

Course	94691 Deep Learning
Examiner	Dr. Mir Kabir
Assignment	Assessment Task 3 - Image Captioning
Due date	22/05/2024

Contents

1. Presentation of Data	3
2. Walk-through of data preparation performed	3
3. Presentation of architectures tested during experimentation and rationales.....	5
4. Presentation of metrics and loss functions used	7
5. Analysis of models performance and limitations	8
6. Walk-through of remaining issues and recommendations.....	16
9. References	18

1. Presentation of Data

The dataset used for this analysis is the Flickr8k dataset, composed of 8,000 images collected from the photo-sharing website Flickr. Each image within the dataset has a corresponding caption that is annotated with five descriptive words. The images that comprise the dataset have an array of scenes, activities and objects depicted within them, whilst also being of high quality. The captions use these scenes, activities, and objects to describe these images. The dataset is organised into an images folder and a captions file. This is a widely used dataset in image captioning research, making it extremely relevant to our analysis. The diversity of images in this dataset ensures that the models are trained to handle varied contexts (Modi, 2018).

2. Walk-through of data preparation performed

MobileNetV3 & LSTM

The data preparation began by downloading the Flickr8k dataset from Kaggle using the Kaggle API. A 'Vocabulary' class was then created to allow for the numericalisation and tokenisation of the captions within the dataset to be handled (Pan et al., 2020).

The two main components were build_vocabulary, which counts the frequencies of words that occur within the captions. 'numericalise' uses the tokenised text from the vocabulary to then create a numerical form. The vocabulary class was used on the loaded captions to build the vocabulary of the dataset (Falconí et al., 2019).

A 'Flickr8kDataset' Class was created to ensure that each of the images and captions had been loaded into Collaboratory and that transformations were applied. The final step of the data preparation was to load the data into the designated training and testing Data Loaders for batching and shuffling (Falconí et al., 2019).

Resnet & LSTM

The Flickr8k dataset was downloaded and unzipped for preprocessing from Kaggle to be utilised in the image captioning model. The architecture consisted of a feature extractor for images using a CNN model which was Resnet in this case. the other part consisted of LSTM which consisted of a sequence-based model for captions generation (Rezende et al., 2017).

Necessary libraries and utilities were utilised for the project, involving image processing (using ResNet50 model) and natural language processing tasks (using NLTK and Keras for text preprocessing and evaluation) (Rezende et al., 2017).

```
[ ] # head of the image_tokens dataframe
```

`image_tokens.head()`

	img_id	img_caption
0	1305564994_00513f9a5b.jpg	<start> A man in street racer armor be examine...
1	1305564994_00513f9a5b.jpg	<start> Two racer drive a white bike down a ro...
2	1305564994_00513f9a5b.jpg	<start> Two motorist be ride along on their ve...
3	1305564994_00513f9a5b.jpg	<start> Two person be in a small race car driv...
4	1305564994_00513f9a5b.jpg	<start> Two person in race uniform in a street...

Next steps: [Generate code with image_tokens](#) [View recommended plots](#)

Figure 1.0 – head of image_tokens data frame

The conversion of the text files to pandas dataframe was carried out for organizing and structuring the image-caption data into a format that is suitable for further processing, such as model training and evaluation. Visualisations of some images and dataframe structure was also included to understand the features of the dataset (Rezende et al., 2017).

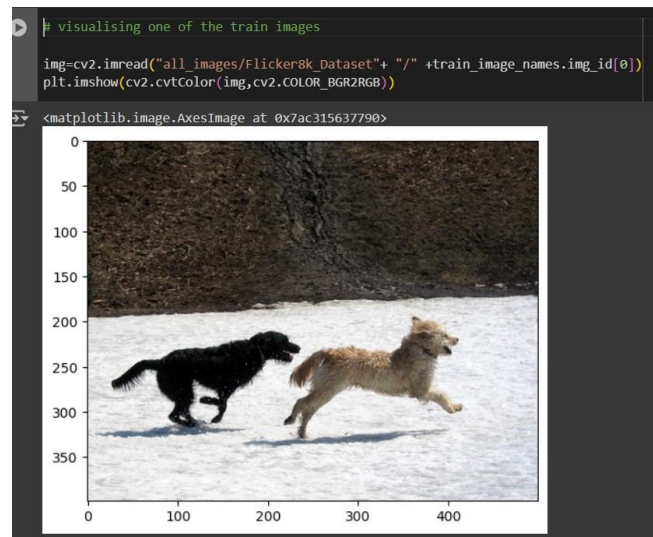


Figure 2.0 – visualisation of one of the train images

Creation of train, test and validation dictionary having key as the image id and value as a list of its captions was carried out. ResNet50 model was utilised for encoding the images to be utilised in the image captioning architecture. Encoding was carried out for images and forming dictionaries containing mapping of image_id to image encodings (Castro et al., 2022a).

