Image Captioning Using VIT-GPT2 on Flickr8K

Team 7

Atharva Patil (1002138260)

Manan Arora (1002143328)

Harshal Devi (1002172497)

Background

Image captioning is a multidisciplinary task that lies at the intersection of computer vision and natural language processing (NLP). The objective is to develop an automated system capable of generating descriptive and coherent natural language sentences that accurately reflect the content depicted in each image.

Historically, image captioning began with rule-based systems and evolved into statistical models, followed by deep learning approaches. The most recent advancements involve transformer-based architectures, which have revolutionized the field by enabling models to understand and represent complex visual and linguistic patterns simultaneously.

The practical applications of image captioning are vast and impactful. For instance:

- Accessibility tools: Help visually impaired users understand image content via screen readers.
- Image indexing and retrieval: Automatically tag photos for better organization and search.
- Content generation: Enable automatic captioning for social media platforms and news agencies.
- Surveillance: Provide verbal summaries of CCTV footage to improve monitoring efficiency.

This project explores how modern transformer-based models, particularly those combining Vision Transformers (ViTs) with language models like GPT2, perform in the context of image captioning using the Flickr8k dataset.

2. Project Motivation

The motivation behind this project stems from the need to evaluate how well state-of-the-art pre-trained models can generalize image captioning tasks without additional fine-tuning. By combining ViTs, which are highly effective at extracting semantic visual features, with GPT2, a strong language model, we aim to generate high-quality captions for real-world images.

The specific goals of this project include:

- Investigating the synergy between vision and language transformers.
- Assessing the model's capability using both qualitative outputs and quantitative metrics like BLEU and METEOR scores.
- Gaining hands-on experience with deep learning workflows, including dataset preprocessing, model integration, caption generation, and evaluation.
- Building an understanding of the challenges and limitations of current pre-trained captioning models.

Dataset: Flickr8k

The Flickr8k dataset is a popular benchmark for image captioning tasks. It consists of 8,000 real-world images, primarily depicting humans and animals involved in various activities. Each image is annotated with five independent human-written captions, ensuring multiple perspectives on what each image conveys.

Key Features of the Dataset:

- Size: 8,000 images.
- Captions: 5 per image, totaling 40,000 captions.
- Language: Captions are written in simple, grammatically correct English.
- Data Split:
    - Training set: 6,000 images
    - Validation set: 1,000 images
    - Test set: 1,000 images

- Annotation style: Each caption provides a concise description of objects, scenes, and activities.

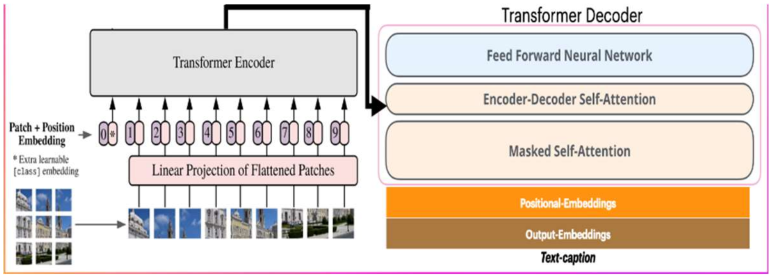
This dataset is ideal for initial experiments due to its manageable size and quality annotations, enabling fast iteration and robust benchmarking of captioning models.

Method

Model Configuration

The project uses the pre-trained ***nlpconnect/vit-gpt2-image-captioning*** model from Hugging Face, which combines a Vision Transformer (ViT) encoder with a GPT2 decoder. To ensure proper decoding and alignment during generation, several configuration steps were applied:

- The ***pad_token_id*** was added to the tokenizer.
- The decoder's token embeddings were resized to match the updated tokenizer vocabulary.
- Key beam search parameters were set for generation:
  - ***num_beams*** = 4 (controls beam width)
  - ***length_penalty*** = 1.0 (prevents overly short or long captions)
  - ***no_repeat_ngram_size*** = 2 (avoids repeated phrases in generated text)


Transformer Decoder
Transformer Encoder
Feed Forward Neural Network
Encoder-Decoder Self-Attention
Masked Self-Attention
Patch + Position Embedding
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches
Positional-Embeddings
Output-Embeddings
Text-caption

Caption Generation Pipeline

A custom function ***generate_caption(path)*** was implemented for caption generation. The steps involved are:

- Images were loaded using the Python Imaging Library (PIL).
- Visual features were extracted using the ViTFeatureExtractor.

