground truth data.

References

[Al-Malla *et al.* 2022] Muhammad Abdelhadie Al-Malla, Assef Jafar, and Nada Ghneim. Image captioning model using attention and object features to mimic human image understanding. *Journal of Big Data*, 9(1):1–16, 2022.

[Alahmadi and Hahn 2022] Rehab Alahmadi and James Hahn. Improve image captioning by estimating the gazing patterns from the caption. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1025–1034, 2022.

[Anderson *et al.* 2016] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*, pages 382–398. Springer, 2016.

[Anderson *et al.* 2018] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.

[Ba *et al.* 2016] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

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

[Banerjee and Lavie 2005] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[Bengio *et al.* 2015] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

[Bird *et al.* 2009] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[Chen and Zhao 2018] Shi Chen and Qi Zhao. Boosted attention: Leveraging human attention for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 68–84, 2018.

[Chen *et al.* 2021] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021.

[Cho *et al.* 2014] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[Desai and Johnson 2021] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11162–11173, 2021.

[Donahue *et al.* 2015] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[Dosovitskiy *et al.* 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[Girshick *et al.* 2014] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[Graves 2012] Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012.

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

[He *et al.* 2019] Sen He, Hamed R Tavakoli, Ali Borji, and Nicolas Pugeault. Human attention in image captioning: Dataset and analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8529–8538, 2019.

[Hochreiter and Schmidhuber 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Hossain *et al.* 2019] MD Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CsUR)*, 51(6):1–36, 2019.

[Jia *et al.* 2015] Xu Jia, Efstratios Gavves, Basura Fernando, and Tinne Tuytelaars. Guiding the long-short term memory model for image caption generation. In *Proceedings of the IEEE international conference on computer vision*, pages 2407–2415, 2015.

[Jiang *et al.* 2015] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. Salicon: Saliency in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1072–1080, 2015.

[Johnson *et al.* 2016] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4565–4574, 2016.

[Karpathy and Fei-Fei 2015] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[Karpathy 2015] Andrej Karpathy. *The Unreasonable Effectiveness of Recurrent Neural Networks*. TheUnreasonableEffectivenessofRecurrentNeuralNetworks, 2015. Accessed: 2023-07-26.

[Kingma and Ba 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Krishna *et al.* 2016] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016.

[Krishna *et al.* 2017] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

[Kuznetsova *et al.* 2020] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, 2020.

[Lamb *et al.* 2016] Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. *Advances in neural information processing systems*, 29, 2016.

[Lin *et al.* 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[Lin 2004] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[Liu *et al.* 2021] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[Lu *et al.* 2017] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 375–383, 2017.

[Meng *et al.* 2021] Zihang Meng, Licheng Yu, Ning Zhang, Tamara L Berg, Babak Damavandi, Vikas Singh, and Amy Bearman. Connecting what to say with where to look by modeling human attention traces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12679–12688, 2021.

[Min and Corso 2021] Kyle Min and Jason J Corso. Integrating human gaze into attention for egocentric activity recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1069–1078, 2021.

[Olah 2015] Christopher Olah. *Understanding LSTM Networks*. https://colah.github.io/posts/2015-08-Understanding-LSTMs, 2015. Accessed: 2023-07-26.

[Pan *et al.* 2020] Yingwei Pan, Ting Yao, Yehao Li, and Tao Mei. X-linear attention networks for image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10971–10980, 2020.

[Papineni *et al.* 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[Pascanu *et al.* 2013] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. Pmlr, 2013.

[Plummer *et al.* 2015] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

[Pont-Tuset *et al.* 2020] Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *European conference on computer vision*, pages 647–664. Springer, 2020.

[Qiao *et al.* 2018] Tingting Qiao, Jianfeng Dong, and Duanqing Xu. Exploring human-like attention supervision in visual question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[Radford *et al.* 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[Ren *et al.* 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[Rumelhart *et al.* 1986] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[Sharma *et al.* 2020] Himanshu Sharma, Manmohan Agrahari, Sujeet Kumar Singh, Mohd Firoj, and Ravi Kumar Mishra. Image captioning: a comprehensive survey. In *2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC)*, pages 325–328. IEEE, 2020.