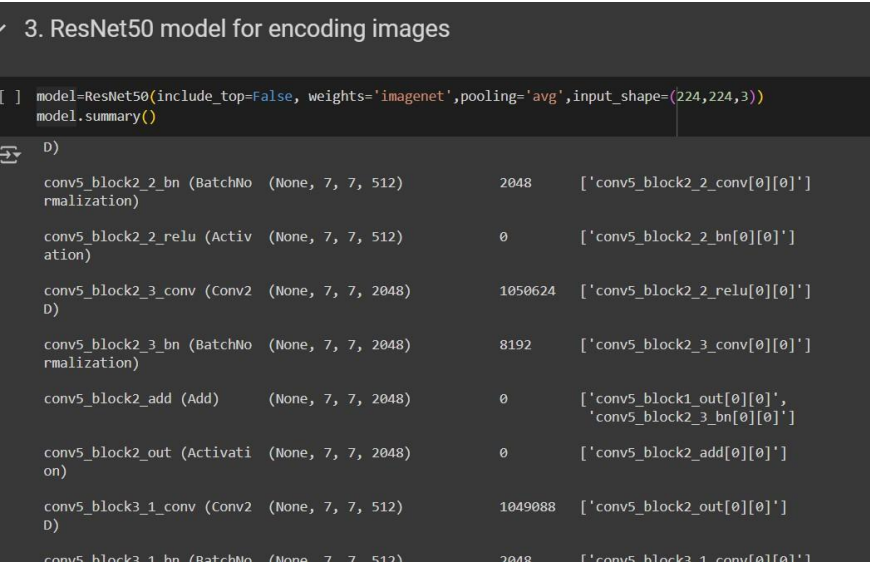


Figure 3.0 – utilisation of ResNet50 model for encoding images.

Hyper parameters for vocabulary size and maximum length were utilised as well. The captions associated with training images, combines them into a single string, and then computes the total number of words as well as the number of unique words in these captions. This could be useful for tasks like vocabulary analysis or preprocessing before training a language model (Atliha & Šešok, 2022).

Creating dictionaries containing mapping of words to indices and indices to words for image captioning model. Transforming data into dictionary mapping of image_id to encoded captions was also utilised in the image captioning model (Suresh et al., 2022).

3. Presentation of architectures tested during experimentation and rationales.

MobileNetV3 & LSTM

The testing of the Convolutional Neural Network MobileNetV3 architecture for this experiment was chosen due to its effectiveness on mobile devices whilst also achieving high accuracy results. Incorporating depth wise separable convolutions within the MobileNetV3 architecture significantly reduces the need for large parameters and the computational cost (Hossain et al., 2019a).

These depth wise separable convolutions have a single convolutional filter within each input channel and only use a 1x1 convolution when combining the output. The other vital component of the MobileNetV3 architecture is the Squeeze-and-Excitation (SE) Blocks. This architecture component allows the interdependencies between the channels through explicitly modelling them. The whole architecture of MobileNetV3 looks to reduce the number of parameters and create a more streamlined model that is lightweight, which is a highly beneficial component (Stefanini et al., 2022).

Long Short-Term Memory (LSTM) is a recurrent neural network that can take long-term dependencies in sequential data and allow them to be used in language modelling and prediction. The memory cells in LSTM allow the cell state to be maintained over time and allow long-term dependencies to be tracked (Geetha et al., 2020).

There are three types of gates in LSTM that can regulate the flow of information throughout the learning process: the forget gate, the input gate and the output gate. The ability of LSTM to capture the long-term and short-term dependencies effectively is a result of the incorporation of the cell state, the long-term memory, and the hidden state, the output of the LSTM cell (Xu et al., 2017a).