- These features were then fed to VisionEncoderDecoderModel to generate output token IDs.
- The generated token sequence was decoded into a human-readable caption using the GPT2 tokenizer.
- GPU acceleration was enabled using ***torch.device***, and the model automatically uses CUDA if available, otherwise falls back to CPU.

Dataset Handling and Preprocessing

The Flickr8k dataset was handled as follows:

- The image and caption ZIP files (***Flickr8k_Dataset.zip*** and ***Flickr8k_text.zip***) were downloaded and extracted using Python's ***zipfile*** and ***os*** libraries.
- Image-caption associations were parsed from Flickr8k.token.txt, where each line contained an image filename and one of its five captions.
- A dictionary structure was used to map each image filename to a list of its five reference captions, facilitating evaluation.

Evaluation Metrics

A comprehensive evaluation was conducted using both corpus-level and image-level metrics:

- **BLEU-1 to BLEU-4 scores** were calculated using ***nltk.translate.corpus_bleu*** with smoothing to account for short references.
- The **METEOR score** was computed using ***nltk.translate.meteor_score*** which accounts for synonymy and word order.
- Sentence-level BLEU-4 scores were also calculated and used for visual analysis.
- Evaluation included additional analyses like:
  - Histogram of BLEU-4 scores to understand score distribution.
  - Caption length histogram analyzes typical output lengths.
  - Word frequency analysis across all predicted captions.

Visualization Techniques

Multiple visualization techniques were used to better interpret the model's performance:

- **Histograms** of BLEU-4 scores across the test dataset provided insight into performance spread.
- **CDF plots** (Cumulative Distribution Functions) of BLEU-4 scores showed how many captions exceeded certain thresholds.
- **Box plots** offered a visual summary of central tendency and outliers in BLEU-4 scores.
- A **scatter plot** comparing caption lengths to BLEU-4 scores revealed a weak correlation, indicating that shorter captions often achieved higher BLEU scores.

Results Visualization

Further analysis focused on qualitative outputs:

- **Side-by-side comparisons** of predicted captions against the five ground-truth captions were shown for several test images.
- The **top 20 most frequently predicted words** were visualized in a bar chart to understand vocabulary patterns.
- Analysis of **caption length vs BLEU-4** scores, backed by a fitted linear regression line, indicated that the most concise captions often scored better, especially on simple images.

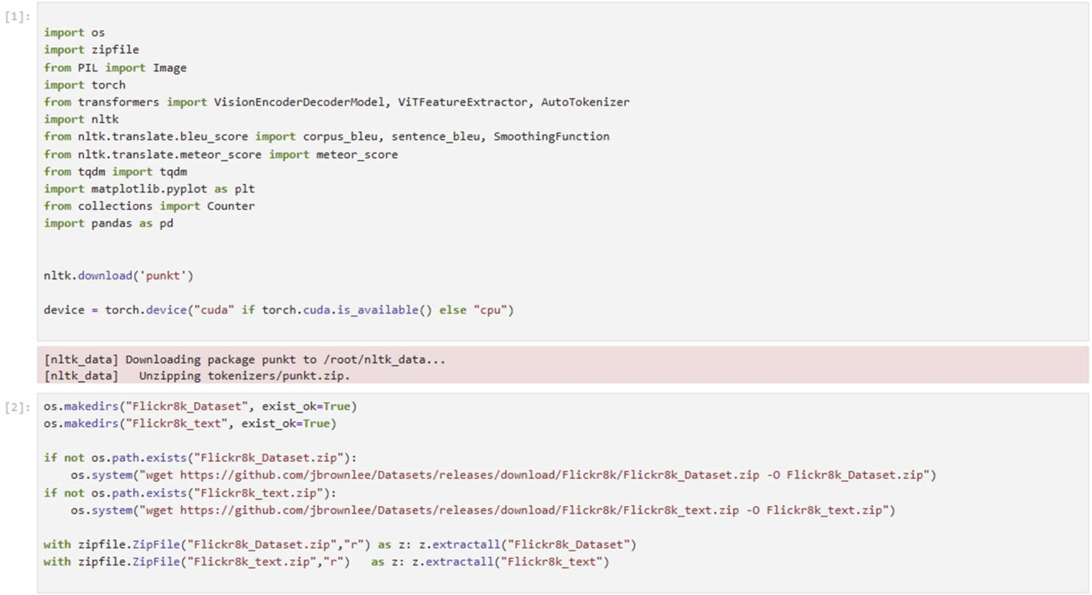
Code Overview

The project followed a systematic implementation structure:

