Exploring Image Captioning with Multiple Neural Network Models

Pasquale Bianco, Mattia Carlino

***Abstract*—This study investigates the performance of various neural network architectures for image captioning using the Flickr30K dataset. We evaluate four models: ResNet combined with RNN, ResNet with an attention mechanism, Vision Transformer (ViT) with RNN, and Microsoft's Generative Image-to-Text (GIT) model. Words are embedded using Global Vector for Word Representation (GloVe), in order to get a quantified representation of them. Performance metrics, specifically BLEU and ROUGE, are employed to assess the quality and relevance of the generated captions. Our findings indicate that models incorporating attention mechanisms and transformer-based architectures outperform traditional CNN–RNN approaches, yielding captions that are more accurate and contextually relevant. This research contributes insights into effective strategies for bridging visual and textual information, highlighting the potential for further advancements in image captioning technologies.**

I. Introduction

RECOGNIZING and describing visual information is a fundamental human ability, essential for daily life and communication. For humans, interpreting an image and describing it with appropriate language requires recognizing key objects, understanding their attributes, and interpreting their relationships in the given context. This process enables a wide range of applications, from explaining visual content to helping people with visual impairments. As artificial intelligence advances, there is increasing interest in enabling machines to perform similar tasks, which can be generalized in multi–model systems gaining information both from an image and text and producing a summarization of it.

Traditional machine learning–based techniques [1], such as feature extraction followed by rule-based language generation, provided foundational methods for this task. However, recent deep learning approaches have demonstrated much greater flexibility and accuracy in capturing complex image–text relationships. Modern deep learning–based techniques, such as convolutional neural networks (CNNs), combined with recurrent neural networks (RNNs), transformers, and attention mechanisms, have made substantial advancements.

A. Models

This research aims to compare the performance of four distinct neural network models: ResNet [2] with RNN, ResNet with an attention mechanism, Vision Transformer (ViT) [3] with RNN, and Microsoft's Generative Image-to-Text (GIT) [4] for image captioning. By systematically analyzing these models, this study seeks to evaluate their strengths, limitations, and overall performance on the task, contributing insight into which approaches may be most effective for accurate and contextually aware image descriptions.

B. Dataset and Evaluation Metrics

We utilize the Flickr30K dataset, which provides diverse images with five human-annotated captions, allowing robust training and evaluation. In addition, we employ BLEU and ROUGE evaluation metrics to measure the quality of the generated captions, enabling a quantitative comparison between the models. Through this analysis, our aim is to identify which neural architectures best handle the challenges of image captioning, bridging the gap between visual and textual information, and advancing the potential applications of image captioning systems.

II. Model Architectures

Model proposed are mainly based on an encoder–decoder architecture involving two main steps:

- image features are extracted through a CNN to encode the image into fixed-length embedding vectors;
- a RNN is typically employed as the decoder to generate a language description.

In the following subsections we will deeply define the four architectures analyzed.

A. ResNet50 and RNN with LSTM

1) Encoder: it uses ResNet50 to extract features from images. It is one of the best CNN models ever developed [2] for this purpose, so it will be the baseline for further studies in the decoder part. The extracted features are then concatenated with words produced time by time by the encoder.

2) Decoder: In our architecture we use a RNN Long Short–Term Memory (LSTM) for their ability to capture long–term dependencies and sequential patterns in text. At each timestamp, the LSTM receives two inputs: the previous hidden state, which memorizes the knowledge of all previous computed data up to now, and the previous word(s) generated, that allows it to generate the next one.

A brief overview is given in Figure 1, where blue blocks are the hidden states.

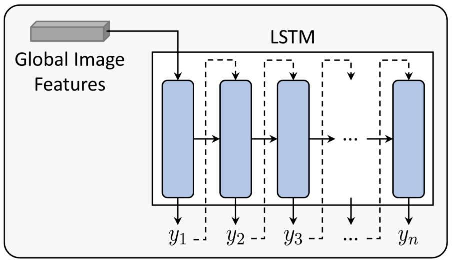

Fig. 1: RNN Decoder with single–layer LSTM

For each element in the input sequence, each layer computes the following parameters, to get to the new hidden state:

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \quad (1)$$

$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \quad (2)$$

$$g_t = \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \quad (3)$$

$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (5)$$

$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

where all matrix W_x are learnable parameters and h_t are the hidden states at time t, and i_t, f_t, g_t, o_t are the input, forget, cell, and output gates, respectively [5].

B. ResNet50 and RNN with LSTM and Attention mechanism

1) Encoder: Refer to Section II-A1.