[Simonyan and Zisserman 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Sood *et al.* 2020] Ekta Sood, Simon Tannert, Philipp Müller, and Andreas Bulling. Improving natural language processing tasks with human gaze-guided neural attention. *Advances in Neural Information Processing Systems*, 33:6327–6341, 2020.

[Sugano and Bulling 2016] Yusuke Sugano and Andreas Bulling. Seeing with humans: Gaze-assisted neural image captioning. *arXiv preprint arXiv:1608.05203*, 2016.

[Sun *et al.* 2020] Xiaoshuai Sun, Xuying Zhang, Liujuan Cao, Yongjian Wu, Feiyue Huang, and Rongrong Ji. Exploring language prior for mode-sensitive visual attention modeling. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4199–4207, 2020.

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

[Takmaz *et al.* 2020] Ece Takmaz, Sandro Pezzelle, Lisa Beinborn, and Raquel Fernández. Generating image descriptions via sequential cross-modal alignment guided by human gaze. *arXiv preprint arXiv:2011.04592*, 2020.

[Tan and Bansal 2019] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

[Thomas 2016] Christopher Thomas. Opensalicon: An open source implementation of the salicon saliency model. *arXiv preprint arXiv:1606.00110*, 2016.

[Truong *et al.* 2022] Thanh-Dat Truong, Quoc-Huy Bui, Chi Nhan Duong, Han-Seok Seo, Son Lam Phung, Xin Li, and Khoa Luu. Direcformer: A directed attention in transformer approach to robust action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20030–20040, June 2022.

[van Miltenburg *et al.* 2018] Emiel van Miltenburg, Ákos Kádár, Ruud Koolen, and Emiel Krahmer. Didec: The dutch image description and eye-tracking corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3658–3669. Association for Computational Linguistics, 2018.

[Vaswani *et al.* 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

[Vinyals *et al.* 2015a] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. *Advances in neural information processing systems*, 28, 2015.

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

[Williams 1992] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

[Xu *et al.* 2015] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[Yu *et al.* 2017] Youngjae Yu, Jongwook Choi, Yeonhwa Kim, Kyung Yoo, Sang-Hun Lee, and Gunhee Kim. Supervising neural attention models for video captioning by human gaze data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 490–498, 2017.

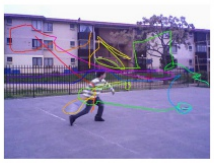
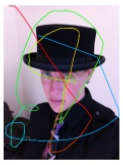
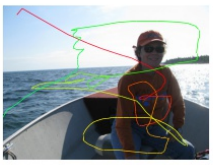
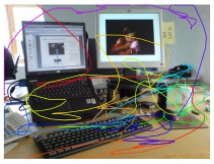
[Zhang *et al.* 2020] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747*, 2020.

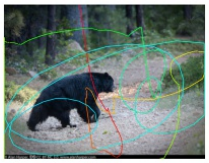
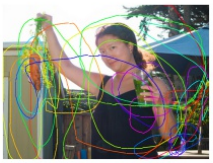
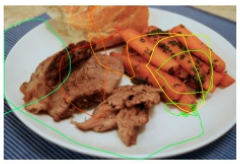
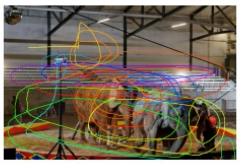
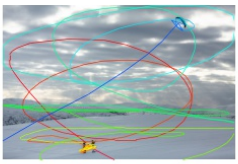
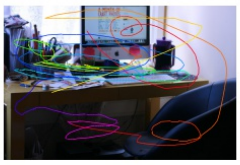
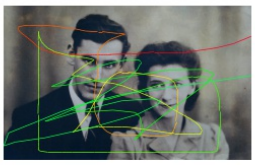
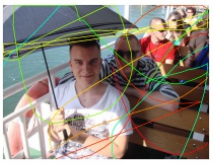
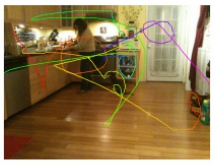
[Zhou *et al.* 2017] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