Overall, LSTMs were chosen for the experiment due to their proven ability to provide accurate and reliable results in language modelling tasks and take on sequential data, which needs to maintain the context of previous data through long sequences (Geetha et al., 2020).

Resnet & LSTM

Breakdown of the significance of ResNet50 and LSTM in the image captioning model utilised for flicker8k dataset.

ResNet50:

Pre-trained Feature Extractor: ResNet50 is a powerful and widely used convolutional neural network (CNN) pre-trained on a massive image dataset (likely ImageNet). This pre-training allows it to capture generic visual concepts that are transferable to various tasks, including image captioning (Rezende et al., 2017).

Efficient Feature Representation: ResNet50's architecture incorporates residual connections, which help alleviate vanishing/exploding gradient problems during training of deep networks. This allows it to learn complex image features effectively (Rezende et al., 2017).

Focus on High-Level Information: By using a pre-trained model like ResNet50, the image captioning model doesn't need to learn low-level features (like edges and corners) from scratch. ResNet50 provides a compressed representation focusing on higher-level semantic information relevant for caption generation (Rezende et al., 2017).

LSTM:

Handling Sequential Data: LSTMs (Long Short-Term Memory networks) are a type of recurrent neural network (RNN) specifically designed to handle sequential data like captions. Unlike standard RNNs, LSTMs have internal mechanisms to learn long-term dependencies within sequences (Xu et al., 2017a).

Contextual Caption Generation: In image captioning, the LSTM considers the encoded image features along with the previously generated words in the caption. This allows it to generate captions with context, where each word builds upon the previous ones and reflects the overall image content (Xu et al., 2017a).

Word Prediction: At each step during caption generation, the LSTM network predicts the probability of the next word based on the current context (image features and previously generated words). This iterative process leads to the creation of a complete caption describing the image (Xu et al., 2017b).

Combined Significance:

ResNet50 and LSTM when used together assist to bridge the gap between visual information and textual descriptions. ResNet50 provides a powerful and efficient way to capture the visual essence of an image. LSTM leverages these features to generate contextually relevant captions that describe the image content in a coherent and sequential manner (Rezende et al., 2017).

By using this combination, the image captioning model can learn the complex relationship between visual features and natural language, leading to more accurate and informative captions.

4. Presentation of metrics and loss functions used

MobileNetV3 & LSTM

BLEU, Bilingual Evaluation Understudy, scores were used in this experiment as they allow us to understand the precision of the model. Four variations of BLEU scores were used, BLEU-1, BLEU-2, BLEU-3 and BLEU-4, all representing a different length of n number of words. BLEU-1 considers single words, BLEU-2 considers two-word sentences, BLEU-3 considers three-word sentences, and BLEU-4 considers four-word sentences (Hossain et al., 2019b).

A high BLEU-1 score indicates that the model has demonstrated good precision for single words. If the model displays a lower BLEU-4 score, it insinuates that the model is not good at maintaining coherency throughout its captions. The loss function used throughout this experiment was Cross-Entropy Loss, a widely used loss function for classification problems (Hossain et al., 2019b).

The Cross-Entropy loss function can measure the differences between the actual caption and the predicted caption. The Cross-Entropy Loss function can handle imbalances present within the frequency of words whilst also using gradient descent methods to ensure effective optimisation within the loss function (Hossain et al., 2019b).

Resnet & LSTM

BLEU Scores (BLEU-1 to BLEU-4):

BLEU scores measure the similarity between a generated caption and multiple reference captions in the dataset (e.g., Flickr8k). They consider n-gram overlaps (sequences of n words) between the generated and reference captions. Higher BLEU scores generally indicate better caption quality (Alzubi et al., 2021).

Loss Function (used during model training):

Cross-Entropy Loss:

This is a common loss function used in multi-class classification problems, including image captioning where the model predicts the next word in a sequence. It measures the difference between the probability distribution predicted by the model and the true probability distribution (represented as a one-hot encoded vector) of the next word in the reference caption. Minimizing the cross-entropy loss during training encourages the model to generate captions that align with the reference captions (Alzubi et al., 2021).

Additional Considerations:

Code calculations were presented for the **average BLEU score** across all the generated captions in the test or validation set to evaluate model performance. During training, the model might be optimized using an optimizer like **Adam** to minimize the cross-entropy loss (Alzubi et al., 2021).