2) Decoder: In order to improve the previous architecture and obtain better performance, we integrated an Attention Mechanism, which allows the model to dynamically attend to different parts of the image and then use this information to generate meaningful and contextually relevant captions.

Firstly, data from decoder and encoder are gathered through linear layers and activated with ReLU:

$$Attention = ReLU(W_E E + B_E \ + \ W_D D + B_D) \quad (7)$$

Then, this information can be handled to find separation hyperplanes using the softmax function:

$$Context = Softmax(Attention \cdot W_A + B_A) \quad (8)$$

that allows the LSTM to focus on specific parts of the image during each timestamp. An overview of this architecure is given in Figure 2.

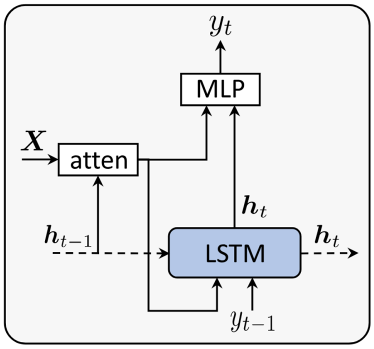

Fig. 2: LSTM with Attention Mechanism

C. Vision Transformer (ViT) and RNN with LSTM

1) Encoder: In this architecture, we used a Vision Transformer [3] as encoder to extract image features. ViT divides the input image into patches tokens (same way as tokens in an NLP application), each of which is passed through a multi-layer transformer, which uses self-attention to capture both local and global relationships between patches. This self–attention mechanism enables the ViT to process long–range dependencies across the entire image, allowing it to capture intricate spatial relationships between objects.

A brief overview of ViT architecture is given in Figure 3.

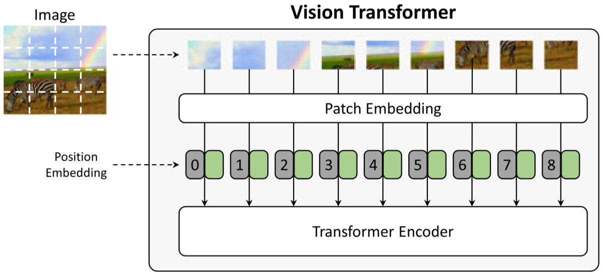

Fig. 3: Vision Transformer architecture

2) Decoder: Refer to Section II-A2

D. Microsoft's Generative Image–To–Text (GIT)

In the last architecture, we use Microsoft's Generative Image–To–Text (GIT) [4] model to generate high–quality captions without requiring an additional CNN encoder. GIT is a pre–trained model that uses CLIP tokens to generate captions, that has been fine–tuned on current Flickr30k dataset. Unlike traditional CNN + RNN pipelines that require explicit feature extraction and sequential text generation, GIT leverages large–scale pre–training on image–text pairs, allowing it to learn robust associations between visual and linguistic data.

III. Global Vectors for Word Representation

Global Vector for Word Representation (GloVe) [6] is a global log–bilinear regression model that combines the advantages of the two major model families in the literature: global matrix factorization and local context window methods. The model efficiently leverages statistical information by training only on the nonzero elements in a word–word co–occurrence matrix, rather than on the entire sparse matrix or on individual context windows in a large corpus.

We use 100-dimensional GloVe embeddings to construct an embedding matrix representing our vocabulary words as fixed-size vectors based on the training set's vocabulary. This resulted in a matrix of shape $[5900, 100]$, where each row corresponds to a unique word encoded in a 100-dimensional vector. This pre–trained word embedding was utilized to initialize the embedding layer in our model, supporting the ResNet50 and ViT encoders by providing semantically rich representations for textual input during caption generation.

For each caption associated with an input image, we represented the text as a fixed-size vector. To achieve this, each word was embedded into a vector using the pre-trained word embedding matrix, and the LSTM was then used to process the sequence of embedded words.

IV. Training and Validation

Training is a critical phase in the development of a robust and effective machine learning model. Each of the architectures presented required a custom training function, tailored to address the unique characteristics of the respective models.

Cross–Entropy Loss was implemented as the loss function for all custom models.

In Table I we propose four parameters over which making some prior evaluation:

- **Epochs**: number of epochs trained;
- **Training time**: total time spent for training;
- **Training Loss**: average training loss in final epoch;
- **Validation Loss**: average validation loss in final epoch.

