Show and Tell v23

Santnam Bakshi **Rishi Ghia** **Purvansh Jain**

Abstract

In this project we aimed to implement a Multimodal Show and Tell model to caption images using the MSCOCO dataset. Our image captioning system was run using a combination of a CNN model to encode the images and an RNN model to produce the captions. This was then extended to include attention on the side of the image encoding and attention to produce the caption. The images were also augmented to improve the robustness of the the model. The results were then evaluated using the Corpus BLEU-4 score to test the accuracy of the produced captions. We observed that augmenting the images as well as training on multiple captions showed a significant increase in the Corpus BLEU score.

1 Introduction

Show and Tell v23, lives at the junction of the domains of Computer Vision and Natural Language Processing (NLP), it aims to create a system capable of comprehending and articulating the content of images in natural language. This field addresses the intricate challenge of interpreting visual data and converting it into coherent, contextually appropriate sentences. This involves a dual-component approach: a visual encoder and a language decoder.

Formal Problem Definition

Visual Encoder

The visual encoder is integral in analyzing and extracting key features from images. It utilizes sophisticated methods like Convolution Neural Networks (CNNs) or Vision Transformers (ViT) to identify patterns, shapes, colors, and spatial configurations. These visual inputs are transformed into a structured form, ready for further processing.

Language Decoder

The language decoder serves to transform the structured visual inputs into natural language. Using advanced models such as Long Short-Term Memory (LSTM) networks or Flan-T5, it crafts descriptions that accurately reflect the image content. The decoder's role is crucial in ensuring that the text is contextually valid, linguistically smooth, and grammatically sound.

Performance Evaluation

Assessing the effectiveness of an image captioning system is key. Metrics like BLEU, ROUGE, or METEOR are used to evaluate the quality of the captions in terms of relevance, fluency, and human language alignment.

Real-World Applications

1. Enhancing Text-Based Models: This system empowers models to comprehend and interpret visual data, thereby broadening their functional scope.

2. Video Summarization: Extended to videos, it can succinctly summarize content, offering quick and insightful overviews.

3. Improved Searchability: Generating descriptions for images without text annotations enhances the searchability and accessibility of visual data.

4. Autonomous Security Systems: In security contexts, it allows systems to interpret visual data autonomously, improving monitoring and response mechanisms.

Illustrative Example

Imagine the system analyzing an image depicting a bustling urban street. The visual encoder identifies elements like cars, pedestrians, and buildings. These features are encoded into a structured format. The language decoder then generates a caption such as "A bustling city street with cars and pedestrians under clear skies," effectively describing the visual elements and the scene's ambiance. This illustrates the system's ability to provide a comprehensive, context-aware depiction of visual data, as illustrated in Figure 1.

Figure 1: Illustrative Example

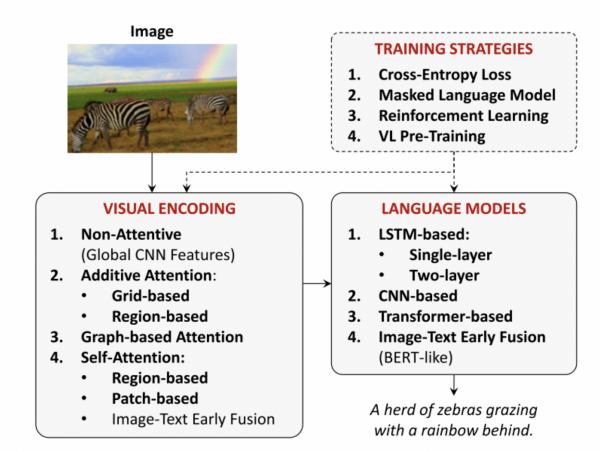

Figure 2: Overview of Multimodal Image Captioning models

In summary, Image Captioning marks a significant advancement in merging Computer Vision and NLP. It opens up vast practical applications and pushes the boundaries of machine understanding and interaction with the visual world.

2 Literature Review

The original caption generating model for automatic image captioning by (Vinyals et al., 2014) integrates CNN and LSTM units. This model encodes visual information into a fixed-length vector using a CNN and then decodes this vector into coherent captions using an LSTM. This model is validated on the MSCOCO dataset, and sets a benchmark in the field by directly converting image data to natural language captions.

(Vinyals et al., 2016) provided an extension of the original Show and Tell paper, and basically makes certain improvements to the original image captioning model. We have included this paper as additional background.

(Xu et al., 2015) explored an improvement to the CNN side of the Show and Tell captioning model by integrating attention mechanisms to describe the content of images. It also innovates on how models are trained, using both deterministic methods through back propagation and stochastic approaches by maximizing a variational lower bound. The model's ability to focus attention on salient image parts during caption generation is highlighted, and its performance was validated on benchmark datasets like Flickr 8k, Flickr30k, and MS COCO. This attention-based approach allows us to visualize the model's "gaze", and provides interpretability to the captioning process.