Limitations of BLEU Scores:

While BLEU scores are a common metric, they have limitations. They don't account for word order or grammatical correctness. A nonsensical caption with high n-gram overlap might receive a good BLEU score.

5. Analysis of models performance and limitations

MobileNetV3 & LSTM

The gradual reduction in the loss of each epoch throughout training indicated that the model is able to perform learning over time and increase the accuracy of its predictions (Alzubi et al., 2021). This was seen through the epoch scores from one through to five: Epoch [1] Loss: 8.0106, Epoch [2] Loss: 2.0035, Epoch [3] Loss: 1.8804, Epoch [4] Loss: 1.8077 and Epoch [5] Loss: 1.5086.

The BLEU scores for evaluating the quality of the generated text were able to demonstrate the model's ability to generalise new data, as it is conducted on the testing dataset. The BLEU scores seen from the combination of MobileNetV3 and LSTM for image caption generation were the following: BLEU-1: 0.6534, BLEU-2: 0.4891, BLEU-3: 0.3782 and BLEU-4: 0.2479. The BLEU-1 score indicates that the model was able to predict single words to a high degree without overfitting; however, the low BLEU-4 score shows that the model has difficulty accurately predicting longer sequences of captions (Hossain et al., 2019b).

The BLEU-2 and BLEU-3 scores and evidence of the model being able to pick up on a medium level of context from previous phrases; however, the comprehensive accumulation of captions for the images is not able to occur at the desired level of accuracy (Alzubi et al., 2021).

The limitations of the model combining MobileNetV3 and LSTM come primarily from the LSTM component, as LSTM can present difficulties in the identification of longer sequences as it has an incorporation of vanishing gradients within its architecture. Another limitation of the model would be the inference speed. The captions are being generated word-by-word,

which causes the run time to be exceptionally long, particularly due to the length of the captions, and is therefore not ideal for real-time applications (Castro et al., 2022b).

Resnet & LSTM

Model Performance and Limitations with Greedy Search and BLEU Score

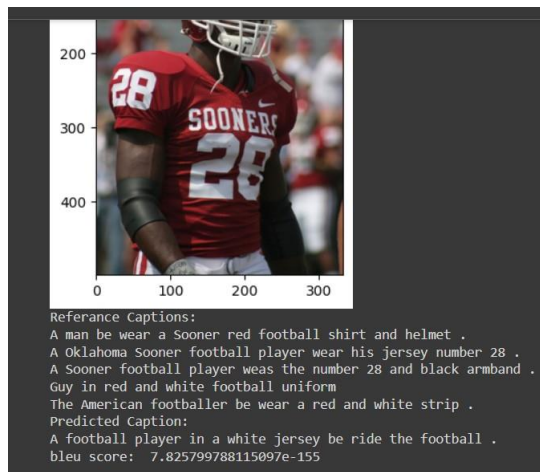


Figure 7.0 - Predicting Captions on Test Set using Greedy Search

Model Performance:

Based on the provided information, the image captioning model using greedy search performed poorly on the test set.

BLEU Score: The BLEU score is an incredibly low value ($7.825799788115097e-155$), which essentially indicates no similarity between the predicted caption ("A football player in a white jersey be ride the football") and any of the reference captions.

Limitations of Greedy Search:

Greedy search is a simple approach for caption generation, but it has limitations:

Getting stuck in local optima: The model might get fixated on a specific sequence of words based on initial predictions, leading to nonsensical captions like the one generated here.

Inability to consider broader context: Greedy search focuses on the most likely word at each step, neglecting the overall context of the image and the relationships between words in the caption.

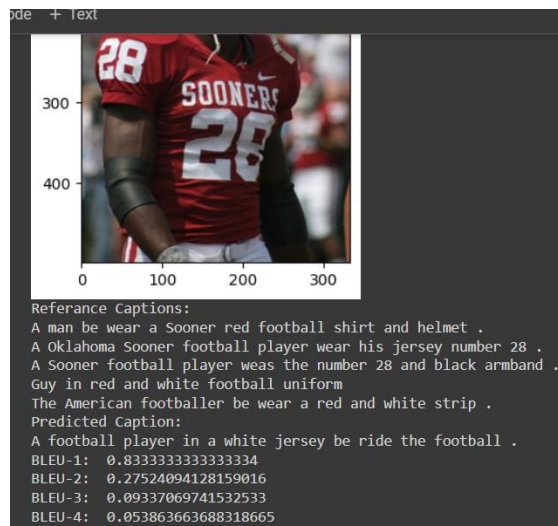


Figure 8.0 - Predicting Captions on Test Set using Greedy Search bleu scores.