TABLE I: Training and validation loss per model

Model	**Epochs**	**T. time**	**T. Loss**	**V. Loss**
Rn + RNN	10	43 min	$\sim$ 2.24	$\sim$ 2.95
Rn + RNN & Att	10	117 min	$\sim$ 2.49	$\sim$ 3.25
ViT + RNN	10	150 min	$\sim$ 2.38	$\sim$ 2.91
Microsoft GIT	3	15 hr	$\sim$ 0.05	—

It is important to note that training time varies significantly between models, largely influenced by architectural complexity rather than predictive accuracy. However, this variability does not directly correlate with the effectiveness of each model in minimizing Cross–Entropy Loss. Therefore, this metric could not effectively meet the objective of this model, making it necessary to use new ad hoc comparison ways, such as BLEU and ROUGE, as discussed in Section V-A.

The progression of training and validation losses across epochs is visualized in Figure 4: still, the curves for most models converge closely around similar loss values, except for the GIT model, which achieves a significantly lower loss after just a few steps of the fine–tuning process.

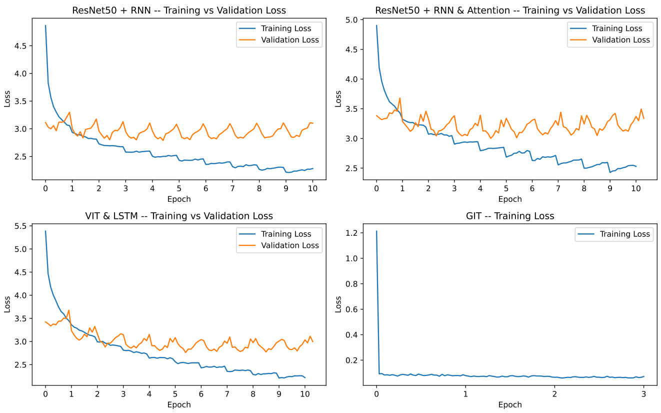

Fig. 4: Training *vs* Validation losses of all models

V. Evaluation

A. Metrics

For evaluating the quality of the generated captions, we employed BLEU (Bilingual Evaluation Understudy) and ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metrics, all presented in this paper [7].

- **BLEU:** it measures the precision of n–grams[2] in *generated vs reference* captions, providing insight into the overlap of short sequences of words:
- **ROUGE:** it focuses on recall, we use the following three variants:
  - *ROUGE–1*: refers to the overlap of 1–grams between the generated and reference captions;
  - *ROUGE–2*: refers to the overlap of 2–grams;
  - *ROUGE–L*: Longest Common Subsequence (LCS) based statistics. It takes into account sentence–level structure similarity and identifies longest co–occurring in sequence n–grams automatically. This is the most preferred metric for image captioning.

Together, these metrics offer a balanced assessment of both the syntactic and semantic quality of the generated captions.

B. Results

Since the trend of the metrics curves is very messy and does not provide any notable information, we propose the percentage of models that perform best in each metric in Table II. As reference, Figure 5 and Figure 6 plots are related to the ROUGE-L metric both for the best performing model (with the cumulative line in the bottom) and their trend.

TABLE II: Best metric distribution for each model

Model	**BLEU**	**ROUGE–1**	**ROUGE–2**	**ROUGE–L**
Rn + RNN	17.26%	15.72%	34.14%	14.75%
Rn + RNN & Att	32.21%	29.41%	24.20%	33.75%
ViT + RNN	12.83%	14.46%	10.13%	13.40%
Microsoft GIT	37.70%	40.41%	31.53%	38.09%

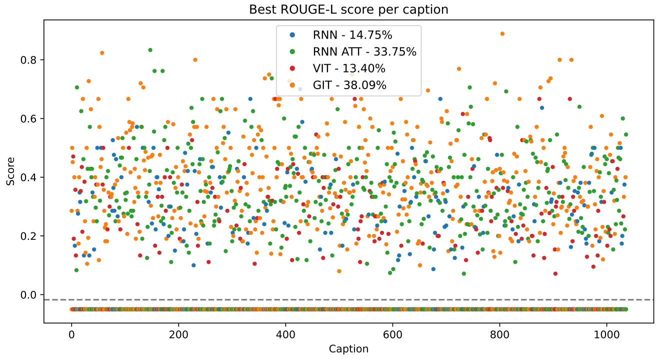

Fig. 5: Best ROUGE-L scores per caption

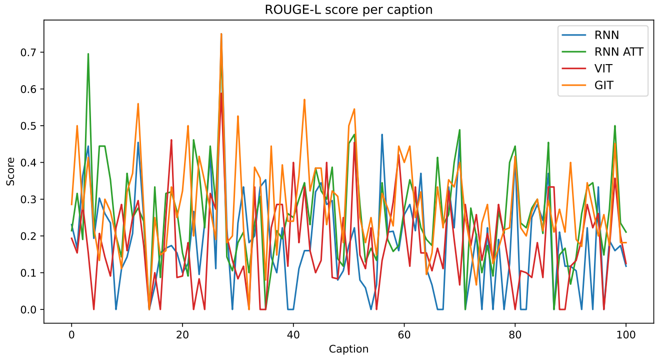

Fig. 6: ROUGE-L plot per caption

[2] An n–gram is a contiguous sequence of n items from a given sample of text; items can be words, characters, tokens, etc.

