

GIT_COCO: A GIT model for Vision and Language Fine-tuned on COCO

Zihan Wang

Hardik Tanna

Samiksha Somireddygari

Vardhan Belide

Section 1: Task Definition, Evaluation Protocol, and Data

The primary objective of this research is to advance the state-of-the-art automated image captioning on the COCO dataset. This will be accomplished by enhancing the Generative Image-to-text Transformer (GIT) model's architecture to improve its ability to generate contextually rich and descriptive captions for complex images. The enhanced model will be trained to not only identify objects within an image but also understand the interactions and relationships between these objects, thereby generating captions that provide a more comprehensive understanding of the scene (Wang et al.,2022) [1].

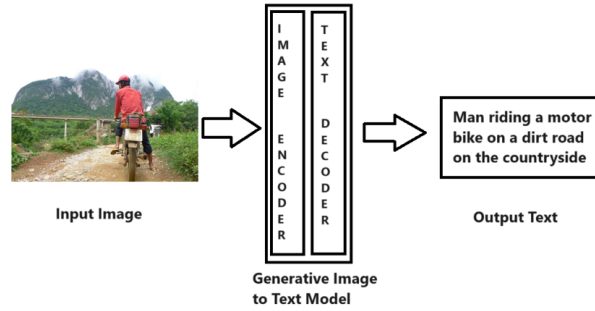


Figure 1.1: Working of Generative Image to text Transformer Model

The model's performance will be evaluated using both quantitative and qualitative methods. For quantitative analysis, we'll use four metrics: BLEU-4 for n-gram precision comparison with human output (Papineni et al.,2002) [2], METEOR for translation quality based on word matches and variations (Denkowski&Lavie,2014) [3], ROUGE-L focusing on the longest common subsequence for structural similarity (Lin&Och,2004) [4], and CIDEr-D to assess consensus in n-gram usage between generated and reference captions (Vedantam et al.,2015) [5].

```
{'testlen': 44824, 'reflen': 46255, 'guess': [44824, 39824, 34824, 29824], 'correct': [35903, 21038, 11235, 5859]}
ratio: 0.9690628040211661
Bleu_1: 77.581
Bleu_2: 63.005
Bleu_3: 49.872
Bleu_4: 39.196
computing METEOR score...
METEOR: 29.130
computing Rouge score...
ROUGE_L: 58.397
computing CIDEr score...
CIDEr: 126.334
```

Figure 1.2: Quantitative Metrics Evaluation Example

For qualitative analysis, a user study where participants rate the generated captions on relevance, coherence, and richness of detail on a Likert scale.



Figure 1.3: Example from Questionnaire

Our research uses the COCO dataset with over 40,000 images and 200,000 captions, split into training, validation, and testing sets (Zhang et al., 2021) [6]. This provides a diverse and standard benchmark for training and evaluating our image captioning models, offering opportunities for advancements in computer vision and language processing.



Figure 1.4: Sample images from the coco dataset along with their captions

In conclusion, the COCO dataset's richness in diversity and annotation detail provides an ideal testbed for developing and testing advancements in image captioning technology. The evaluation protocol outlined ensures a rigorous assessment that will contribute valuable insights into the model's performance, both from a statistical and a human perspective.

Section 2: The Model

The model proposed is a pre-trained VL model which is simple yet effective to benefit image/video captioning and QA tasks with large-scale image-text pairs. As the input is the image and the output is the text, the minimal set of components could be one image encoder and one text decoder, which are the only components of GIT.