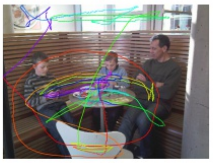
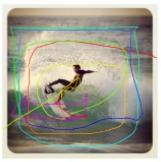
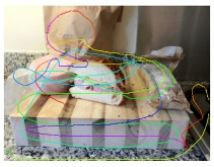
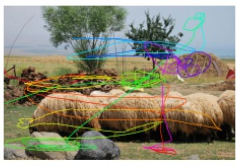
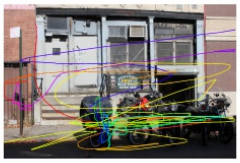
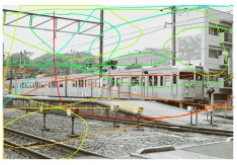
Appendix A

Additional Examples

For the purposes of transparency we include an unbiased, random selection of 60 images taken from our test split. Images are presented overlaid with their associated traces, and with their ground truth captions and predicted captions on the left and right respectively. Predicted captions are those taken from our final Alternating Cross-Modal architecture.

this is the man standing and holding tennis racket. this is the tennis ball. he is playing tennis. i can see another racket placed here. this looks like a iron barre gate. this is the building with windows and doors. these are the stairs. i can see trees and bushes.		this is the man jumping and holding skateboard. he wore t-shirt, trouser and shoes. this is a fence. i think this is the dustbin. this looks like a gate. these are the buildings with windows and doors. this looks like a fencing. i can see trees and a car.	a person is holding a baby and this baby is sleeping and holding a mobile.		a person is sitting on a couch. he is holding a mobile phone in his hands.
in this image i can see ground full of snow and on it i can see number of people are standing. i can also see all of them are standing on skiboards and all of them are holding sticks. i can see everyone are wearing jackets, caps, gloves and glasses. over here i can see few boards and on it i can see something is written.		in this image i can see few people are standing on skiboards. i can see all of them are wearing jackets and few of them are holding sticks. i can see all of them are wearing gloves. in the background i can see few poles, few wires and few more people. here i can see he is holding a stick and wearing a bag and standing.	this picture is clicked in the railway station. on the right there are group of persons standing on the platform. in the center we can see a railway track and a train which is running and on the left corner we can see the buildings. in the background there is a sky, clouds and some buildings and skyscrapers.		this picture is clicked outside the railway station. on the right we can see the group of persons standing on the platform. in the center there is a railway track and a train running on the railway track. on the left there are some persons standing on the platform. in the background we can see the sky and buildings.
there are four dogs standing. one dog is holding a disk with his mouth. this is the grass. this looks like a fencing with green poles. at background these are trees and plants.		there are three dogs running. this is the grass which is green in color. this is the watermark. i can see a fence with the poles. this looks like a iron sheet.	in this image i can see a mouse with grey color and a black color wire is attached. to that wire i can see a key. it is on the white table. on the mouse i can see something is written.		in this image i can see a toilet and the name`` is written on it. to the side of the toilet there is a wooden board. there is a blurred color pole. on the top i can see a colorful light.
this person wore tie, black hat and looking forward. this is black chair. wall is in white color.		this is the man smiling. he wore a hat. in the background, i can see two empty chairs.	this picture describe about the women wearing orange color t-shirt with jeans sitting on the boat. behind you can see clear sea water.		in this image the person is sitting on the boat and holding some board and behind the person some trees are there background is cloudy.
in this picture we can see a man lying on horse which is running on road and aside to this horse we can see a stand with a board, wall, hut, trees, wooden sticks, door.		in this picture we can see a man wore cap, shoes sitting on a ground and beside to this horse we can see a pot with plant in it, wooden sticks, wall, wooden sticks, trees.	i can see in this image a person is doing skating on the wooden table. the person is wearing a cap, black dress and shoes. in the background i can see sky, poles and a group of people are standing on the ground. i can also see a wooden bench on the ground. and in the right side of the image i can see buildings, few people are holding bags and other objects in there hand. i can also see few persons are sitting on the ground.		i can see in this image a man is doing skating. the man is wearing a cap on his head and black colour dress. in the background i can see a sky, on the right side of the image i can see a group of people are sitting on a wooden table and on the right side of the image i can see a tent and group of people are standing on the ground. i can also see there is a tent on the ground and shadows.