C. Final considerations

There is no best performing model, each of the proposed ones has its own potentiality and get better results depending on type of features captured in the image, how it generates the caption and the way the reference caption actually matches with image itself. This situation can generally be solved by choosing the best caption among the one that achieves the highest sum of all metrics:

$$m = \arg \max_{m \in \text{Models}} \sum_{i \in \text{Metrics}} \text{Score}(m, i) \tag{9}$$

However, still a human observation could lead to a different result, because of the complexity of the task.

VI. Conclusion and Future Directions

In this study, we evaluated the performance of four distinct neural network architectures: ResNet with RNN, ResNet with RNN and attention mechanism, Vision Transformer (ViT), and Microsoft's Generative Image–to–Text (GIT) in the task of image captioning. Our results indicate that all models are somehow effective, leading to a situation in which none of them is the absolute best performer and all of them are useful to generate a very coherent, descriptive and correct caption.

However, the attention mechanism implemented in the second model led to a significant improvement over the first model, which is identical except for the absence of this layer. By integrating attention, the model can selectively focus on specific parts of the input sequence during each decoding step, allowing it to capture context more effectively and produce more relevant outputs. This is particularly beneficial compared to a standard LSTM-only decoder, where the model is heavily relying on a fixed-length context vector that can dilute important information over longer sequences.

In contrast, a large-scale model such as GIT outperformed the other models in the majority of captions. This result was expected, as GIT was used as a comparative benchmark. However, it is important to note that while GIT achieved high-quality captions, this performance comes at the cost of exponentially longer training times and a significantly more complex forward pass to generate each caption. This highlights that the model's size and computational demands do not necessarily translate proportionally to quality improvements.

A. Future Directions

While deep learning techniques, particularly encoder–decoder models that employ CNNs for feature extraction, have proven to be highly effective for image captioning, further advancements may be achieved by integrating object detection features. This approach could enhance caption accuracy and contextual relevance by allowing the model to focus more precisely on key elements within an image, potentially capturing richer details and relationships among objects.

Furthermore, combining models such as CLIP and GPT–2 offers a promising direction for future research. CLIP (Contrastive Language–Image Pretraining) [8] aligns visual and textual information by encoding images and text in a shared embedding space, effectively capturing semantic relationships and enhancing the model's understanding of image content and context. GPT–2 [9], a transformer-based language model, excels at generating coherent and contextually appropriate text by leveraging extensive language patterns. Using CLIP as the encoder to process and embed the image, followed by GPT-2 as the decoder to generate captions, a combined model could produce captions that are both semantically relevant and lexically rich.

Exploring self-supervised learning methods could also be valuable, especially given the challenges of obtaining high-quality image-caption pairs in specialized fields such as medical imaging. Self-supervised approaches allow models to learn meaningful visual representations from unlabeled data, potentially reducing the dependence on extensive annotated datasets and expanding the applicability of image captioning to domains with limited resources.

References

[1] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt *et al.*, "From captions to visual concepts and back," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1473–1482.

[2] S. Mascarenhas and M. Agarwal, "A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification," in *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)*, vol. 1. IEEE, 2021, pp. 96–99.

[3] S. Cao, G. An, Z. Zheng, and Z. Wang, "Vision-enhanced and consensus-aware transformer for image captioning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 10, pp. 7005–7018, 2022.

[4] J. Wang, Z. Yang, X. Hu, L. Li, K. Lin, Z. Gan, Z. Liu, C. Liu, and L. Wang, "Git: A generative image-to-text transformer for vision and language," 2022. [Online]. Available: https://arxiv.org/abs/2205.14100

[5] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[6] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[7] G. Luo, L. Cheng, C. Jing, C. Zhao, and G. Song, "A thorough review of models, evaluation metrics, and datasets on image captioning," *IET Image Processing*, vol. 16, no. 2, pp. 311–332, 2022.

[8] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021. [Online]. Available: https://arxiv.org/abs/2103.00020

[9] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. W. Kim, S. Kreps, M. McCain, A. Newhouse, J. Blazakis, K. McGuffie, and J. Wang, "Release strategies and the social impacts of language models," 2019. [Online]. Available: https://arxiv.org/abs/1908.09203