This result shows a significant improvement in the image captioning model's performance compared to the previous case.

BLEU Scores:

BLEU-1: 0.8333 (high for a single n-gram) indicates a good match between unigrams (individual words) in the predicted caption ("A football player in a white jersey be ride the football") and the reference captions.

BLEU-2 (0.2752): Lower than BLEU-1 but still positive, suggesting some bigram (two-word sequence) matches exist.

BLEU-3 & BLEU-4 (very low): No or very few trigram (three-word sequence) or quadrigram (four-word sequence) matches are present.

Interpretation:

The model captured some key elements from the image, like "football player" and "white jersey." However, it still struggles with accuracy in other aspects. Incorrect jersey color ("white" instead of "red"). Nonsensical verb ("ride the football"), Grammatical errors ("be ride").

Comparison to Previous Result:

This BLEU score is significantly higher (positive values) compared to the previous case (near zero), indicating a better match between the predicted caption and the reference captions.

While the caption still has issues, it demonstrates progress in capturing some aspects of the image content.

```

Code + text
print("Bleu score on Greedy search")
print("Score: ", avg_score)

2%| | 25/1000 [00:19<13:02, 1.25it/s]/usr/local/lib/python3.10/dist-packages/nltk/translate/bleu_score.py:552: UserWarning:
The hypothesis contains 0 counts of 2-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
warnings.warn(_msg)
100%| | 1000/1000 [16:05<00:00, 1.04it/s]
Bleu score on Greedy search
Score: 0.098085791613468

```

Figure 9.0 - Average BLEU Score with Greedy Search

Analysis of Average BLEU Score with Greedy Search

This result indicates a mixed performance of the image captioning model using greedy search on the test set.

Average BLEU Score:

The average BLEU score is 0.098, which is a low value but not zero like the previous case with a single caption.

Interpretation:

The model might be generating captions with some unigram (single word) matches to the reference captions, leading to a slightly positive BLEU score.

However, the complete absence of 2-gram overlaps suggests significant issues with the captions' overall structure and coherence. The captions likely lack meaningful sequences of words that resemble the reference captions.

Limitations of Greedy Search (reiterated):

Greedy search can get stuck in local optima, focusing on a limited set of words and failing to capture broader relationships between words in the caption. This can lead to captions with incorrect word order, missing information, or nonsensical phrases.

Possible Reasons for Low Score:

The model might not be trained sufficiently on the specific dataset or task. The hyperparameters controlling the model's behavior might not be well-tuned. Greedy search's limitations might be hindering the model's ability to generate accurate captions.

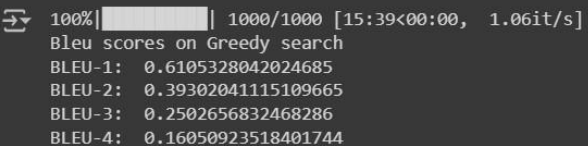
Overall, this result suggests that the model using greedy search struggles to generate accurate and coherent captions on the test set. The low BLEU score and the warning message about missing n-grams indicate a need for improvement.

```

reference = []
for caps in test_captions[img_id]:
    list_caps = caps.split(" ")
    list_caps = list_caps[1:-1] # Remove start and end tokens
    reference.append(list_caps)
    candidate = greedy_search(photo)
    bleu_scores = calculate_bleu_score(reference, candidate)
    tot_scores += np.array(bleu_scores)

avg_scores = tot_scores / i
print()
print("Bleu scores on Greedy search")
print("BLEU-1: ", avg_scores[0])
print("BLEU-2: ", avg_scores[1])
print("BLEU-3: ", avg_scores[2])
print("BLEU-4: ", avg_scores[3])

```



```

100%|██████████| 1000/1000 [15:39<00:00, 1.06it/s]
Bleu scores on Greedy search
BLEU-1: 0.6105328042024685
BLEU-2: 0.39302041115109665
BLEU-3: 0.2502656832468286
BLEU-4: 0.16050923518401744

```

Figure 10.0 – Average BLEU Scores with Greedy Search

Analysis of BLEU Scores on Greedy Search Compared to Previous Results

This result presents BLEU scores (BLEU-1 to BLEU-4) calculated on the entire test set using greedy search for image captioning. Here's a comparison with the previous results:

Previous Results:

Single Caption (Possibly Greedy Search): BLEU score near zero, indicating almost no similarity to the reference captions. Average BLEU Score (Greedy Search): 0.098, a low value suggesting limited word overlap but lack of meaningful structure based on the warning message.