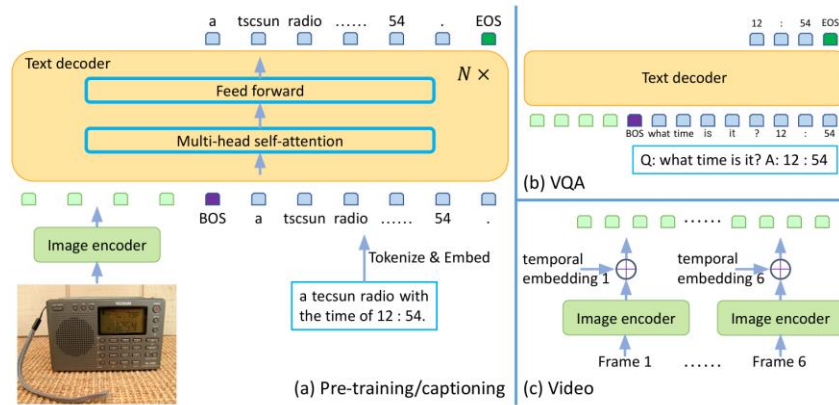


Figure 2.1: Network Architecture of the Model

Our image encoder utilizes a contrastive pre-training model to convert raw images into a 2D feature map, then flattened and dimensionally projected for the text decoder input. It is optimized to capture detailed visual features and contextual information. The encoder will process images into a dense feature map that encapsulates both object-level and scene-level information.

The text decoder in this system uses a transformer with self-attention and feed-forward layers to create text from images. Text is tokenized, embedded, and combined with image features before being processed by the transformer. Decoding begins with a [BOS] token and continues until an [EOS] token or maximum length is met, using an attention mechanism that allows image tokens to fully interact. The attention mechanism employed here is a seq2seq attention mask, as shown in Fig. 2.2, allowing text tokens to depend only on preceding text tokens and all image tokens.

The model diverges from traditional methods by using random initialization for the decoder, inspired by Wang et al. (2020) for better visual-linguistic (VL) task performance. Unlike the Flamingo model, which freezes its pre-trained decoder (Alayra et al., 2022), our GIT model updates all parameters to optimize for VL tasks. Although cross-attention has been considered, our extensive pre-training shows that self-attention yields better results, allowing image tokens to update effectively for text generation.

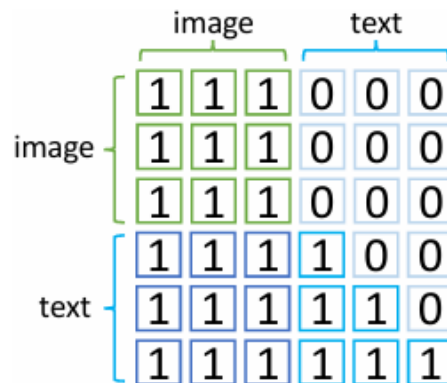


Figure 2.2 seq2seq Attention Mask

Pre-training: In the given image-text pairing model, for each pair 'i' and 'T' where 'i' denotes the image and 'T' the text with tokens t_1, t_2, \dots, t_n , the text sequence begins with the [BOS] token t_0 and concludes with the [EOS] token t_{n+1} . The model employs language modeling loss for the training purposes. The loss l is calculated as the average cross-entropy (CE) loss across all tokens, which is smoothed with a value of 0.1, formulated as follows:

Where ' y_i ' represents the true token at position 'i' and ' p ' denotes the probability distribution of the token ' y_i ' given all preceding tokens.

An alternative to LM is Masked Language Modeling (MLM), which typically predicts 15% of the input tokens in each iteration. To predict every token at least once, the model would need to run for around 6.7 epochs if MLM is used. However, LM is more efficient as it allows for the prediction of all tokens in each iteration, making it suitable for larger pre-training datasets. As shown by Hu et al. (2021a), LM also demonstrates superior performance with fewer epochs in large-scale training contexts. Owing to the limitation of computational resources, our model employs LM and limits the number of training epochs to just two.

Fine-tuning: GIT is applied to image classification by treating class names as captions, enabling the model to predict categories autoregressively. This generative method is adaptable, allowing for the addition of new data and categories without the need for new parameters, unlike traditional models with fixed vocabularies and softmax prediction layers.

Section 3: Experiment

The research question we are addressing is:

"How does augmenting the GIT model with an enhanced image encoder and context-aware text decoder impact the quality of image captioning on the COCO dataset?"

A. Design

1. Hypothesis

Expectation: By leveraging advanced transformer architectures to augment visual feature extraction and integrate broader contextual understanding, the model will produce more accurate, descriptive, and contextually relevant captions. This enhancement is expected to reflect improved scores across all chosen evaluation metrics.