In (Cui et al., 2022), the authors propose the CLIP-benchmark, a framework to evaluate and analyze Contrastive Language-Image Pre-training (CLIP) models systematically. CLIP learns from image-text pairs to perform tasks such as zero-shot recognition without labeled examples. However, inconsistencies in training recipes and data usage have made fair comparisons between methods challenging. This paper addresses this by assessing CLIP models across three factors: data quality, supervision type, and model architecture and reveals significant insights like the impact of data quality on performance and the differential effects of supervision on ConvNets and Vision transformers. The paper also speaks about CLIP variants like SLIP, DeCLIP, and FILIP, which have improved upon the original by incorporating various supervision signals.

(Stefanini et al., 2021) is an amalgamation of the 'history' of image captioning models, starting from the first Show and Tell paper in 2015. It serves as the backbone of our understanding of this topic, as it documents all the various techniques implemented over the years, how they improved upon the previous best, and how they could be improved further. The paper discusses a range of encoding strategies including attention mechanisms over CNN features and the advent of transformer-based models like BERT, which have significantly improved the relevance and specificity of the captions. It also highlights the shift towards large-scale pre-training and fine-grained attention models, em-

phasizing their success in achieving state-of-the-art performance. It also acknowledges the importance of reinforcement learning and cross-modal pre-training strategies.

2.1 Data

We used the MSCOCO Dataset, which is one of the largest sets of labelled images used for the task of image captioning. It has 118,000 training images, and 5,000 validation images. It is a well researched data-set with relatively clean images, requiring minimal pre-processing.

We defined a Dataset Class to play the role of managing the image IDs, filenames, and captions. It ensures that each element is processed and ready for the model to use. This involves preprocessing steps such as normalization for images and tokenization for captions.

It's important to note that the test data is not utilized due to the absence of captions. Instead, the validation dataset doubles as the test dataset. To avoid overfitting and ensure a fair representation, IID Sampling is employed on the dataset, and the "shuffle" parameter is activated within the data loader, providing samples in a random order each time the model is trained.

The captions are an integral part of the model's understanding and generation process. Hence, they undergo a thorough preprocessing routine. Each caption is modified by adding a start token 'startword' and an end token 'endword'. This modification is done to denote the beginning and end of a sentence, aiding in the model's understanding. Additionally, all captions are converted to lowercase and tokenized using the .split() method for uniformity and ease of processing.

A comprehensive vocabulary is then constructed from these processed captions, encompassing every unique word. This step is crucial as it forms the basis for the model to recognize and generate words effectively.

2.2 Training and Inference

Cross Entropy Loss was used to train our text decoder and the loss calculation was done by comparing the latest generated word to the word that would otherwise be in the ground truth caption.

Inference is performed by the LSTM generating a caption taking into account the start word as the first word, followed by generating each new word with the previous hidden state as the input. In this manner, an entire caption is generated, limited to 10 words.

2.3 Evaluation Metric

The metric used is the BLEU score.

The BLEU score measures the precision of n-grams between the candidate translation and the reference translation, taking into account the maximum n-gram length (usually 4) and the brevity penalty to penalize overly short translations.(Papineni et al., 2002)

The BLEU score is calculated as

$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$

Where:

p_n is the precision of n-grams.

w_n is the weight for each n-gram size (typically equal weights are used).

BP is the brevity penalty to penalize short candidate translations.

N is the maximum order of n-grams.

The BLEU (Bilingual Evaluation Understudy) Score is a widely used metric for evaluating the quality of machine-translated text against high-quality human translations. Its extensive use in research is attributed to its ability to provide a standardized, objective, and quantitative measure of translation quality. BLEU score efficiently correlates with human judgment, particularly at the corpus level, making it a valuable tool for assessing the overall translation quality.

In the realm of image captioning, BLEU is utilized to assess how closely a model's generated captions align with human-generated reference captions. This metric is crucial not only for verifying the factual correctness of the captions but also for ensuring linguistic similarity to human expression.

We have calculated 2 different kinds of BLEU scores, the sentence level BLEU, and the Corpus BLEU score.

When comparing individual caption scoring (sentence-level BLEU) and overall corpus scoring (corpus-level BLEU), there are notable differences. Sentence-level BLEU evaluates each generated caption against its reference, averaging these scores for a final result. This method, while insightful for individual translations, can be sensitive to caption length and may not consistently align with human judgment. On the other hand, corpus-level BLEU calculates the score over the entire dataset. This approach considers all n-grams in the generated corpus, providing a more comprehensive assessment

of the model's performance. Corpus-level BLEU is typically more stable and accurately reflects the system's effectiveness, making it the preferred method for evaluating and comparing the performance of translation or captioning systems.