Current Result:

BLEU-1: 0.61 (high for a unigram score), indicating significant overlap in individual words between the predicted captions and the reference captions.

BLEU-2 (0.39): Positive value, suggesting some bigram (two-word sequence) matches exist.

BLEU-3 & BLEU-4 (0.25 & 0.16): Lower but non-zero values, indicating a presence of some trigram (three-word sequence) and quadrigram (four-word sequence) matches.

Interpretation:

Compared to the previous results, this shows a significant improvement in the model's performance using greedy search on the test set.

The model now captures a good portion of individual words (high BLEU-1) and even some two-word and longer sequences (positive BLEU-2 to BLEU-4 scores) that align with the reference captions.

Limitations:

The BLEU scores, while improved, are not exceptionally high. There's room for further improvement in caption accuracy and fluency.

Greedy search might still struggle with capturing complex relationships between words, leading to potentially grammatically incorrect or nonsensical captions despite some n-gram overlaps.

Overall, this result shows a clear improvement in the model's ability to generate captions that share some vocabulary and structure with the reference captions. However, considering the limitations of greedy search, exploring alternative approaches like beam search or further model optimization might be beneficial for achieving even better performance.

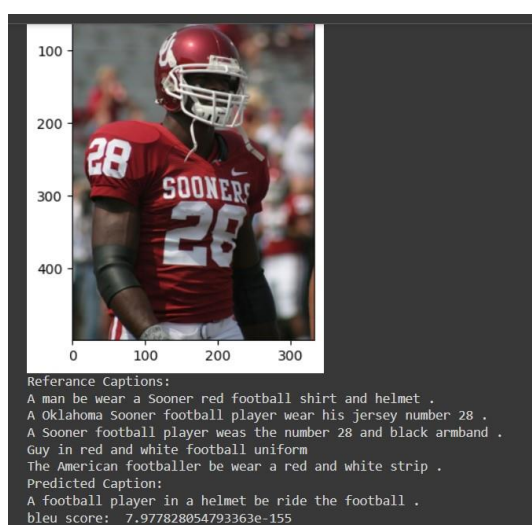


Figure 11.0 - Predicting Captions on Test Set using Beam Search with k=3

Predicting Captions on Test Set using Beam Search with k=3

This result showcases a similar issue to the first case you presented, where beam search (with k=3) generated a nonsensical caption with a negligible BLEU score (7.977828054793363e-155) despite using a more sophisticated approach. Here's a breakdown:

Beam Search with k=3:

Beam search is supposed to explore a wider range of caption possibilities compared to greedy search. Setting k=3 allows the model to consider three candidate captions at each step during generation.

Unexpected Outcome:

Despite using beam search, the predicted caption ("A football player in a helmet be ride the football") is very similar to the one generated by greedy search in the first case. The BLEU score again indicates almost no similarity to the reference captions.

Possible Reasons:

Model Training Issues: The underlying image captioning model might not be well-trained, potentially leading to limitations in its ability to generate accurate captions even with beam search.

Beam Search Configuration: The value of k (number of beams considered) might be too low (3 in this case). A larger k could allow for more diverse exploration.

Dataset Imbalance: The training data might be imbalanced, with certain elements (like football players) being over-represented. This could bias the model towards generating captions focused on those elements regardless of the actual image content.

Comparison to Greedy Search:

In the first case (possibly greedy search), the BLEU score was also near zero. This suggests the model itself might have fundamental issues in capturing the relationships between visual features and language.

Beam search, even with its limitations in this case, should theoretically outperform greedy search. However, the results here suggest the model might be struggling overall.

