Multilingual & Color-Focused Image Captioning For Visually Impaired Using Deep Learning Techniques

Poojan Gagrani

Department of Applied Data Science, San José State University

DATA 270: Data Analytics Processes

Dr. Eduardo Chan

December 6, 2023

Introduction

Image segmentation and annotation have always been intriguing topics in computer vision and natural language processing (NLP), Several deep learning algorithms capable of image segmentation and captioning are reviewed to understand their capabilities and limitations (Minaee et al. 2020). In an attempt to bridge the perceptual gap that visually impaired people experience, this study presents a novel approach that leverages existing deep learning technologies. Firstly, the objects and context in the images are identified, and then color-focused captions are generated to help the visually impaired understand the content. Lastly, the captions are translated into multiple languages enabling wider audience understanding.

The goal of the study proposed to is generate multilingual color-focused captions utilizing the CLIP (Contrastive Language–Image Pretraining) (Radford et al. 2021) model. The study proposes to integrate the advanced image analysis capabilities of CLIP with GPT-2's (Generative Pre-Trained Transformers) (Brown et al. 2020) sophisticated language generation. The multimodal adaption allows the model to not only identify the objects and colors in the image but also articulate these elements meticulously. The dataset used for the study was gathered from MS COCO (Microsoft Common Objects In Context) (Lin et al. 2014) and Flickr30k (Young et al. 2014) which comprises over 330,000 images and 1,650,000 related annotations. A subset comprising 1,392 images and 6,6960 annotations was used, In the data preprocessing phase duplicated images and annotations were removed first, Then images were normalized, by adjusting the pixel values between [0,1], and finally, dimensionality reduction was done using Support Vector Machine (SVM). Finally, the data was split into train, and validation sets with 80%, 10%, and 10% ratios respectively, and was subjected to model training.

Model Development

Model Proposals

Images are unstructured data composed of two-dimensional vector grids which often contain a wealth of information that conveys a tapestry of emotions and meanings. While humans can effortlessly interpret these subtleties, computers rely on sophisticated methods to decipher them. In the realm of computer vision and NLP, researchers have developed an array of complex algorithms inspired by human brain anatomy which can imitate it, these algorithms strive to unravel the intricate details embedded in images, enabling computers to comprehend and respond to text in a human-like manner. While the typical image captioning tasks used to rely heavily on Convolutional neural networks (CNNs), the study done by Dosovitskiy et al. (2020) demonstrated that dependence on CNNs is not necessary and image classification can be effectively done by using transformers only, the results yielded were excellent when compared to the state-of-the-art (SOTA) convolutional networks. Some of the existing research and literature survey for this study exploring the similar architecture is presented below.

Yu (2019) presented a novel Multimodal Transformer (MT) model for captioning images that is considered groundbreaking. This novel model improves image-to-text translation by creatively repurposing the well-known Transformer from machine translation. The integration of multi-view visual representation, which improves the model's comprehension of images, is its main innovation. MT model uses guided attention and self-attention mechanisms to capture complex modal interactions, which are necessary for producing accurate captions. The model outperforms existing methods and ranks first on the leaderboard on the MSCOCO image captioning dataset, validating its superiority and demonstrating its efficacy in combining deep visual understanding with logical caption generation.

Vaswani et al. (2017) proposed a new architecture called Transformer which is solely based on the attetion mechanism. The authors contrast traditional sequential models that convolutional neural networks use and demonstrate their superiority in terms of translation quality, parallelizability, and training time. The Transformer model consists of multi-head self-attention and point-wise, fully connected layers in both the encoder and decoder stacks which makes them more efficient for parallelization and eliminates the need for recurrence or convolution. Transformer outperforms existing models in terms of translation quality. On the WMT 2014 English-to-German translation task, the Transformer achieved a BLEU score of 28.4, surpassing the best existing models by over 2 BLEU. On the WMT 2014 English-to-French translation task, the Transformer achieved a new state-of-the-art BLEU score of 41.8. Overall, the paper shows that attention mechanisms can be used as the core building block of a sequence transduction model, yielding superior performance in terms of quality, parallelizability, and training time. The Transformer model presents a new approach to sequence modeling that eliminates the limitations of recurrent and convolutional networks.

Image segmentation, according to Minaee et al. (2020), is essential to image processing and computer vision for a variety of uses. Image segmentation tasks were performed exceptionally well by deep learning models. Image segmentation has been accomplished using a variety of deep learning architectures and methods, including fully convolutional networks, encoder-decoder models, and recurrent neural networks. Training and assessing segmentation models is commonly done using popular image segmentation datasets like Cityscapes, MS

COCO, and PASCAL VOC. The survey explores potential directions for future research in this area and offers insights into the shortcomings and effectiveness of deep learning-based image segmentation models.

Radford et al. (2021) presented an innovative approach to computer vision through the CLIP model, which leverages natural language for training rather than relying on traditional fixed datasets. The model, pre-trained on a massive collection of 400 million internet-sourced (image, text) pairs, excels in understanding and predicting correct pairings. This training strategy enables CLIP to adeptly transfer its learning to various tasks without specific dataset training, a significant deviation from conventional methods. The paper's extensive evaluation across 30 diverse computer vision datasets reveals CLIP's remarkable capabilities, particularly in a zero-shot setting where it competes with, and sometimes surpasses, fully supervised models. This groundbreaking approach not only demonstrates the efficacy of using natural language as a supervision tool in computer vision but also paves the way for more versatile and generalizable visual models.