- **Step 1: Dataset Preparation**: The Flickr8k dataset was downloaded, extracted, and organized into appropriate directories.
- **Step 2: Image Preprocessing**: Each image was transformed as required by the ViT encoder, including resizing and normalization.
- **Step 3: Caption Association**: Captions were parsed and grouped by image ID, allowing efficient evaluation using reference-target pairs.

Code Snippets

- Downloaded and extracted Flickr8k dataset and captions.



[1]:

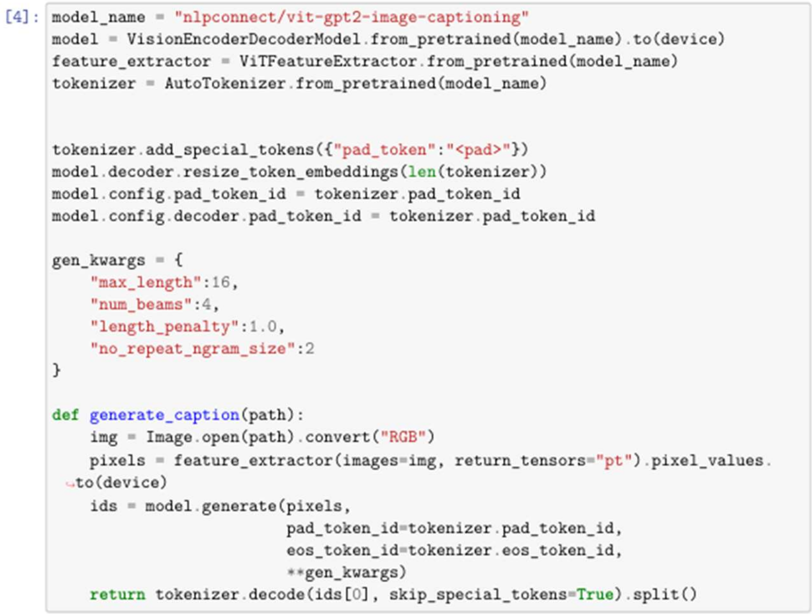
```
import os
import zipfile
from PIL import Image
import torch
from transformers import VisionEncoderDecoderModel, ViTFeatureExtractor, AutoTokenizer
import nltk
from nltk.translate.bleu_score import corpus_bleu, sentence_bleu, SmoothingFunction
from nltk.translate.meteor_score import meteor_score
from tqdm import tqdm
import matplotlib.pyplot as plt
from collections import Counter
import pandas as pd


nltk.download('punkt')

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

[2]:

```
os.makedirs("Flickr8k_Dataset", exist_ok=True)
os.makedirs("Flickr8k_text", exist_ok=True)

if not os.path.exists("Flickr8k_Dataset.zip"):
    os.system("wget https://github.com/jbrownlee/Datasets/releases/download/Flickr8k/Flickr8k_Dataset.zip -O Flickr8k_Dataset.zip")
if not os.path.exists("Flickr8k_text.zip"):
    os.system("wget https://github.com/jbrownlee/Datasets/releases/download/Flickr8k/Flickr8k_text.zip -O Flickr8k_text.zip")