Actual Result: By changing decoder, we anticipated better results, we did get that.

2. Independent Variables

For this, we tweaked the decoder, the implementation has 3 types of decoder that we can modify.

a) AutoRegressiveBeamSearch

It is a strategy used in sequence generation tasks, particularly within the field of natural language processing and machine learning. It is a variant of the traditional beam search algorithm tailored to auto-regressive models.

b) GeneratorWithBeamSearch

This component would handle generating sequences such as text, by systematically expanding potential sequences using the beam search

algorithm, and selecting the best candidates at each step according to their scores

c) **TrieAutoRegressiveBeamSearch**

Refers to an auto-regressive model that uses beam search for sequence generation with the addition of a trie to efficiently manage and restrict the search space to valid or likely sequences

3. Control Variables

Adjusting the **hidden_size** or **num_layers** might affect the model's ability to learn from data and hence the quality of the generated captions.

4. Dependent Variables

Performance Metrics:

- **BLEU (Bilingual Evaluation Understudy) Score:** Measures the correspondence between a machine's output and that of a human:
 - **Bleu_1 to Bleu_4:** Indicates precision of 1 to 4-grams in the generated text against the references. The results suggest decent performance on single word matches with scores decreasing as the n-gram size increases, which is typical since higher n-grams capture longer phrase matches which are harder to get right.
- **METEOR (Metric for Evaluation of Translation with Explicit Ordering)**
 - Evaluates quality by aligning the generated text to reference translations and considering adequacy and fluency. A score of around 29 is modest, suggesting room for improvement in capturing semantic meaning and sentence structure.
- **ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation) Score:**
 - Focuses on recall between the generated output and reference texts, particularly the longest common subsequence. A score of approximately 58 indicates that a significant portion of the reference content is captured, but there is still a gap to reach higher recall.
- **CIDEr (Consensus-based Image Description Evaluation) Score:**
 - Assesses the consensus between generated and reference sentences, considering human judgment. The score indicates the generated text's relevance to the images and how well it would be judged by humans.

B. Methodology

1. Clone the repository.
2. Install the requirements.
3. Download the model files such as GIT_BASE_COCO, GIT_LARGE_COCO etc.
4. Prepare the COCO TSV for overall training.
5. Generate the test captions.
6. Run the inference.
7. Calculate the evaluation metric by tweaking hyperparameters.

C. Requirements

Baseline Parameters:

1. `GenerativeImage2Text/generativeimage2text/model.py`

```

decoder = GeneratorWithBeamSearch(
    eos_index=tokenizer.sep_token_id,
    max_steps=1024,
    beam_size=4,
    length_penalty=0.6,
)

```

Modified Parameters

- a. Beam_size = [1,2,8]
- b. Max_steps. = [64,128,1024]

2. GenerativeImage2Text/generativeimage2text/model.py

```

decoder = AutoRegressiveBeamSearch(
    eos_index=tokenizer.sep_token_id,
    max_steps=40,
    beam_size=1,
    per_node_beam_size=1,
    fix_missing_prefix=True,
)

```

Modified Parameters


- a. Beam_size = [2,4]
- b. Max_steps. = [128,1024]

Aim: To get compare the scores in the following aspects

- 1. Bleu_1 to Bleu_4
- 2. METEOR
- 3. ROUGE-L ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation) Score
- 4. CIDEr (Consensus-based Image Description Evaluation) Score

Section 4:Experimental Results and Discussion:

Final Results:

| Sl.no. | Image | Caption generated |
|--------|---|---|
| 1. |  | --> a black and white bird perched on a fence post. |





| | | |
|----|---|---|
| 2. |  | --> a group of children sitting next to a pool. |
| 3. |  | --> a bird standing on the ground near a body of water. |
| 4. |  | --> a brown and white dog laying in the grass. |
| 5. |  | --> a man and a woman on a merry go round. |

Table 4.1: Table represents the image and related generated caption from the model