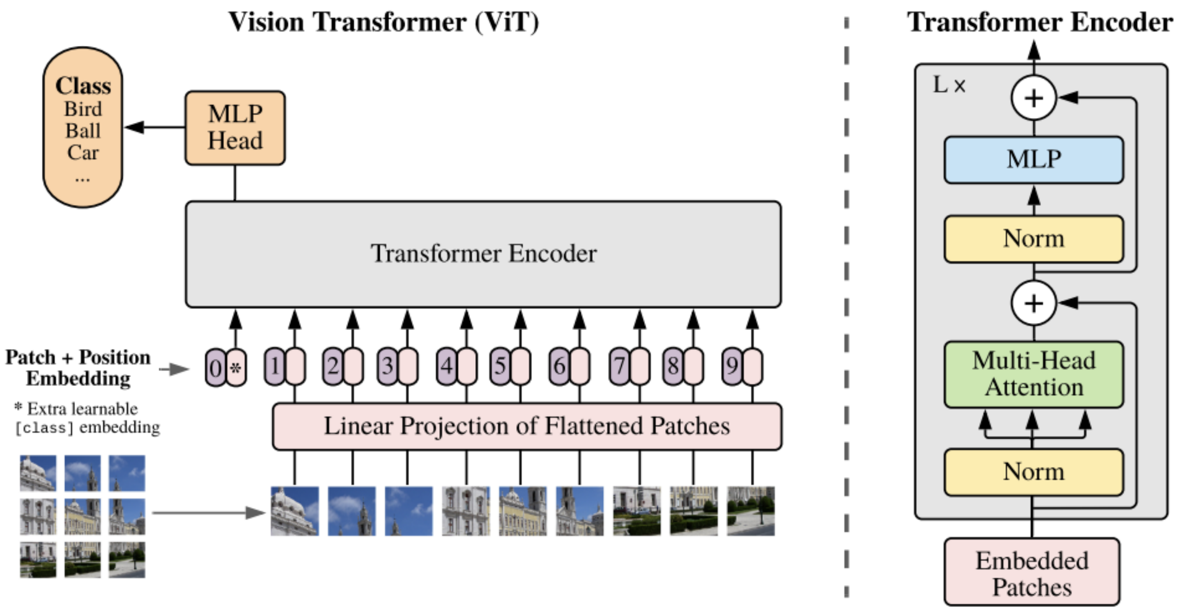
The purpose of this research is to suggest using two of these cutting-edge algorithms, CLIP and GPT-2, which are effective at extracting features from the images and producing color-focused captions that accurately capture the context. In this study, we leverage the innovative architecture of CLIP and GPT-2 both of which are grounded in the transformative paradigm of transformers models. CLIP adopts a multimodal approach that utilizes a transformer-based either a Vision Transformer (ViT) or a variant of the Residual Networks (ResNet) architecture encoder with a vision encoder for image processing, adeptly linking text and visuals, crucial for the generation of descriptive captions. On the contrary, GPT-2 is a language-focused architecture that focuses only on the transformer decoder and does a great job producing coherent text. The multimodal strategy of CLIP and the linguistic capability of GPT-2 work together to improve assistive technology for the visually impaired.

Contrastive Language Image Pretraining (CLIP)

Contrastive Language Image Pretraining (CLIP) implementation utilized for the study specifically utilizes a ViT-based transformer encoder architecture for image encoding, illustrated in Figure 1. A Vision transformer uses self-attention mechanisms introduced by Vaswani et al. (2017) to capture both local features and the overall context of an image by treating it as a series of smaller, fixed-size patches, ViT consists of 3 model variants and are depicted in Figure 2, the base variant has been used for this project. According to Dosovitskiy et al. (2020), an image is first divided into fixed-size patches using patch processing, then each patch is flattened into a single-dimensional vector and a trainable matrix E is used to transform the image patch linearly resulting in a new vector referred to as patch embedding. Alongside the patch embeddings, a learnable classification token is also added to the sequence, and a positional encoding is also added for each patch embedding. The resultant vector is then fed into the transformer encoder which consists of a softmax function that accepts classification tokens to classify the image and comprises multiple layers, each layer containing a self-attention and a feed-forward neural network. The output is normalized at each layer and is subjected to a feed-forward network. The output represents a comprehensive embedding of the image and can be used for tasks like object recognition, classification, etc. The formula used to calculate the patch processing, the linear embedding of patches, positional encoding, and the softmax function are shown in Table 1.

Figure 1

Architecture of ViT



Note. The figure depicts the detailed architecture of a Vision Transformer. From "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." by Dosovitskiy, A. (2020), arXiv.org (Cornell University).

Table 1

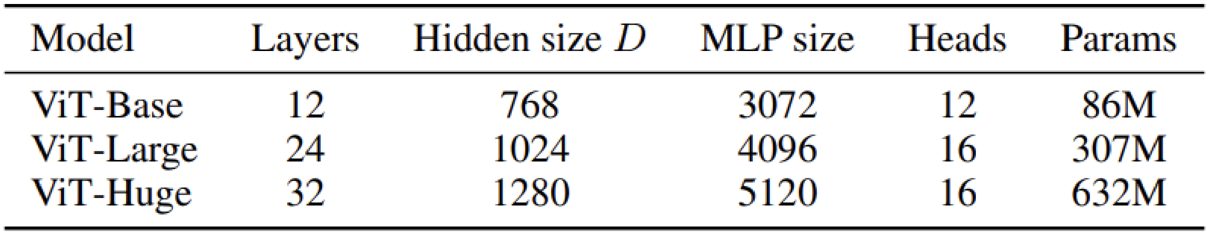
Formula to calculate patch processing, linear embedding, and softmax function

Name	**Formula**	**Description**
Patch processing	$N = \frac{HW}{P^2}$	Input Image I of size H x W x C (height, width, and channels) respectively. P is the patch size. N is the number of patches calculated P x P.

Name	Formula	Description
Linear embedding	$N \times (P^2C)$	P is the patch size. N is the number of patches calculated using P x P. C is the number of channels
Softmax function	$\left(\frac{Q_i k_j^T}{\sqrt{d_k}}\right)$	For each embedded patch transformer computes query vectors using Query Q, Key K, and Value V vectors

Figure 2

Visoion Transformer model variants



Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Model Optimization. When compared to CNNs, Transformer models are highly efficient and can save on computational resources, as they can deal with parallel processing which CNNs lack. This makes them an ideal choice for handling image data. Since the volume of image data can be enormous when subjected to real-time processing. The use of Graphics Processing Units (GPUs) or Tensor Processing Units (TPUs) can also substantially increase the processing power as GPUs and TPUs inherently are capable of handling parallel processing unlike Central

Processing Units (CPUs) which handle the execution of processes in sequence. Additionally, implementation of sharding techniques can also enable faster processing and is a viable option in scenarios where integration of GPU or TPUs is not possible.

Research from Lepikhin et al. (2020) showcased that implementation of GShard in transformer-based models can significantly increase computational efficiency, particularly for large-scale neural networks. The sparsely-gated mixture-of-experts transformer with 600 billion parameters is an instance of a sophisticated model that can be effectively trained to employ GShard. This model utilizes automatic sharding to allocate computational workloads across TPUs, thereby optimizing neural networks and necessitating only a fraction of the time and resources.

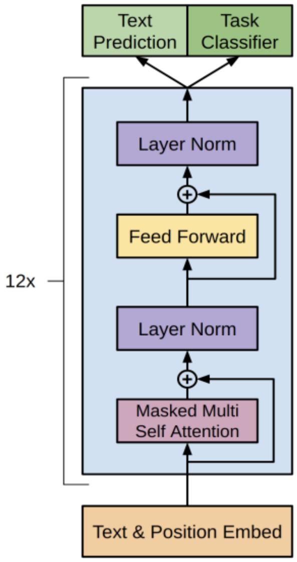
Generative Preprocessed Training 2 (GPT-2)

GPT-2 is a sophisticated neural network that is designed to replicate aspects of human-like language processing. GPT is composed of transformer-based architecture which is suitable for tasks like natural language processing and is well known for generating coherent and contextually relevant text.