in this picture we can see one commode and tissue rolls along with table.		in this image i can see a toilet, tissue roll, tissue roll and a wall.	to the front of the image there is a lady with white dress is standing and holding the tray in her hand with gloves. inside the tray there is a food item with carrot pieces on it. to the right side corner of the image there is a tissue roll and beside the roll there is a cake on the tray with candles on it. behind the lady there is a cupboard with jars, bottles and few other items on it. to the left side of the image there is a blue and red color packet on the table.		this picture describe about the women wearing white color t-shirt and black skirt is sitting on the chair and working on the pizza. in front we can see a pizza with some tomato slices and cheese in the plate. behind we can see a kitchen table top and some wine bottles and a red color box on the top of the table top. on the top we can see a big red color wall with some hanging on it. on the left corner we can see a coffee machine with some tea cups and some other items are placed.
in this picture there is a person wearing a red color t shirt standing on the skating board. there is a floor here. in the background there are some trees and a sky here.		in this image we can see this person wearing red t-shirt is standing on the skateboard and skating on the curved surface. in the background, we can see people, trees and the sky.	this image is taken indoors. at the bottom of the image there is a table with a few things on it and there is an empty chair. in the background there is a wall with a few cupboards and doors. on the left side of the image there are a few cupboards. on the right side of the image there is a cover. in the middle of the image there is a table with a sink, taps, tissue rolls and many things on it.		this image is taken indoors. at the bottom of the image there is a table with a few things on it and there is a bottle. in the background there is a wall with a door, cupboards, shelves and many things on it and there are a few clothes on the table. on the right side of the image there is a bag on the floor. in the middle of the image there is a table with a few things on it. there is a sink.
in this in the middle there is giraffe. on the left there is giraffe, fence. on the right there is wall, stones, plants, trees, hill and sky,.		in the center of the image there is a giraffe. on the left there is a bird. on the right there are stones, trees and sky.	this is the picture of the railway track. in the foreground there is a board. at the back there is a railway track and there is a fence behind the track. in the background there are trees on the mountain. at the top there is a sky.		this is the picture of the city. in the foreground there is a board. at the back there is a road and stones. at the back there are trees and buildings. at the back there is a mountain and mountains. at the top there is a sky.
in the center of the image there is a table. there is a computer, laptop, mouse, glass, books, headset, and a telephone placed on the table.		in the center of the image there is a table and there is a computer, keyboard, mouse, glasses, papers and a laptop placed on the table.	in the picture there is a road, on the road there is a pole, to the pole a dog is tied with the belt, beside the pole there is a person standing, back of the person there is a wall with the frame.		in this picture we can see a dog and a belt is tied to a pole. we can see the road and a person walking on the path. there are poles. we can see some other objects and the background is blurry.
there is a sign board on the left side. it is representing address board. there is a houses on top of the left. there is a wooden board,posters,poles banners on the right side. we can see in the background sky. trees.		in this image i can see a board which is in white and black color. we can see building and glass window. on the board`` stop'' is written. there are trees and a light pole. the sky is in white and blue color.	on the right side of the image there is a man standing and holding a bat in his hand, behind him there is a wicket keeper. in the background we can see people sitting and some of them are standing.		in the center of the image there is a man standing and holding a bat in his hand. on the right there is a man. in the background there are people sitting. we can see mesh. there are bags.