with zipfile.ZipFile("Flickr8k_Dataset.zip","r") as z: z.extractall("Flickr8k_Dataset")
with zipfile.ZipFile("Flickr8k_text.zip","r")   as z: z.extractall("Flickr8k_text")
```

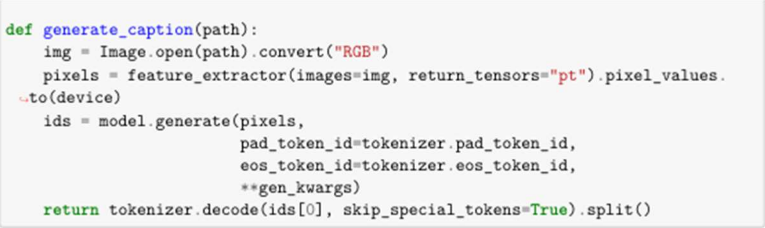
- Loaded pre-trained VisionEncoderDecoderModel (ViT + GPT2)



[4]:

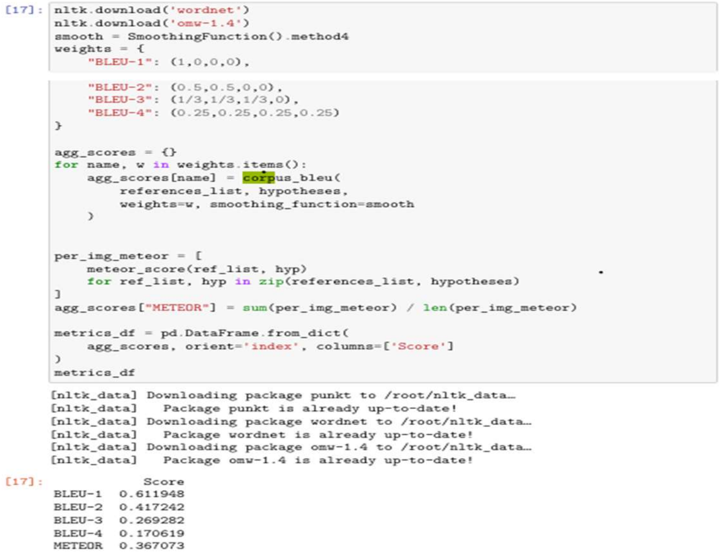
```
model_name = "nlpconnect/vit-gpt2-image-captioning"
model = VisionEncoderDecoderModel.from_pretrained(model_name).to(device)
feature_extractor = ViTFeatureExtractor.from_pretrained(model_name)
tokenizer = AutoTokenizer.from_pretrained(model_name)


tokenizer.add_special_tokens({"pad_token":"<pad>"})
model.decoder.resize_token_embeddings(len(tokenizer))
model.config.pad_token_id = tokenizer.pad_token_id
model.config.decoder.pad_token_id = tokenizer.pad_token_id

gen_kwargs = {
    "max_length":16,
    "num_beams":4,
    "length_penalty":1.0,
    "no_repeat_ngram_size":2
}

def generate_caption(path):
    img = Image.open(path).convert("RGB")
    pixels = feature_extractor(images=img, return_tensors="pt").pixel_values.
  ↪to(device)
    ids = model.generate(pixels,
                         pad_token_id=tokenizer.pad_token_id,
                         eos_token_id=tokenizer.eos_token_id,
                         **gen_kwargs)
    return tokenizer.decode(ids[0], skip_special_tokens=True).split()
```

- Created a function $\boldsymbol{generate\_caption}(\boldsymbol{path})$ to predict captions