According to Radford, A., et al. (2019), the GPT is based on transformer-based decoder architecture which utilizes a self-attention mechanism and comprises 12 such transformer layers in particular. The GPT's transformer architecture is depicted in Figure 3.

Figure 3

Architecture of GPT's transformer decoder



Note. The figure depicts the detailed architecture of GPT's Transformer-based decoder architecture. From "Improving Language Understanding by Generative Pre-Training" by Alec, A. (2018), openAI.

A unique approach of this transformer is the self-attention mechanism. The model involves two phases. The first phase is unsupervised pre-training in which the input of text data and labels are segregated into the corpus of tokens represented by U and then the attention score for each input token is calculated using the likelihood L1, the objective here is to maximize the likelihood. Furthermore, the model applies the multi-head self-attention over the input tokens and then is fed to the feed-forward layer to generate output distribution over the target tokens. Finally, in the second phase after training the model supervised fine-tuning is performed, in which the pre-trained model is adapted to a specific task of prediction P. Additionally, providing an objective during fine-tuning represented as L3 demonstrated the ability to improve generalization and accelerated convergence. The formula for the corpus of tokens U, Likelihood

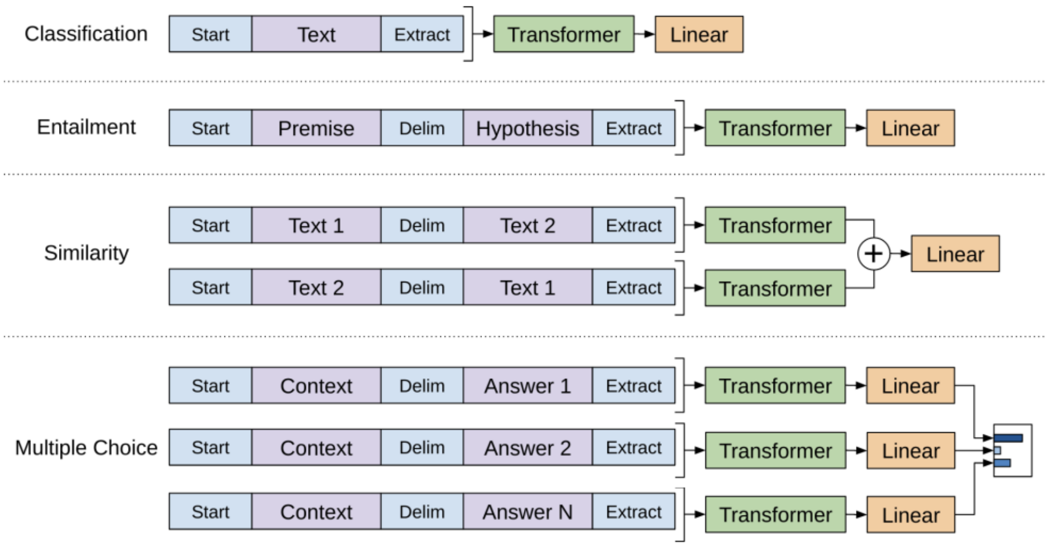
L1, Prediction P, and Objective L3 are shown in Table 2. The input transformation architecture is depicted in Figure 4.

Table 2

Formula for Corpus of Tokens, Likelihood, Prediction and Objective

Name	**Formula**	**Description**
Corpus of Tokens	$u\ =\ \{u_{1},.\ .\ .\ .\ .,u_{n}\}$	Input text and labels segregated into a corpus of tokens
Likelihood	$L_{1}(u)\ =\ \Sigma_{i}logP\ \{u_{i}\|u_{i-k},\ .\ .\ .\ .\ ,u_{i-1};\ \theta\ \}$	Likelihood where u is a corpus of tokens, P is conditional probability and Θ are model parameters
Prediction	$P(u)\ =\ softmax\ \{h_{n}W_{e}^{T}\}$	Hn represents the transformed representation of input and WeT output from the transformer block
Objective	$L_{3}(C)\ =\ L_{2}(C)\ +\ \lambda\ *\ L_{1}(C)$	L3 represents objective function and **λ** is weight

Figure 4



Model Supports

Environment, Platform and Tools

Several factors such as environment, platform, and tools play a vital role in the efficient development of the CLIP-GPT-2 model. Setting up an environment that favors the seamless development of the model is quintessential, especially while dealing with image data as image data contains a magnitude of information, processing it requires high computational resources. The transformer-based model architecture is capable of utilizing parallel computation resources such as GPU. Thus to harbor the data processing and modeling requirements suitable allocation of resources is provided with exceptional processing capabilities. The models underwent training and assessment on a system that was built on Windows 11, with an Intel Core i9-13900H 16 cores CPU, 32 GB RAM, and NVIDIA-Geforce RTX 4070 GPU. Programming, evaluation, and graph plotting were carried out using Visual Studio Code and Jupyter Notebook development environments.

The models were primarily built on Python, as Python is a vast programming language with many libraries, and the majority of data manipulation and models are readily available in Python making it an ideal choice for model-building purposes. Models were loaded using the transforms library and were configured using transforms.VisionEncoderDecoderModel class in which ViT was configured as encoder and GPT-2 as decoder, the feature extractor and tokenizer were instantiated using transforms.AutoTokenizer library and data transformations were applied using PyTorch's module and a Comma Separated Values (CSV) file containing captions was loaded as a dataframe using the pandas library. The data was split using training and validation set using sklearn.model_selection. Finally, training arguments were initialized using transforms.Seq2SeqTrainingArguments and models were trained using transforms.Seq2Trainer and the model were evaluated using several metrics such as training loss, validation loss, precision, recall, f-measure using ROUGE-2 (Recall-Oriented Understudy for Gisting Evaluation) from datasets library and generated captions quality evaluation using BLEU (Bilingual Evaluation Understudy) from nltk.translate.bleu_score. Finally, the metrics were plotted using matplotlib. Table 3 below shows the library details used to implement the model.