the kid placed his legs on a skate board which is in air and there are few people beside him and there are buildings and trees in the background.		the person wearing black dress is in the air and holding a skateboard and there are few people beside him and there are buildings and trees in the background.	this picture might be taken from forest. in this image, in the middle, we can see an animal walking on the land. on the right side, we can see some plants. on the left side, we can see some trees, plants, at the bottom there is a land with some stones.		this picture might be taken from forest. in this image, in the middle, there is a black color animal. on the right side, we can also see another animal walking. on the left side, there are some trees, plants, stones, at the bottom there is a grass and a land with some stones.
here a big pillar consists of a wall clock in the left it's a traffic signal.		in this picture we can see a clock, and also we can find a tower and clouds.	there is a person standing and holding umbrella. a far we can see water,persons and sky.		there is a woman walking and holding umbrella and we can see water, group of people and sky.
in the picture we can see a car on the path and stand on it we can see a crow, and just beside it we can see a truck and car, and far from it we can also see two cars one is red and white in color, and pole near to the road, we can also see some trees, flowers, to the plants and building and grass.		in the picture we can see a street, in the street we can see a path, on it we can see a bird which is black in color standing and just beside it we can see a car which is red in color and in the background we can see some cars on the road and some plants near the railing and some grass on the path.	in this picture there is a lady who is standing at the center of the image, by holding a root of the carrot in her hand and there is a glass in her other hand, glass contains leaves in it, there is a door at the left side of the image, and there are trees at the right side of the image, there is a table at the right side of the image on which there is a bottle and an umbrella.		in this picture there is a woman at the center of the image, she is holding a glass in her left hand, she is smiling and there is a small girl at the right side of the image, there is a house at the left side of the image behind the lady, there is a tree at the right side of the image, and there is a bench at the right side of the image, on which there is glass.
these are the food items and in the right it's carrot pieces.		in this image there is a food item placed on the white plate.	in the picture there are two elephants walking on the sand there is a small wall near to them,there are two persons standing near to the elephant,there is a stand near to the elephant with the light there is a stadium a person a holding a table,there are many lights and poles in the stadium there is also a fence.		in the picture there is one big elephant and two elephants are present in the middle and one person is standing at the right corner and there are two people present in the middle of the picture and there are iron rods, lights are present on the wall and there are big pipes present on the roof in the picture in the picture.
in this image at the bottom there are two people standing and there is snow, in the background there are snow mountains. and at the top there is sky and at the top we could see a parachute.		in this image there is a person standing on the snow. there is snow on the land. background there are hills. top of the image there are clouds and sky. there is a parachute flying in the air.	it is a food in white color plate.		these are the food items in a plate.
in this image, we can see a laptop, cups, pens, papers, a keyboard, a mouse and there are some other objects on the table and we can see cables. at the bottom, there is a chair and in the background, we can see a curtain and a wall.		in this image, we can see a monitor, keyboard, mouse, cup, pens, pens, pens, pens, pens, pens, pens, pens, pens, cup, books, pens, cup, papers, pens we can see. background, there is a wall. window, curtain, glass and trees we can see.	in this image we can see a couple. the image look like an old photograph.		in this image we can see two persons wearing suit and tie are on table.
in this image we can see this woman is wearing black and holding a cake in her hands. in the background, we can see few more people standing.		in this image we can see a woman holding a cake and a mobile phone in her hands. in the background, we can see people standing near door.	in the foreground we can see rocks stones grass and railway track. in the middle of the picture we can see a train, people and hill. on the right corner we can see people and vehicle. at the top it is the sky.		in this image we can see a track and stones. in the background, we can see a train moving on the railway track with people walking on it, we can see a building and trees. at the top of the image, there is sky.
in the left i can see a wall and a wall painting on it. at the bottom i can see a sofa. in the middle i can see a window and a lamp and a door. through window i can see the sky and trees and mountains. this image is taken in a room.		in the left i can see a window and a door and a cloth on it. in the right i can see a wall on which a wall painting is there and a wall on which a wall painting is there and window and trees are visible. this image is taken in a room.	in this image we can see there are so many people sitting i n a boating ship where many of them are wearing shades and one of the is holding mobile phone and other one is holding umbrella and sitting in front is floating on water beside him there is a tube tied on ship.		in this picture there are group of people sitting and smiling and there is an umbrella in the foreground. at the back there is water. at the left side of the image there is a woman sitting and smiling and she is holding the phone. at the right bottom there is a railing and water.
to the left side bottom there is a platform. behind the platform there is a pole with signals. behind the signals there is a train track on the ground and to the left corner there is a room with the roof. to the right side on the ground there are many train tracks with train on it. and also there trees, poles, boxes and to the top of the image there is a sky.		to the left bottom of the image there is a platform with a pole with a traffic signal light and also there is a train on it. to the left corner of the image there is a building with pillars. to the right side of the image there is a railway track with grass and poles. in the background there are many trees, plants and poles. to the top of the image there is a sky.	in this image we can see a minivan on the road. here we can see few persons sitting inside the minivan. in the background, we can see few cars. here we can see few persons on the sideways of the road. here we can see a pole on the right side.		in this image we can see a bus on the road. here we can see many people sitting inside the bus. here we can see a car on the road. here we can see a person walking on the left side. here we can see a traffic signal pole and building.
in this image i can see a person standing on the floor in front of the kitchen cabinet. backside of her i can see a dog standing on the floor and in the middle i can see the cupboard and there are the bottles kept on the rack and there is a door visible in the middle and there is a bottle kept on the floor in the left side, and there is a micro oven kept on the table in the middle.		in this image i can see a kitchen in the middle i can see a table, on the table i can see a dog and a dog stand on the floor and in the top left i can see a cupboard and in front of the cupboard i can see the wall and a photo frame attached to the wall and in the right side i can see a cupboard and on the cupboard i can see a cat visible and a window visible.	in this image there is a laptop, electronic device, remote , cup and pen stand, picture frames are on the table. on the desk there is a keyboard. there are few racks having books in it. before the table there is a chair. right side there is a window to the wall.		in this picture we can see laptop, monitor, mouse, cup, pen holder, box, pen, pen stand and few other things on the table. we can see few books, pens in a shelf and other things on the table. there is a chair on right side. we can see a wall.