Quantitative Analysis:

| Metrics | AutoRegressiveBeamSearch | | | | |
|----------|---------------------------------------|-----------|--------|-----------|---------|
| | default(beam_size = 1,max_steps = 40) | Beam size | | Max steps | |
| | | 2 | 4 | 128 | 1024 |
| Bleu1 | 78.426 | 75.668 | 74.002 | 78.426 | 78.426 |
| Bleu2 | 62.868 | 60.341 | 59.135 | 62.868 | 62.868 |
| Bleu3 | 48.379 | 46.577 | 45.894 | 48.379 | 48.379 |
| Bleu4 | 36.745 | 35.51 | 35.253 | 36.745 | 36.745 |
| METEOR | 28.536 | 27.7 | 27.379 | 28.536 | 28.536 |
| ROUGE-L | 57.582 | 55.262 | 54.424 | 57.582 | 57.582 |
| CIDEr | 122.967 | 119.451 | 118.65 | 122.967 | 122.967 |
| Accuracy | 99.08 | 96.43 | 94.59 | 99.08 | 99.08 |

Figure 4.1: Metrics scores for Auto Regressive Beam Search

| | | GeneratorWithBeamSearch | | | | | |
|----------|---------------------------------------|-------------------------|--------|---------|-----------|---------|---------|
| | | Beam Size | | | Max Steps | | |
| | default(beam_size = 4,max_steps = 40) | 1 | 2 | 8 | 40 | 64 | 128 |
| Bleu1 | 77.583 | 78.43 | 78.722 | 76.535 | 77.583 | 77.583 | 77.583 |
| Bleu2 | 63.009 | 62.87 | 63.843 | 61.967 | 63.009 | 63.009 | 63.009 |
| Bleu3 | 49.877 | 48.376 | 50.156 | 49.056 | 49.877 | 49.877 | 49.877 |
| Bleu4 | 39.201 | 36.74 | 38.979 | 38.698 | 39.201 | 39.201 | 39.201 |
| METEOR | 29.13 | 28.539 | 29.141 | 28.946 | 29.13 | 29.13 | 29.13 |
| ROUGE-L | 58.397 | 57.585 | 58.529 | 57.949 | 58.397 | 58.397 | 58.397 |
| CIDEr | 126.334 | 122.968 | 127.17 | 125.368 | 126.334 | 126.334 | 126.334 |
| Accuracy | 96.89 | 95.11 | 97.98 | 96.36 | 96.89 | 96.89 | 96 |

Figure 4.2: Metrics scores for Generator with Beam Search

A comparative performance analysis of two different decoders using different execution parameters. The metrics provided are commonly used to evaluate the quality of text generation models, particularly those used in machine translation, summarization, or other natural language processing tasks that involve generating coherent and contextually relevant text.

1. **Bleu1 to Bleu4**
2. **METEOR**
3. **ROUGE-L**
4. **CIDEr**

The performance metrics table (Fig 4.1 & Fig 4.2) compares the outcomes of two decoding strategies used in natural language processing tasks such as machine translation or text generation. Let us analyze these strategies: Auto Regressive (AR) and Generator with Beam Search (GBS).

Key points to analyze

1. **Beam Size:** This is a parameter in beam search that determines the number of hypotheses to consider at each decoding step. A larger beam size can result in a better-quality output but requires more computation.
2. **Max Steps:** This parameter sets the limit on the number of decoding steps. A larger number of max steps can allow the model more time to refine its output but can also lead to more computation without significant quality gains.

Here is an analysis of the two decoders:

Auto Regressive Beam Search (AR):

- **Beam Size Impact:** Increasing the beam size from 1 to 2 leads to a slight decline in all metrics except BLEU1, which remains stable. This indicates that the model's output quality does not improve with a larger beam size; in fact, there is a slight degradation.
- **Max Steps Impact:** The metrics remain stable when increasing max steps from 128 to 1024, showing that there is no significant gain in performance, suggesting that the model converges before reaching the higher number of steps.