Table 3

Libraries used for model building

Library	Method	Usage
Transformers	VisionEncoderDecoderModel	Pretrained composite transformer model composed of ViT-based encoder and GPT-2 as language decoder
	ViTFeatureExtractor	Preprocessing images to be fed into Vission Transformer
	AutoTokenizer	Tokenizer for model
	GPT2Config	GPT-2 model configuration class
	default_data_collator	Batch processing for data pipeline
PyTorch	torch.nn	PyTorch class methods for building neural networks
	torch.nn.functional	Submodule for operations such as convolutional operations, activation functions, and loss functions
	torch.io	Utilities for image input/output operation
	torch.transforms	Submodule for image transformations such as normalizing, rescaling, cropping, etc

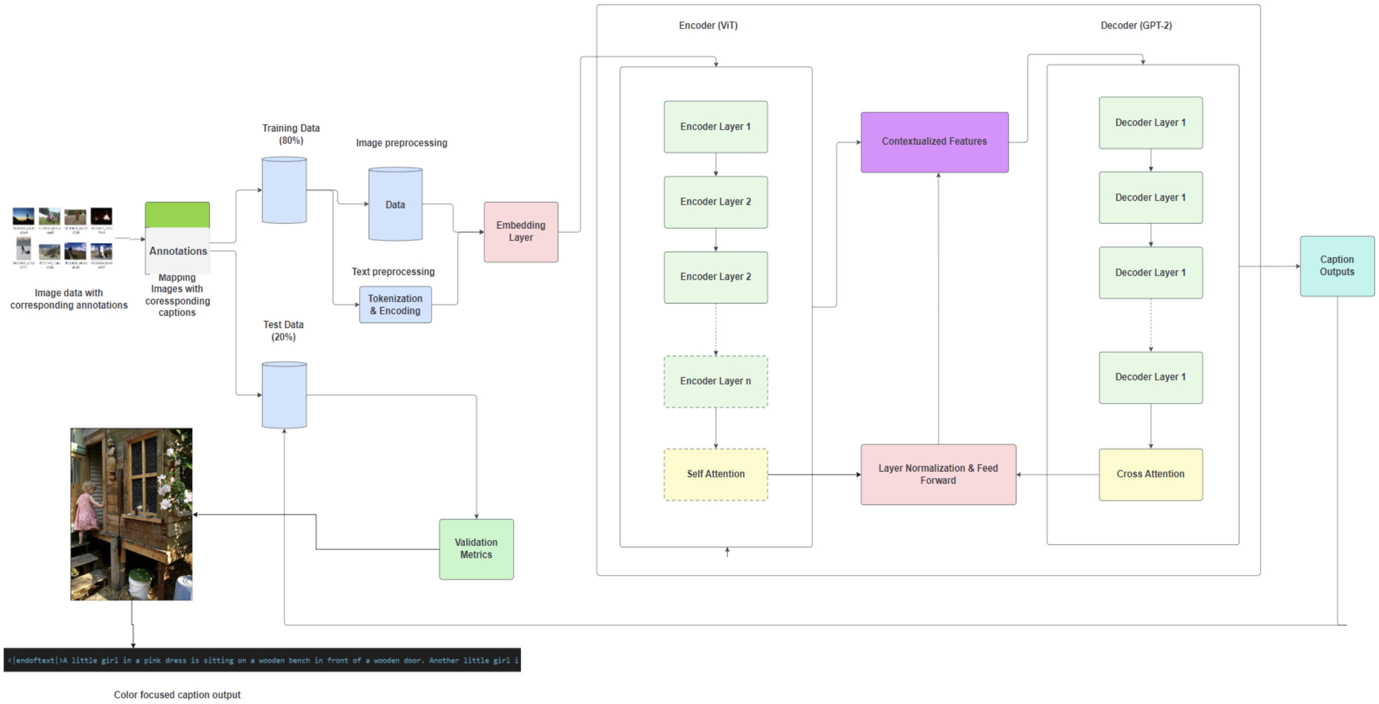
Library	Method	Usage
Scikit-Learn	train_test_split	Splitting dataset into train and validation set
Numpy	numpy	Generation matrices and performing mathematical operations
Pandas	pandas	Reading data into dataframes and manipulating them
Pillow	PIL.Image	Reading, writing, and manipulating image files

Model Architecture and Dataflow

Figure 5 below depicts the data flow architecture of the proposed CLIP with the GPT-2 model. The image dataset gathered from COCO and Flickr30k is first mapped in which each image is labeled with its corresponding annotation, the data is then segregated into train and validation sets with 80% and 20% ratios respectively. The train set is then subjected to preprocessing in which the image data is scaled and normalized and captions are tokenized. The image data is then converted into a linear vector and then projected into embedding space, similarly, text data is embedded to retain the semantic meaning of the words and is fed into the ViT encoder. The encoder layer then processes the image embeddings, each layer includes layer normalization and sel-attention mechanisms followed by a feed-forward network which helps stabilize the learning process, this enables the encoder to effectively extract and refine the features from the data. The output from the encoder layer is a set of contextualized features representing the various aspects of the image. The GPT-2 decoder receives the contextualized image features and text embeddings, the decoder then leverages the attention mechanism for text embeddings and cross attention mechanism for image features. The output generated from the decoder is a color-focused caption of the input image. After the successful execution of the training part, the model's performance is evaluated against the validation set, this helps assess the model's performance in generating captions in contrast to the original captions. Finally, the trained and validated models are subjected to various evaluation metrics such as ROUGE-2 to assess the model’s performance during training and generated captions are assessed with actual captions using the BLEU score and Metric for Evaluation of Translation with Explicit Ordering (METEOR) score.

Figure 5

Data flow architecture diagram of CLIP-GPT-2 Model



```
<|endoftext|>A little girl in a pink dress is sitting on a wooden bench in front of a wooden door. Another little girl i
```

Model Comparison and Justification

The two models proposed for this study are CLIP-GPT-2 and Convolutional Neural Network and Long Short-term Memory (VGG16-LSTM). Although both CLIP-GPT-2 and VGG16-LSTM are types of neural networks and are popular choices for object detection and captioning tasks. However, both have quite different architectures and are suitable for different scales of data. CLIP-GPT-2 is efficient for large and complex datasets where capturing intricate details is crucial. Even though the major limitation of CLIP-GPT-2 is that they require a substantially large dataset for effective training are computationally intensive and can overfit when worked with a small dataset. On the contrary, VGG16-LSTM performs exceptionally well with smaller datasets, especially structured sequential data, but can overfit with large data and perform adversely in highly complex tasks. Table 4 below compares and contrasts the differences between these models.