```
def generate_caption(path):
    img = Image.open(path).convert("RGB")
    pixels = feature_extractor(images=img, return_tensors="pt").pixel_values.
 ↪to(device)
    ids = model.generate(pixels,
                         pad_token_id=tokenizer.pad_token_id,
                         eos_token_id=tokenizer.eos_token_id,
                         **gen_kwargs)
    return tokenizer.decode(ids[0], skip_special_tokens=True).split()
```

- Evaluated model performance using BLEU-1 to BLEU-4 scores.

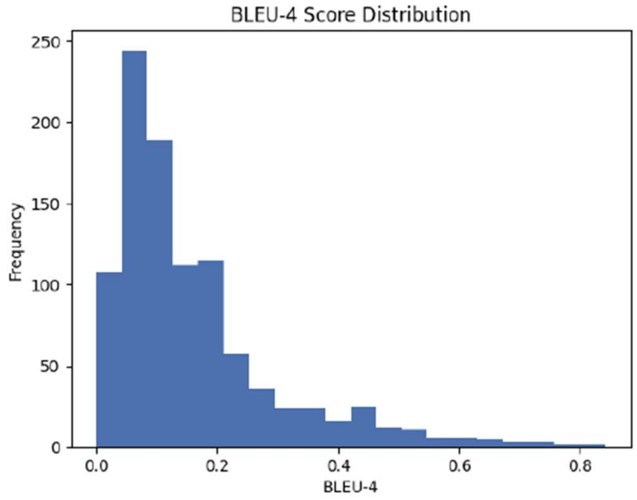


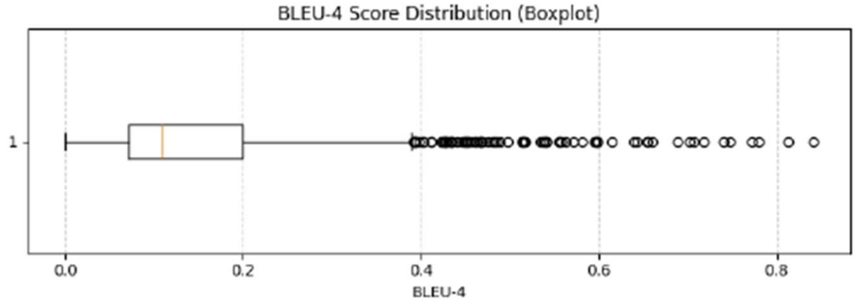
```
[17]: nltk.download('wordnet')
      nltk.download('omw-1.4')
      smooth = SmoothingFunction().method4
      weights = {
          "BLEU-1": (1,0,0,0),

          "BLEU-2": (0.5,0.5,0,0),
          "BLEU-3": (1/3,1/3,1/3,0),
          "BLEU-4": (0.25,0.25,0.25,0.25)
      }

      agg_scores = {}
      for name, w in weights.items():
          agg_scores[name] = corpus_bleu(
              references_list, hypotheses,
              weights=w, smoothing_function=smooth
          )


      per_img_meteor = [
          meteor_score(ref_list, hyp)
          for ref_list, hyp in zip(references_list, hypotheses)
      ]
      agg_scores["METEOR"] = sum(per_img_meteor) / len(per_img_meteor)

      metrics_df = pd.DataFrame.from_dict(
          agg_scores, orient='index', columns=['Score']
      )
      metrics_df
      [nltk_data] Downloading package punkt to /root/nltk_data…
      [nltk_data]   Package punkt is already up-to-date!
      [nltk_data] Downloading package wordnet to /root/nltk_data…
      [nltk_data]   Package wordnet is already up-to-date!
      [nltk_data] Downloading package omw-1.4 to /root/nltk_data…
      [nltk_data]   Package omw-1.4 is already up-to-date!