this image is taken in a railway station. on the left side of the image there is a platform with poles, pillars and roof. in the middle of the image a train is moving on the track. on the right side of the image there is another platform with street lights, a board with text, cabin and roof and there is a building with walls, windows and roof. at the top of the image there is a sky.		this image is taken in a railway station. in the left side of the image there is a platform with pillars, street lights and there is a train on the track. there is a train on the track. in the right side of the image there is a platform with pole and a board with text on it. in the background there are many buildings with walls, windows and roofs. at the top of the image there is a sky.	this is an inside view. here i can see one man and two children are sitting on the bench in front of the table on which few glasses, saucers and a sheet are placed. the children are holding mobiles in their hands and looking into that. on the bottom of the image i can see one empty chair. on the top of the image there is a window through that we can see the outside view. on the top corner of the image there is a frame is attached to the wall.		this is an inside view. here i can see a man and a woman are sitting on the chair. in front of them there is a table on which glasses, bowls and some other objects are placed. everyone are holding the glasses in their hands. on the table i can see a white color chair on the floor. on the top of the image i can see the wall on which few glasses are placed. on the top left corner of the image there is a frame is attached to the wall.
in this image a man is seeing surfboarding on the water. in the background there is a ocean.		in this image a man is surfing on the water. in the background there is a water body.	in this image i can see an air craft which is in orange and pink color painted on it, at down there are two wheels, at the back ground i can see the other air craft and few trees in green color and two buildings, at the top the sky is in blue color, at front there are few poles and a board in orange color.		in the image we can see this is a airplane on the runway, this is a number plate and this is a number plate of the airplane. there are even buildings and trees around. there are many trees and buildings in the background and a cloudy sky and this is a cloudy. this is a road on the road there is a cones and a board.
in this image we can see an aircraft which is in white and red color and a woman is standing in front of it. she is wearing a black dress and scarf and she is smiling. in the background, we can see grass, mountains and sky with clouds.		in this image we can see a aircraft which is of red and white color and there is a woman wearing black color dress standing on the ground and smiling and at the background of the image there are some trees, mountains, grass and a clear sky.	there are two boats on the water and a man is at the corner of the boat. standing and in the boat there is a woman sitting and a man sitting and a vehicle. in the background there are trees,hoarding and water.		this image is clicked outside. there is boat in the right side. there is a man standing. in the middle there is a woman and a man, man and woman are sitting in the boat. there are trees in the top and middle.
there is a wooden chair. on that there are two teddy bear. one is white and other is brown. in the back there's a wall. on the left side there is something with orange, green and white color.		there is a chair. on that there is a chair. on that there is a white teddy bear. in the back there's a wall. on the left side there is a table. on that there is a toy.	in this picture we can see a boy standing on grass and smiling and in the background we can see trees, pole, kite, sky with clouds, path.		in this picture we can see a boy standing on grass and flying a kite and in the background we can see trees, sky with clouds, poles.