Finally, a smoothing function has been used to ensure that if any n-gram is not present at all, the entire BLEU score does not drop to 0 (as it is computed using a geometric series of all the n-grams from 1 to 4).

2.4 Simple Baseline

For a simple Baseline, we created a random sequence of tokens from the vocabulary. The total number of words, and count of each words were stored, allowing for the creation of a probability distribution. Using the 'pick word' function, the model picks one word from the vocabulary by "rolling a dice" and through the probability distribution. Thus, words that were more common in the corpus have a higher likelihood of appearing.

The 'simple' function effectively runs the 'pick word' function 10 times over to generate a caption for each image.

The results obtained for our Simple Baseline are shown below

Test Scores:
Maximum BLEU Score (1-1) = 0.0929
Average BLEU Score (1-1) = 0.0334
Corpus BLEU score = 2.122e-155

Train Scores:
Maximum BLEU Score (1-1) = 0.1562
Average BLEU Score (1-1) = 0.0353
Corpus BLEU score = 1.886e-79

3 Experimental Results

Model	Max BLEU (1-1)	Avg BLEU (1-1)	Corpus BLEU
Simple Baseline	0.092	0.0333	2.12E-155
Strong Baseline	0.365	0.064	0.025
Milestone 3	0.759	0.071	0.0347
Milestone 4	0.863	0.128	**0.086**
M4 Grayscale	0.759	0.127	0.081
M4 Horizontal Flip	0.725	0.131	0.084

Figure 3: Results Obtained with 8192 train images, 1024 test images, 10 epochs, 3 captions per image and batch size 32

3.1 Strong Baseline

The strong baseline contains a CNN in order to perform image feature extraction. The model used is a pretrained ResNet50. The final fully connected layer has been removed, which means there is no classification or regression needed, but only a feature representation of the image that can be provided as input to the LSTM.

This LSTM takes the image features and tokenized captions and input and trains on them. Once train is complete, inference can take place.

Test Scores:
Maximum BLEU Score (1-1) = 0.3655
Average BLEU Score (1-1) = 0.0645
Corpus BLEU score = 0.0256

Train Scores:
Maximum BLEU Score (1-1) = 0.8825
Average BLEU Score (1-1) = 0.0984
Corpus BLEU score = 0.0743

3.2 Extension 1

As our first extension, we aimed to upgrade the vision side of our model, while keeping the textual side intact. We attempted to go from a Resnet50 model that produced global features for each image to a transformer based model that incorporated attention over visual regions, in the hope to capture more accurate information from images.

Vision Transformer (ViT)

We tried two different approaches. We started with Google's pertained Vision Transformer, and attempted to transform the feature vectors it produced to a form that was suitable for our LSTM.

This pipeline was where we faced the most challenges. In the case of our strong baseline, we were easily able to produce single feature vectors of size 2048 that were suitable to be taken in by our LSTM. However, when using the Vision transformer in this case our output was a massive matrix of the size 197x768. In order to feed this massive information into our LSTM, we attempted different approaches, like average pooling all the layers, and then transforming it, picking a single layer and transforming it. In both cases we found that we were losing far too much Information, and this lead to very poor captions generated by our model.

SWIN Transformer

In order to combat the challenge of losing too much information and to speed up our computa-

tions. We switched to a SWIN Transformer that produced an output Matrix of the size 49x1024. We first attempted the same approaches of picking the final layer, the first layer and average pooling. All of these approaches yielded very poor results.

We then inserted several linear layers to transform the output of the flattened vector produced from the output of the SWIN Transformer to a vector of size 2048. We observed one promising result when run on a smaller training dataset of 1024 images. We obtained a corpus BLEU score on our validation set of 256 images, of 0.027, which was marginally higher than the Corpus BLEU score of 0.025 obtained in our strong baseline which was trained on 4096 images, and validated on 256 images. On eyeballing the results, they seemed a lot worse than the strong baseline.

Following this, we attempted to raise the size of the training dataset to 20,480 images, and the test dataset to 4096 images. We also increased the number of epochs we were training for, from 10 epochs to 15 epochs. Following this, we actually obtained far more accurate captions. Our Corpus BLEU score went up to about 0.0347 Our main constraint here, in terms of raising the size of the dataset has been the long computation time. We did observe that when it came to using a Transformer based model to ancode the images. The performance was quite poor on a smaller subset of the data, but shot up when we used a larger fraction of the dataset.