[17]:            Score
      BLEU-1  0.611948
      BLEU-2  0.417242
      BLEU-3  0.269282
      BLEU-4  0.170619
      METEOR  0.367073
```

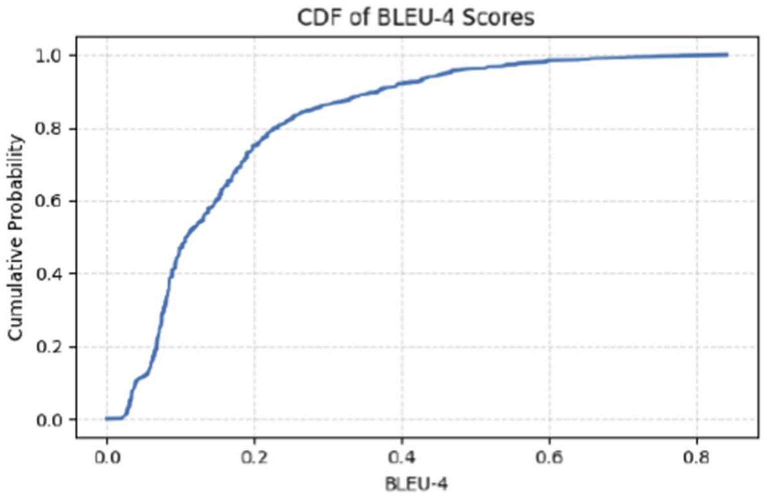
Experiments and Results

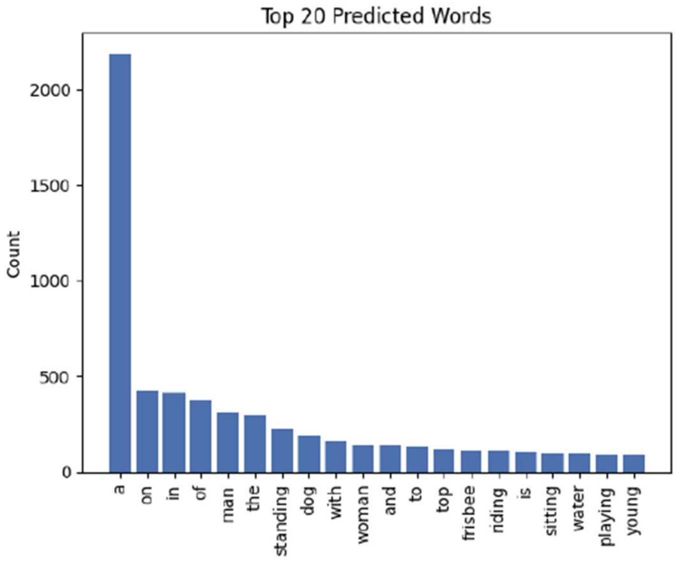
The model was evaluated on the Flickr8k test set. The following BLEU scores were achieved:

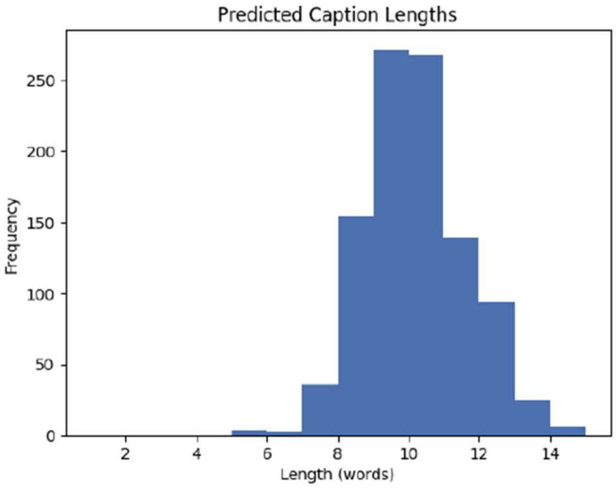
Metric	Score
BLEU-1	0.6119
BLEU-2	0.4172
BLEU-3	0.2693
BLEU-4	0.1706
METEOR	0.3670


BLEU-4 Score Distribution
Frequency
250
200
150
100
50
0
0.0
0.2
0.4
0.6
0.8
BLEU-4

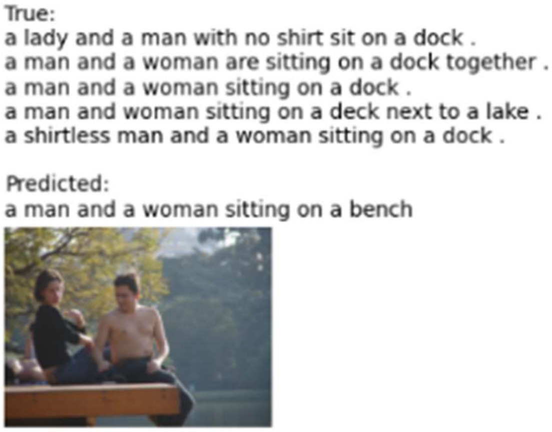

BLEU-4 Score Distribution (Boxplot)
1
0.0
0.2
0.4
0.6
0.8
BLEU-4


CDF of BLEU-4 Scores
Cumulative Probability
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
BLEU-4



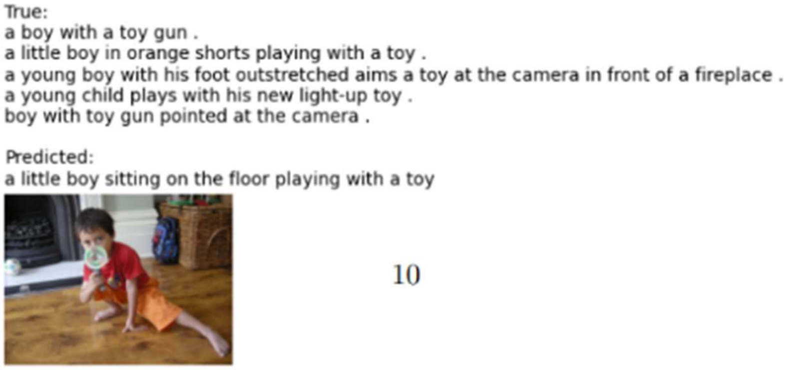


Sample Image captions generated:



True:
a lady and a man with no shirt sit on a dock .
a man and a woman are sitting on a dock together .
a man and a woman sitting on a dock .
a man and woman sitting on a deck next to a lake .
a shirtless man and a woman sitting on a dock .

Predicted:
a man and a woman sitting on a bench