Table 4

	CLIP-GPT-2	**VGG16-LSTM**
1	Tranformer-based neural network	Convolutional neural network (Linear model) and an LSTM network (sequential model)
2	Efficient with images and text data	Efficient with images and time-series data
3	Typically requires large data. Less effective with small sparse data	Effective with small-sized data, able to handle sparse data
4	Vulnerable to overfitting when subjected to small dataset	Vulnerable to underfitting on large complex dataset
5	Requires substantial pre-processing like normalization for images, tokenization for text	Less pre-processing required
6	Larger training times due to model's size and complexity	Lesser training time required
7	Space complexity is high due to multiple parameters and larger model size	Moderate complexity as it requires fewer parameters

	CLIP-GPT-2	VGG16-LSTM
8	Can reap benefits of computational resources as it's efficient with parallel processing	Lesser benefits from computational resources as it inherently uses sequential processing

Despite their differences, they share some similarities too. Both models are based on deep learning and are neural network frameworks and are capable of handling complex tasks, they both leverage multimodal learning capabilities making them highly efficient for tasks like image and text processing. They both can handle sequential data and perform efficient feature extraction, although they use different mechanisms. They both demonstrate substantial performance growth in complex tasks as they can efficiently utilize computational resources. Both models are suitable for complex tasks but require different needs of parameter tuning, and can be leveraged to mitigate the growing needs of complex task handling, especially in the field of computer vision and natural language processing.

Model Evaluation Methods

Evaluation metric plays an important role in machine learning and data science projects as they provide objective measures for assessing the performance of models. These metrics enable the practitioners to effectively assess the performance of their models and validate if their model is performing as expected. This can be effectively achieved by employing suitable metrics such as accuracy, precision-recall, and other pertinent aspects. Evaluation metrics also additionally help identify the strengths and weaknesses of the model, ensuring that they not only perform well on training data but also generalize to novel or unseen data. This is of utmost importance in building robust and reliable models for real-world applications. For tasks like image captioning metrics such as ROUGE-2 and BLEU are usually employed to measure the accuracy of generated captions against the original captions to validate the quality of generated text and ensure that the model can retain the context efficiently.

Accuracy

Accuracy is one of the simple and most intuitive performance metrics. According to Japkowicz (n.d.), Accuracy is a simple ratio of correct predictions divided by the total number of predictions made. In this case, it represents the model's capability to accurately generate captions concerning the original caption. Equation 1 below depicts the calculation to measure accuracy.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

Where, TP = True Positives, number of correctly predicted positive instances, TN = True Negatives, number of correctly predicted negative instances, FP = False Positives, number of incorrectly predicted as positive instances, and FN = False Negatives, number of incorrectly predicted as negative instances.

Cross Entropy Loss

In machine learning, Loss is another crucial metric that indicates how bad a model predicted at every cycle. The CLIP-GPT2 is an encoder-decoder model and typically uses a Sequence Cross-Entropy Loss which is widely used in sequence-to-sequence learning tasks as suggested by Sutskever et al. 2020. The loss measures the negative likelihood of the predicted token sequence given the actual sequence. It then aggregates the cross-entry loss for each token in all the sequences in a batch. Mathematically, it can be represented as shown in equation 2.

$$L =- \frac{1}{N}\Sigma_{i=1}^{N}\Sigma_{j=1}^{M_i} log(p_{ij}(y_{ij})) \tag{2}$$

Where, N is the batch of tokens, M_i is the length of the batch and L is loss. $p_{ij}(y_{ij})$ represents the predicted probability of correct token at y_{ij} position j in sequence i.

Recall-Oriented Understudy for Gisting Evaluation (ROUGE-2)

ROUGE is assessment metric for generated captions. The evaluation is based on a pair of consecutive words technically referred to as bigrams. According to Lin, C. Y. (2004), ROGUE calculates the overlap of bigrams in the generated text with the original text. It focuses on co-occurrences of a pair of words, making it an ideal choice for assessing coherence and fluency in the text. ROGUE calculates the three factors to assess the occurrence namely, Recall (R), Precision (P), and F-Measure (F) which is often a balanced 1:1 ratio of precision and recall, and can be mathematically represented as shown below in the equation 3.

$$\begin{aligned} R &= \frac{Number\ of\ overlapping\ bigrams\ in\ generated\ and\ reference\ text}{Total\ number\ of\ bigrams\ in\ text} \\ P &= \frac{Number\ of\ overlapping\ bigrams\ in\ generated\ and\ reference\ text}{Total\ number\ of\ bigrams\ in\ text} \\ \mathrm{F} &= 2 \times \frac{P \times R}{P + R} \end{aligned} \tag{3}$$

Bilingual Evaluation Understudy (BLEU)

It's a popular statistic used in NLP to compare the generated text in reference with one or more original text. Papineni et al. (2001) demonstrated that BLEU measures the frequency and precision count of n-grams in the generated text compared to the reference text. It can be represented mathematically as shown in equation 4.

$$BLEU = BP \times exp(\Sigma_{n=1}^{N} w_n log(p_n)) \quad (4)$$

Where, p_n is the n-gram precision and w_n is the weight of each n-gram.

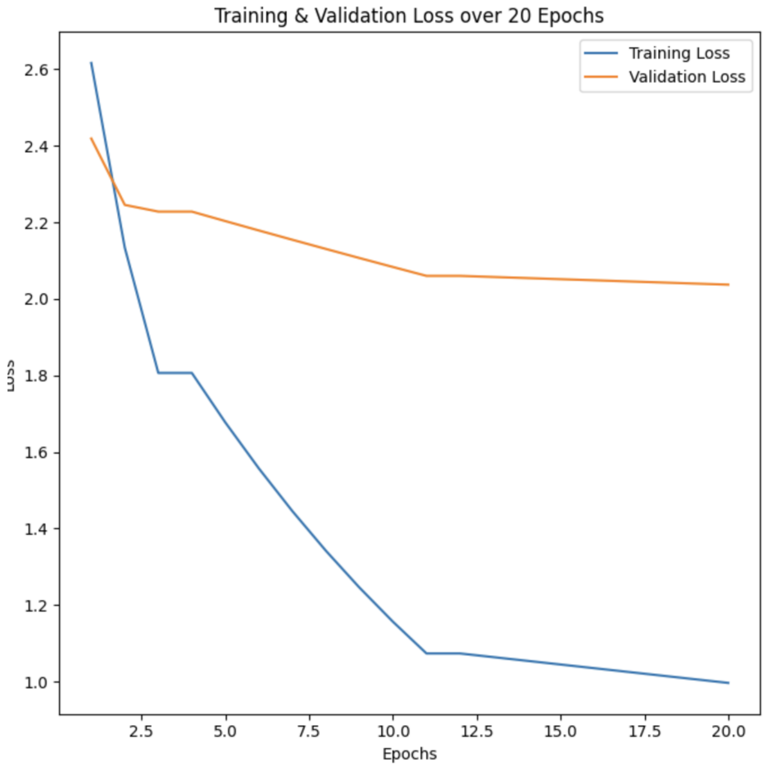
Model Validation and Evaluation

CLIP-GPT-2

The model underwent to training phase in which the training data was fed and ran over 20 epochs. An epoch in this context refers to one complete cycle through the full training dataset. On successful training of the model, the model was then subjected to a rigorous evaluation to validate its performance. In this, evaluation phase, the model achieved a final training loss of 0.9 and a validation loss of 2.1, these values are indicators of the model's error rate and they quantify the difference between the model's predicted captions and the actual captions. The loss values observed show a prominent downtrend implying that the the model fitting is being done efficiently and the captions generated are becoming more aligned with the expected outputs. Figure 6 below visualizes the graph portraying the loss values over the epochs.