3.3 Extension 2

FLAN T5 Implementation Challenges: In our first extension, we focused on enhancing the vision encoder. For the second extension, our aim was to improve the language component. We chose FLAN T5 for its state-of-the-art performance in language modeling tasks. However, integrating FLAN T5 proved to be a complex endeavor. The transformer-based architecture of FLAN T5 posed significant challenges, primarily in training. Achieving meaningful output required an in-depth understanding of T5's literature and architecture. Attempts to streamline the process by freezing FLAN T5 and training the surrounding components were unsuccessful. The weight propagation did not align as expected, leading to subpar results.

Adopting Alternative Strategies: Due to the challenges encountered with FLAN T5, we shifted our focus to other strategies to enhance the accuracy and performance of our model. A significant improvement was training the model on three captions per image instead of one. This approach increased the likelihood of the model finding a low loss with at least one of the captions, thereby enhancing accuracy.

Image Augmentation and Error Analysis: We implemented image augmentation techniques, such as flipping images and converting them to grayscale. An error analysis was conducted on these modified images to understand the impact of these alterations on the model's learning process. This analysis provided insights into the features that most significantly affect the model's performance.

Efficient Storage of Image Embeddings: To improve computational efficiency, we devised a method for storing image embeddings. We utilized JSON and gzipped files to store these embeddings. This approach facilitated efficient uploading to collaborative platforms like Google Colab, enabling us to work with a larger subset of data. This strategy was crucial in mitigating computational constraints and allowing for more extensive training and testing of the model.

In summary, despite the challenges faced with FLAN T5 integration, the adoption of alternative strategies like multiple caption training, image augmentation, and efficient data storage significantly contributed to the overall enhancement of our image captioning model.

Limitations

Developing an image captioning system presented several significant challenges that led to important learnings for us:

Handling Large Image Data

Managing large volumes of image data posed substantial challenges in terms of computational resources, storage, and processing speed. The variability in the data added complexity to feature extraction and representation. Using limited training data reduced our model accuracy, and slower computational resources available for such image-based data reduced the number of different hyperparameters we could test with. We were also constrained by the size of our dataset to run the vision part of our model locally, because we kept hitting the google colab limit of input output operations because of the large size of our dataset. **Encoder-Decoder Transition**

The transition from the visual perception of the

image encoder to the linguistic expression of the language decoder required seamless integration. Ensuring that the visual information was effectively communicated and interpreted by the decoder was a key challenge. Improving this dimensionality reduction between the two could lead to significantly less information loss, leading to the language decoder performing better.

Attention in Vision Encoder

Incorporating attention mechanisms into the vision encoder to focus on salient image features was complex. Fine-tuning these mechanisms to align with the linguistic output added another layer of difficulty.

Fine-Tuning Pre-Trained Transformers

Utilizing pre-trained transformers like FLAN T5 presented challenges in fine-tuning and troubleshooting. Adjusting these models to specific project needs, balancing training data, and preventing overfitting required an in-depth understanding of their architecture.

These challenges highlighted the intricacies of building an effective image captioning system and underscored the need for a balanced approach between handling data, integrating technology, and fine-tuning advanced models.

4 Conclusions

Our project aimed to develop a sophisticated Image Captioning System, blending the domains of Computer Vision and Natural Language Processing. Throughout this journey, we encountered various challenges, particularly in handling large image data sets, transitioning between image encoders and language decoders, integrating attention mechanisms in the vision encoder, and fine-tuning pre-trained transformers.

While our implementations demonstrated significant advancements in automated image captioning, they did not achieve state-of-the-art performance. The key reasons for this gap can be attributed to the limitations in computational resources, challenges in seamlessly integrating complex transformer models like FLAN T5, and the inherent complexities in encoding and decoding multimodal data.

The extensive data requirements for training sophisticated models posed a constraint, limiting our ability to explore a wider range of hyperparameters and model architectures. Additionally, fine-tuning pre-trained models like FLAN T5 required a deep understanding of their architecture, which was a steep learning curve for our team.

Despite these challenges, our project made notable progress in understanding and implementing advanced techniques in image captioning. We successfully incorporated attention mechanisms in vision encoders and explored innovative strategies like image augmentation and efficient storage of image embeddings. These efforts culminated in a system that, while not state-of-the-art, represents a significant step forward in the field of image captioning.

Acknowledgements

We would like to thank our TA Yufei Wang for her deep insights throughout the project. She was quite familiar with the topic we worked on, and had some very good advice throughout the project. We would also like to thank @nalbert9 on github. We took inspiration from his code for the implementation of our strong baseline. We also used the Pytorch and Hugging face documentation.

References

Yufeng Cui, Lichen Zhao, Feng Liang, Yangguang Li, and Jing Shao. 2022. Democratizing contrastive language-image pre-training: A clip benchmark of data, model, and supervision.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. 2021. From show to tell: A survey on image captioning. *CoRR*, abs/2107.06912.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and D. Erhan. 2016. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:652–663.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2014. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *CoRR*, abs/1502.03044.