```
True:
a boy with a toy gun .
a little boy in orange shorts playing with a toy .
a young boy with his foot outstretched aims a toy at the camera in front of a fireplace .
a young child plays with his new light-up toy .
boy with toy gun pointed at the camera .

Predicted:
a little boy sitting on the floor playing with a toy
```

Observations:

- The model successfully captured primary objects and actions in most images.
- Captions were generally fluent, grammatically correct, and contextually accurate.
- However, the descriptive richness and diversity of expressions were sometimes lacking, particularly for complex scenes.
- Visualizations showed good alignment between predicted captions and ground-truth references, validating the model's effectiveness despite being used without fine-tuning.

Discussion

The results obtained from the pre-trained ViT-GPT2 model on the Flickr8k dataset demonstrate promising capabilities in generating coherent and relevant captions for images. The BLEU and METEOR scores reflect that the model effectively captures key visual elements such as objects and primary actions, especially in simpler scenes. However, these scores also reveal limitations when the model is confronted with complex compositions involving multiple entities or ambiguous contexts.

A notable observation is that shorter captions tended to yield higher BLEU-4 scores. This suggests that while the model generates grammatically fluent descriptions, it may rely on generic phrasing to maximize n-gram overlaps with references. Visualizations such as histograms and scatter plots supported these patterns and emphasized the lack of diversity in the output vocabulary.

The absence of fine-tuning likely contributed to some of the observed weaknesses, such as reduced specificity and difficulty handling abstract scenes. Nonetheless, the ability to achieve reasonable results with zero-shot inference highlights the strength of transformer-based models and their applicability in rapid prototyping and deployment scenarios.

Going forward, fine-tuning the model on larger and more diverse datasets, incorporating advanced decoding strategies, and integrating multimodal representations could significantly improve the richness and accuracy of generated captions.

Strengths and Weaknesses

Strengths:

- Captions show strong object and action recognition.
- Language generation is fluent and human-like, thanks to GPT2.
- The pre-trained model offers solid results without the need for retraining, making it easy to deploy.

Weaknesses:

- Captions sometimes lack detail and are overly general.
- Limited vocabulary leads to repetitive phrasing.
- Complex scenes with multiple objects or activities are often simplified in captions.

Future Work

To enhance the performance and generalizability of the model, we propose the following future directions:

1. Fine-tuning:
    - Adapt the model on the Flickr8k or other larger datasets (e.g., Flickr30k, MS COCO) to improve specificity and robustness.
2. Advanced Decoding Strategies:
    - Implement top-k sampling and nucleus sampling to encourage more diverse outputs and reduce repetitiveness.
3. Attention Visualization:

   - Introduce visual attention maps to better understand which image regions contribute to specific parts of the caption.

4. Multimodal Pretraining:

   - Incorporate embeddings from CLIP (Contrastive Language–Image Pretraining) to provide richer contextual understanding from both modalities.

5. Human Evaluation:

   - Supplement automatic metrics with human-based evaluations for grammaticality, accuracy, and relevance to ensure real-world utility.

References

- Flickr8k Dataset: https://github.com/goodwillyoga/Flickr8k_dataset?tab=readme-ov-file
- Model (ViT-GPT2): https://huggingface.co/nlpconnect/vit-gpt2-image-captioning
- BLEU Score Evaluation: NLTK Documentation https://www.nltk.org/