Figure 6

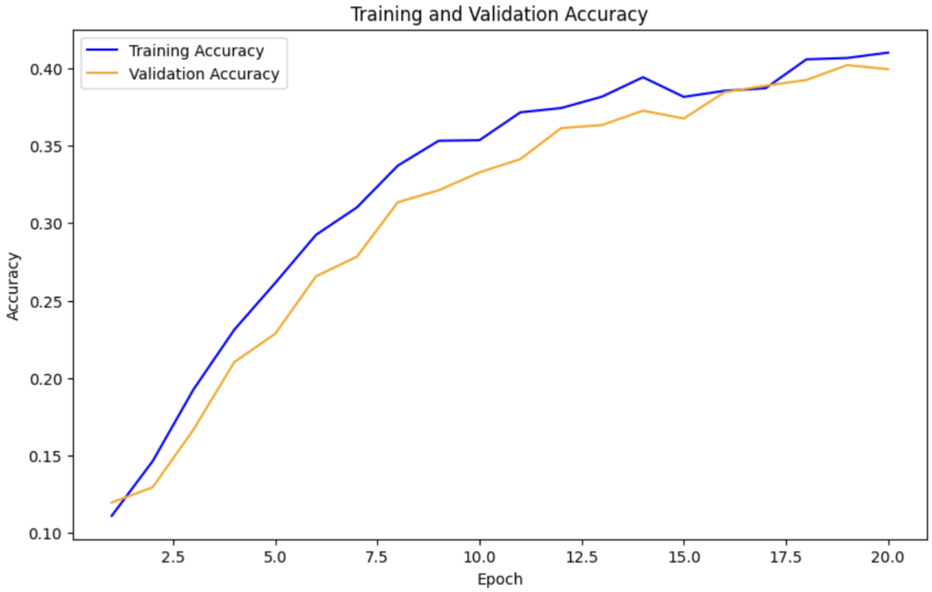
CLIP-GPT-2 model loss curve



Additionally, the model demonstrated a significant validation accuracy of 0.396 approximately 40%, maintaining training accuracy in the same range of about 0.413 about 41%. The validation accuracy signifies that the model is capable of accurately generating color-focused and coherent captions. The validation accuracy signifies the model's adeptness in captioning new images that were not present in the training data. The accuracy statistics underscore the model's effectiveness in learning and robust pattern recognition capabilities. The near convergence is commendable as it suggests that the model resists overfitting. Figure 7 below depicts the model’s accuracy scores. Additionally, to evaluate the accuracy of generated captions the model was evaluated using ROGUE-2 and BLEU scores.

Figure 7

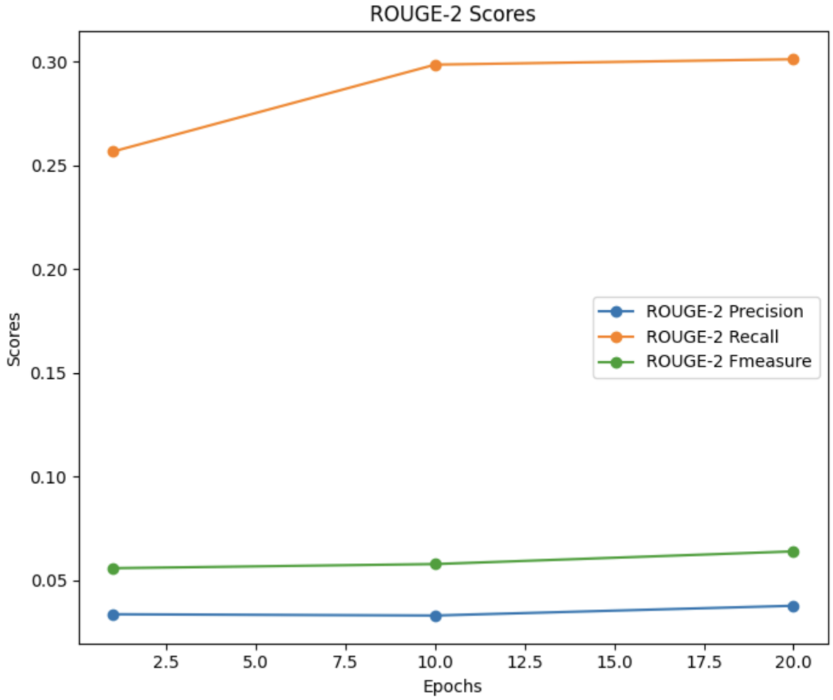
CLIP-GPT-2 models accuracy scores



In addition ROUGE-2 and BLEU scores are used to evaluate the quality of generated captions. A ROUGE-2 precision score of 0.03 suggests the model has some ability to capture some relevant content, but still fails to identify the accurate context when compared to original images. In general, the model needs improvement in reproducing exact word sequences from the references, based on the f-measure Figure 8 below visualizes the ROGUE-2 scores.

Figure 8

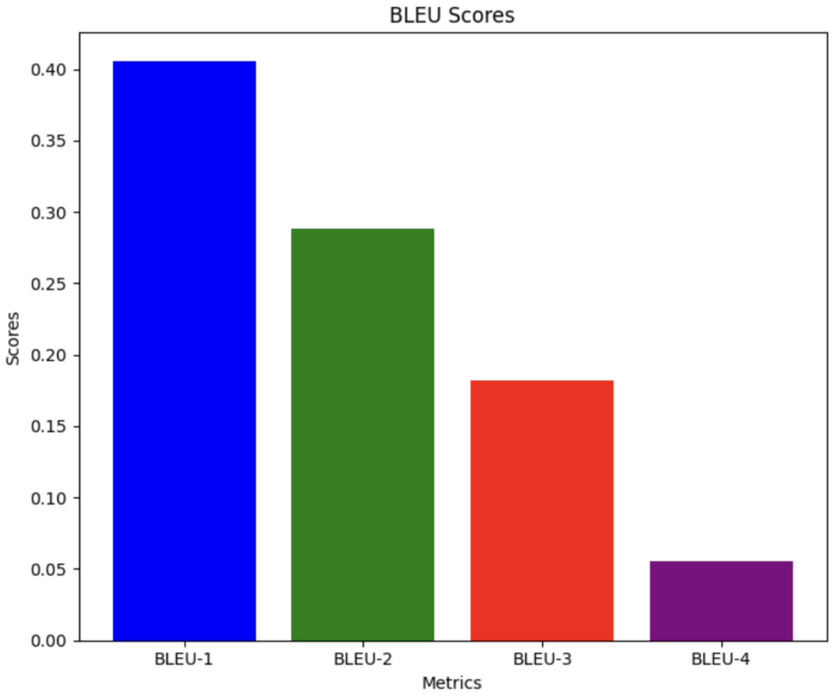
Line plot depicting ROGUE–2 scores



When matching single words (BLEU-1) and four-word sequences (BLEU-4), the model performs reasonably well, but its performance decreases significantly when matching more complex phrases. The model may capture key vocabulary, but it may not be able to translate it into coherent, contextually accurate phrases to enhance captioning accuracy. Figure 9 below visualizes the BLEU scores.