this picture is an inside view of a washroom. on the left side of the image we can see a lady is standing and smiling and holding a cloth. at the bottom of the image we can see the floor and toilet seat. in the background of the image we can see the wall.		in this image i see a woman who is standing and she is wearing a blue jeans and she is holding a thing which is of white in color and i see the floor which is of brown in color and in the background i see the wall which is of brown in color.	this image is clicked outside. there is a door in the middle. there is a window in the top right corner. there is a person standing in the middle wearing a coat, cap and shoes. she is holding a mobile phone.		this image is clicked outside. there is a pillar in the middle. there is a window in the top right corner. there is a person in the middle, he is wearing brown jacket and shoes. he is also wearing a cap.
in this picture we can see food items, plastic cover, paper bag and these all are placed on a wooden plank and aside to this wooden plank we can see a stove.		in this picture we can see bread slices, bread, bread and some other food items on a plate and this all are placed on a table and here we can see a chair.	in this image there are group of sheeps eating the grass on the ground. at the there are stones. at the top there is sky. in the background there are trees and dry grass beside the trees. on the ground there is grass. there are stones in the background and there is a grill beside it.		in this image there are group of sheep standing on the ground. on the ground there is grass. at the top there is sky. in the background there are trees and there are rocks on the ground. on the ground there is sand and there are small rocks. in the middle there is a wooden fencing.
there is a person in grey color t-shirt, wearing a tie which is in violet color, holding a bottle and drinking. in the background, there are other persons standing, there are lights, ceiling and other objects.		there is a person in blue color shirt, wearing blue color tie and blue color tie, holding a glass with one hand and drinking. in the background, there is a person standing, lights, ceiling and other objects.	in the center of the image we can see a few bikes. and we can see a black cloth on one of the bikes. in the background there is a wall, pole, board with some text and a few other objects.		in the picture there is a road, on the road there are many vehicles present, near to the road there is a black bag, there is a pole, there is a building, there are many poles, there is a clear sky.
in this image we can see there is a train riding on the tracks beside that there is a platform and other tracks also we can see there is are electrical poles, building, mountains and trees.		in this image we can see there is a train on tracks, behind that there is a railway track with stones, fence and electric poles with cables and buildings with trees in front of the building.	in this image i can see a photo frame in which i can see a person and the wall.		in this image i can see the person with tie. i can also see the background is the wall.
the picture is clicked in a forest where we observe a wooden bench on which there are three women and one of them is holding a red basket and in the background we observe a tree.		in this picture we can see a bench, there are three women sitting on the bench, in the background we can see a woman and a kid sitting on the bench, we can see grass here.	on wall there is a white curtain. on this bed we can able to see cloth, box, pillows, bag and a dog.		in this image there is a curtain and a box on which there is a cat and some clothes beside it.
women in middle of the picture wearing white shirt is holding mobile phone in her hands and she is riding bicycle. beside her, we see plants and board and beside that, we see cars and many vehicles moving on the road and on background, we see a building which is purple in color and we even see trees.		in this image i can see a woman wearing white and black color dress is sitting on the bicycle which is on the sidewalk. in the background i can see few cars and few other vehicles on the road, and i can see few buildings, few sign boards, few trees, few traffic signals, a sign board and few buildings.	in the image we can see there is a sign board and hoarding board kept in front. behind there is a building which is made up of red bricks and on the top there is a clear sky.		in the image we can see there is a hoarding which is kept in the middle and beside there is a building and on the wall there is a window and on the top there is a clear sky.