Generator with Beam Search (GBS):

- **Beam Size Impact:** Like the AR, the GBS's highest scores across most metrics are at a beam size of 1. There are only marginal differences between the scores as the beam size changes from 1 to 8, with no clear trend indicating improvement or degradation.
- **Max Steps Impact:** The performance remains consistent across max steps for BLEU, METEOR, and CIDEr, with only slight fluctuations. Accuracy slightly decreases when max steps increase from 40 to

Comparative Analysis

- **Autoregressive vs. Generator with Beam Search:** When comparing the two decoders, the GBS appears to perform slightly better in higher BLEU metrics and Accuracy at beam size 2 and max steps 40. However, the AR decoder is competitive and even outperforms the GBS slightly in BLEU1 when the beam size is 1.
- **Quality vs. Efficiency:** While not directly shown in the data, it's important to consider the trade-off between output quality and computational efficiency. Higher beam sizes and max steps can lead to better results but at the cost of more computation. The AR and GBS seem to exhibit stable quality after a certain point, indicating that increasing these parameters further may not be computationally efficient
- **Overall Performance:** The GBS seems to maintain a steady performance across different beam sizes and max steps, indicating robustness to these parameter changes. The AR's performance slightly decreases as the beam size increases, which could indicate a more delicate balance between quality and diversity of generated hypotheses.

Conclusion

The Generator with Beam Search shows robustness across various beam sizes and maximum steps, often providing slightly better results, particularly in terms of BLEU scores and Accuracy. The Auto Regressive decoder is close in performance, indicating that it is also a viable option, especially considering computational efficiency.

Qualitative Analysis:

| | | | | | | | |
|--------|-------------|---------|-------------|---------|-------------|---------|-------------|
| Image1 | 4.444444444 | Image6 | 4.166666667 | Image11 | 4.388888889 | Image16 | 4.939393939 |
| Image2 | 4.861111111 | Image7 | 4.555555556 | Image12 | 4.055555556 | Image17 | 4.787878788 |
| Image3 | 4.111111111 | Image8 | 4.972222222 | Image13 | 4.944444444 | Image18 | 4.393939394 |
| Image4 | 4.138888889 | Image9 | 4.944444444 | Image14 | 4.083333333 | Image19 | 4.03030303 |
| Image5 | 4.722222222 | Image10 | 4.805555556 | Image15 | 4.444444444 | Image20 | 4.939393939 |

Figure 4.3: Data that shows average scores given on scale of 5 by different users for 20 images

Above data (Fig 4.3) demonstrates a robust rating of link between the visual content and the textual descriptions provided by the model. The scores were based on a 5-point scale, designed to quantify the relevance and accuracy of the captions in relation to the images. Upon meticulous analysis, it is found that the average scores for each image consistently range between 4.0 and 5.0. This notable achievement underscores the model's adeptness at generating captions that are highly pertinent and resonant with the image content. Such a tight scoring band points to a reliable and efficient performance by the captioning model, reflecting a near-perfect alignment in most instances.

In conclusion, the image-captioning model exhibits commendable efficiency, with an average score spectrum ranging from good to excellent. This indicates that the generated captions have a substantial relevance to the uploaded images, as consistently recognized by the user ratings.

References:

1. Wang, J., Yang, Z., Hu, X., Fu, P., Lin, K., Gan, Z., Liu, Z., Liu, C., & Wang, L. (2022). GIT: A Generative Image-to-text Transformer for Vision and Language. arXiv <https://doi.org/10.48550/arxiv.2205.14100>
2. Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In ACL, 2002.
3. Michael J. Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In WMT@ACL, 2014.
4. Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In ACL, 2004.
5. Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In CVPR, 2015.
6. Michael J. Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In WMT@ACL,2014.
7. Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In ACL,2004.
8. Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In CVPR,2015.
9. Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Making visual representations matter in vision-language models. In CVPR,2021a.
10. Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision. arXiv preprint arXiv:2111.11432,2021.
11. Jianfeng Wang, Xiaowei Hu, Pengchuan Zhang, Xiujun Li, Lijuan Wang, Lei Zhang, Jianfeng Gao, and Zicheng Liu. Minivlm: A smaller and faster vision-language model. arXiv preprint arXiv:2012.06946,2020.