Figure 9

Bar chart depicting BLEU scores



Overall, the metrics suggest that the model possesses generalization capabilities as they are evident in the model's accuracy scores and convergence of train and validate accuracies suggesting a great fitting. The decreasing loss also suggests the model’s effective learning rate. However, the ROGUE-2 and BLEU scores suggest captions generated are cohesive but are not contextually accurate and model refinements need to be done to achieve alignment with human-like language generation capabilities, which possibly can be achieved using extensive dataset, refining the model or by re-iterating the architecture.

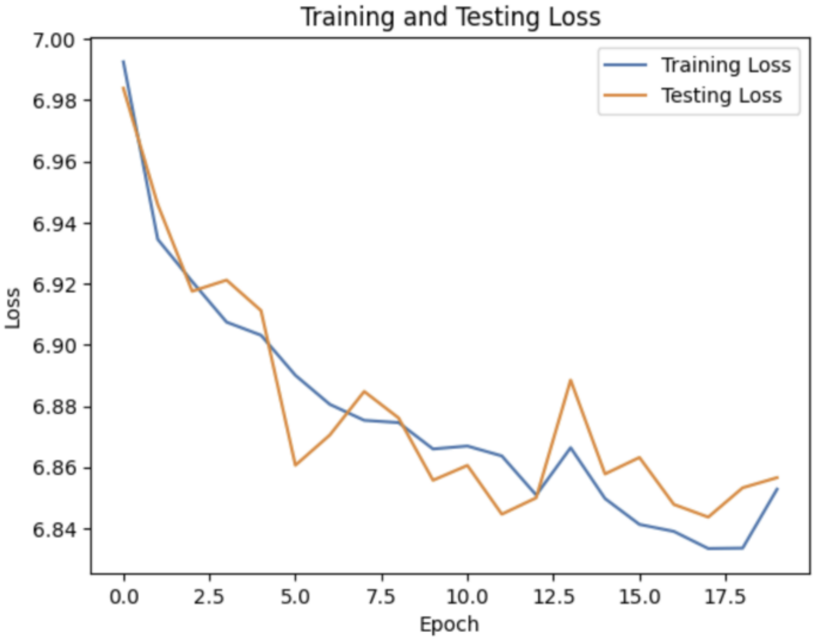
VGG16-LSTM

When compared to the evaluation results of the CLIP–GPT-2 model, VGG16-LSTM achieved a loss value of 6.8712% of training data which is significantly higher than CLIP-GPT-2 when run over 20 epochs. The key reason behind this could be the CLIP-GPT 2’s model complexity and ability to understand complex problems which CNNs lack.. This implies that the

VGG16-LSTM learning rate is significantly slower. Figure 10 below depicts the loss value trends of VGG16-LSTM.

Figure 10

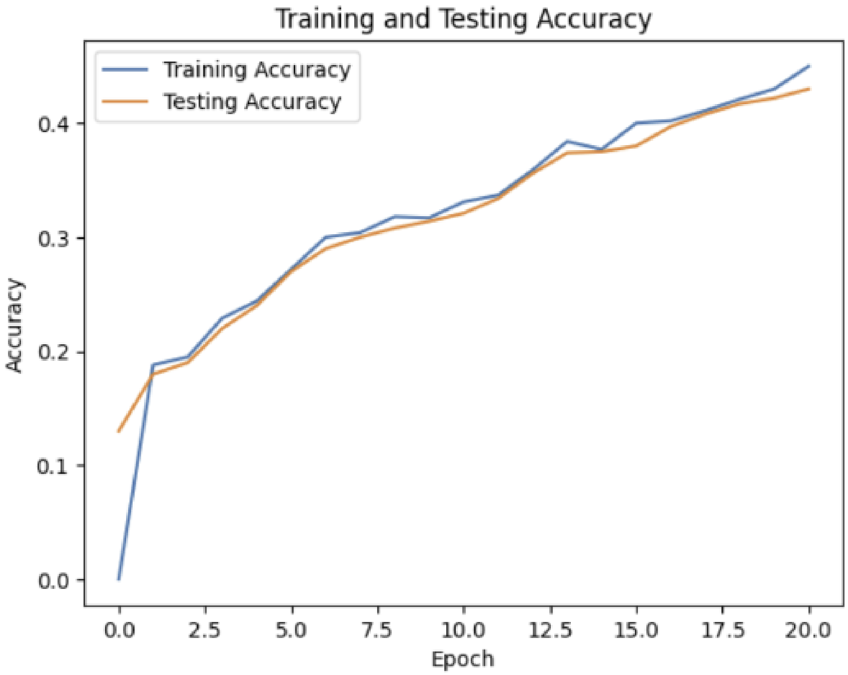
VGG16-LSTM model loss curve



The accuracy score of VGG16-LSTM were about 45% for the train set and 43% for the validation set which are slightly better than CLIP-GPT-2 and the accuracy scores are converging significantly, suggesting that the model’s ability to identify the image context is better. The key reason behind this could be the model's ability to efficiently handle the sequential data as CNNs can efficiently handle image data and can help them understand the context of the image better. Figure 11 below depicts the accuracy scores of VGG16-LSTM.