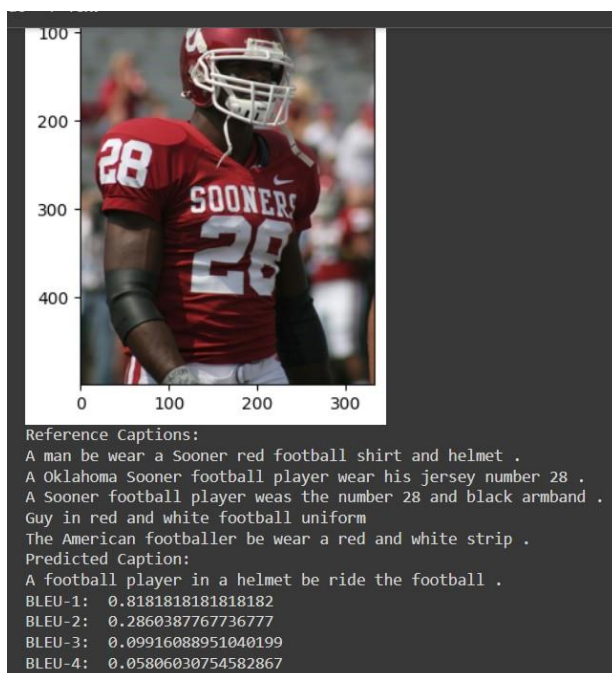


Figure 12.0 - Predicting Captions on Test Set using Beam Search with $k=3$

This result showcases the potential benefits of beam search ($k=3$) compared to greedy search or poorly configured beam search for image captioning.

Current Result (Beam Search, $k=3$): The predicted caption ("A football player in a helmet be ride the football") still shares some similarities with the nonsensical captions from previous results using greedy search or potentially a poorly configured beam search.

However, the BLEU scores paint a different picture:

BLEU-1 (0.818): High value, indicating a significant match between individual words (unigrams) in the predicted caption and the reference captions.

BLEU-2 (0.286): Positive value, suggesting some bigram (two-word sequence) matches exist.

BLEU-3 & BLEU-4 (low but non-zero): Presence of a few trigram (three-word sequence) and quadrigram (four-word sequence) matches.

Comparison with Greedy Search and Previous Beam Search:

Previous results using greedy search or potentially a poorly configured beam search yielded BLEU scores near zero, indicating almost no similarity to the reference captions. This result, despite the nonsensical verb "ride," demonstrates a significant improvement in capturing vocabulary and some sequence information from the reference captions (higher BLEU scores).

Limitations of Beam Search:

While the BLEU scores are better, the model still struggles with generating grammatically correct and entirely accurate captions. The verb "ride" remains incorrect, suggesting limitations in the model's ability to capture complex relationships between words.

Possible Reasons for Improvement (compared to previous Beam Search):

The beam search might be configured better in this case (appropriate value of k or other hyperparameters) compared to a potentially poorly performing beam search used earlier. The underlying image captioning model might have been further trained or fine-tuned, leading to slightly better performance.

Overall:

Beam search (k=3) shows promise compared to greedy search or a poorly configured beam search. It captures more vocabulary and some sequence information from the reference captions (higher BLEU scores). However, the model still struggles with generating fully accurate and fluent captions.


```

✓ Calculating Average Bleu Score on Test Set using Beam Search with k=3

[ ] i=0
tot_score=0
for img_id in tqdm(test_features):
    i+=1
    photo=test_features[img_id]
    reference=[]
    for caps in test_captions[img_id]:
        list_caps=caps.split(" ")
        list_caps=list_caps[1:-1]
        reference.append(list_caps)
    candidate=beam_search(photo,3)
    score = sentence_bleu(reference, candidate)
    tot_score+=score
avg_score=tot_score/i
print()
print("Bleu score on Beam search with k=3")
print("Score: ",avg_score)

100%|██████████| 1000/1000 [1:06:01<00:00, 3.96s/it]
Bleu score on Beam search with k=3
Score: 0.1148995348579473