Figure 11

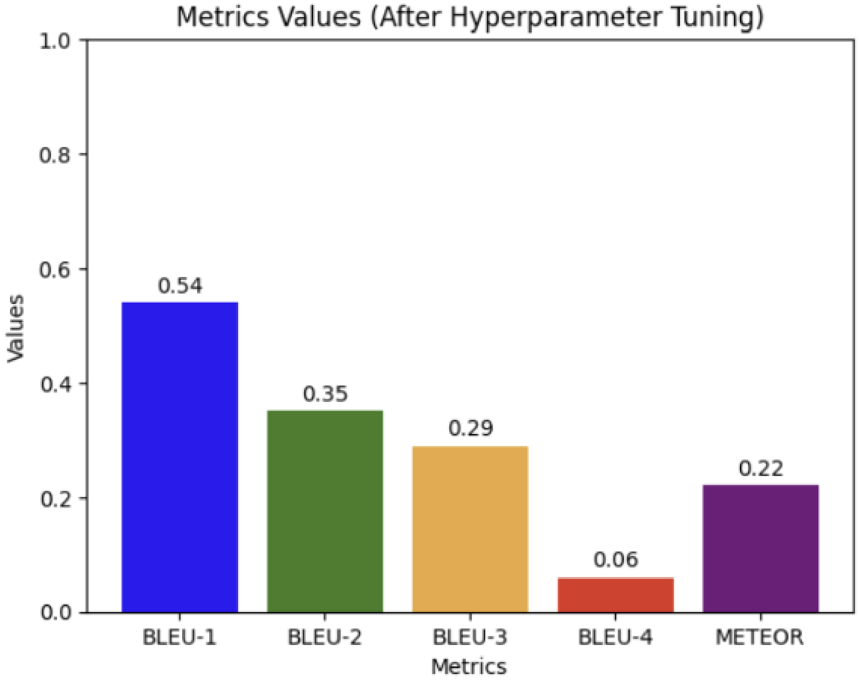
VGG16-LSTM models accuracy scores



Finally the BLEU and METEOR scores of the VGG16-LSTM are about 0.54 for BLEU-1 and 0.22 for METEOR which are slightly better than the CLIP-GPT-2 which also suggests that the model is efficient in understanding the context of the image and can generate cohesive captions. Figure 12 below shows the BLEU and METEOR scores of the VTT16-LSTM model.

Figure 12

VGG16-LSTM models accuracy scores



Results

After analyzing and comparing the accuracy and loss values of the CLIP-GPT-2 and VGG16-LSTM models, significant differences were observed. When it comes to accuracy and BLEU evaluation, the VGG16-LSTM model performed better than the CLIP-GPT-2. However, the loss values of CLIP-GPT-2 were better than the VGG16-LSTM. Overall the accuracy scores of 45% of VGG16-LSTM in contrast to the 40% accuracy of CLIP-GPT-2, suggests that the VGG16-LSTM is efficient at identifying images and higher BLEU scores also suggest that the model is capable of generating more cohesive captions.

On the contrary, the loss scores of CLIP-GPT-2 are significantly better than the VGG16-LSTM which suggests that the CLIP-GPT-2 has adept learning capability and can efficiently learn the intricacies of the image while training. Comparison of CLIP-GPT-2 and VGG16-LSTM are shown in Table 5 below.

Table 5

Model comparison of CLIP-GPT-2 and VGG16-LSTM

Model	Num. of Epochs	Training Loss	Test/Validation Loss	Training Accuracy	Testing/Validation Accuracy
CLIP-GPT-2	20	0.9	2.1	0.41	0.396
VGG16-LSTM	20	6.8	6.8	0.45	0.43

Conclusion

Based on the above findings, it can be concluded that the VGG16-LSTM model is well suited for image captioning tasks. As a result of its higher accuracy of 0.45 and better BLEU scores of 0.54, suggests that it's more reliable and precise at generating coherent captions. Hence, it can be concluded that VGG16-LSTM is an effective model for this study as it has a simple architecture and can be easily hypertuned. However, it is critical to remember that models should be chosen based on the particular needs and limitations of the project. While selecting the best model for a particular application, other elements like dataset characteristics, computational complexity, and resource limitations should also be taken into account.

Limitations

Despite being a novel approach to image captioning, the CLIP-GPT-2 model has certain limitations. Due to the computational intensive needs, it can only be trained on smaller datasets, which may limit its capacity to generalize to new data. The model's processing demands require advanced optimization techniques to ensure efficient handling of diverse image datasets, which demands domain expertise. These difficulties introduce a trade-off between the model's learning capabilities and computational efficiency.

Future Scope

The CLIP-GPT-2 models set the standard for image captioning tasks with their impressive visual and linguistic capabilities. The model achieves intricate image recognition with well-articulated captions by extending the sophisticated image processing capabilities of CLIP and the powerful language processing of GPT-2, both of which have been pre-trained on a vast array of data. The model performs better than conventional models, which have trouble deriving intricate subtilities of language processing. Despite its computational requirements, CLIP-GPT-2 is appropriate for real-world applications due to its resilience against overfitting, which is supported by its consistently low validation losses. This makes it well-suited for the changing needs of growing datasets. A VGG16-LSTM ensemble model with CLIP-GPT-2 can also be developed, utilizing natural language processing and sequential processing capabilities. This model can be applied to image captioning tasks, potentially yielding benchmark results.

References

Brown, T. B. (2020, May 28). Language Models are Few-Shot Learners. arXiv.org. https://arxiv.org/abs/2005.14165

Dosovitskiy, A. (2020, October 22). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv.org. https://arxiv.org/abs/2010.11929

Japkowicz, N., & Shah, M. (2011, January 17). Evaluating Learning Algorithms. *Cambridge University Press*. https://doi.org/10.1017/CBO9780511921803

Lepikhin, D. (2020, June 30). GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding. arXiv.org. https://arxiv.org/abs/2006.16668

Lin, C. Y. (2004, July 1). ROUGE: A Package for Automatic Evaluation of Summaries. *ACL Anthology*. https://aclanthology.org/W04-1013

Lin, T. Y. (2014, May 1). Microsoft COCO: Common Objects in Context. arXiv.org. https://arxiv.org/abs/1405.0312

Minaee, S. (2020, January 15). Image Segmentation Using Deep Learning: A Survey. arXiv.org. https://arxiv.org/abs/2001.05566

Papineni, K., Roukos, S., Ward, T. J., & Zhu, W. J. (2001, January 1). BLEU. https://doi.org/10.3115/1073083.1073135

Radford, A., (2018) Improving Language Understanding by Generative Pre-Training. (2018, June 11). OpenAI. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

Radford, A., Wu, C., Luan, D., & Sutskever. I., (2019). Language Models are Unsupervised Multitask Learners. OpenAI. https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf

Radford, A. (2021, February 26). Learning Transferable Visual Models From Natural Language Supervision. arXiv.org. https://arxiv.org/abs/2103.00020

Sutskever, I. (2014, September 10). Sequence to Sequence Learning with Neural Networks. arXiv.org. https://arxiv.org/abs/1409.3215

Vaswani, A. (2017, June 12). Attention Is All You Need. arXiv.org. https://arxiv.org/abs/1706.03762

Young, P. T., Lai, A., Hodosh, M., & Hockenmaier, J. (2014, December 1). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics. https://doi.org/10.1162/tacl_a_00166

Yu, J. (2019, May 20). Multimodal Transformer with Multi-View Visual Representation for Image Captioning. *arXiv.org*. https://arxiv.org/abs/1905.07841