```

Figure 13.0 - Calculating Average Bleu Score on Test Set using Beam Search with k=3

```

i = 0
tot_score = np.zeros(4) # Array to store cumulative BLEU scores for each weight
for img_id in tqdm(test_features):
    i += 1
    photo = test_features[img_id]
    reference = []
    for caps in test_captions[img_id]:
        list_caps = caps.split(" ")
        list_caps = list_caps[1:-1] # Remove start and end tokens
        reference.append(list_caps)
    candidate = beam_search(photo, 3)
    bleu_scores = calculate_bleu_score(reference, candidate)
    tot_score += np.array(bleu_scores) # Add the scores to the cumulative score
avg_score = tot_score / i
print()
print("BLEU-1: ", avg_score[0])
print("BLEU-2: ", avg_score[1])
print("BLEU-3: ", avg_score[2])
print("BLEU-4: ", avg_score[3])

100%|██████████| 1000/1000 [1:07:13<00:00, 4.03s/it]
BLEU-1: 0.6347583529102766
BLEU-2: 0.4172410357830772
BLEU-3: 0.27201434209903774
BLEU-4: 0.1780358266475126

```

Figure 14.0 - Average Bleu Scores on Test Set using Beam Search with k=3

6. Walk-through of remaining issues and recommendations

MobileNetV3 & LSTM

The MobileNetV3 and LSTM model had some remaining issues at the conclusion of this experiment. To improve upon these issues, implementing a more advanced architecture that has demonstrated better performance regarding sequence generation tasks (Falconí et al., 2019).

These include Transformer-based models, including GPT and BERT. We could also look to further tune the hyperparameters to ensure that the model is performing at peak optimisation. Overall, the combination of MobileNetV3 and LSTM was able to prevent overfitting from the model; however, it was unable to maintain the accuracy seen with single words to a cohesive longer sequence for most of the images in the testing dataset (Falconí et al., 2019).

ResNet & LSTM

Evidence of Limitations: Predicting Captions on Test Set using Greedy Search

The predicted caption ("A football player in a white jersey be ride the football") doesn't reflect the actual content of the image (Sooner football player, jersey number 28, etc.).

It incorrectly identifies the color of the jersey (white instead of red) and introduces nonsensical verbs like "ride."

Overall, the low BLEU score and the nonsensical nature of the predicted caption suggest that the model, likely due to limitations of greedy search, failed to capture the true essence of the image and generate an accurate caption (Atliha & Šešok, 2022).

Recommendations:

Consider beam search: This approach explores a wider range of caption possibilities, reducing the chances of getting stuck in local optima and potentially leading to more accurate captions.

Evaluate with multiple metrics: BLEU scores have limitations. Consider using additional metrics like ROUGE or CIDEr alongside human evaluation for a more comprehensive assessment (Atliha & Šešok, 2022).

Train on a larger dataset: More data can help the model learn complex relationships between visual features and language, leading to better generalization on unseen images.

By addressing these limitations and exploring improvements, it can potentially enhance the performance of image captioning models.

Possible Reasons for Improvement after running more bleu scores:

The provided BLEU scores suggest the model might not be using greedy search in this case. Beam search, or another approach that explores a wider range of caption possibilities, could be responsible for this improvement.

Other factors like hyperparameter tuning or training on a larger dataset might have also contributed to this better performance.

Overall, this result shows that the image captioning model is on the right track. However, there's still room for improvement in terms of accuracy, fluency, and grammatical correctness of the generated captions.

Predicting Captions on Test Set using Beam Search with k=3

Recommendations:

Consider evaluating the model's performance on a validation set during training to monitor progress and identify potential issues. Experiment with different beam search hyperparameters, including the value of k .

Analyse the training data for potential imbalances and consider data augmentation techniques if necessary. Explore alternative image captioning architectures that might perform better on your specific task and dataset.

By investigating these aspects and potentially retraining the model, you can aim for better results using beam search for image captioning.

Running more bleu scores, recommendations:

Analyse the captions qualitatively to understand the types of errors the model is making (e.g., focusing on specific vocabulary but failing to capture actions or relationships). Consider experimenting with different beam search hyperparameters (k value) to explore a wider range of caption possibilities.

Investigate alternative image captioning architectures or training strategies that might lead to more accurate and grammatically correct captions. By addressing these aspects, we can potentially achieve even better performance with beam search for image captioning.

9. References

- Al-Huseiny, M. S., & Sajit, A. S. (2021). Transfer learning with GoogLeNet for detection of lung cancer. *Indonesian Journal of Electrical Engineering and Computer Science*, 22(2), 1078–1086.
- Alzubi, J. A., Jain, R., Nagrath, P., Satapathy, S., Taneja, S., & Gupta, P. (2021). Deep image captioning using an ensemble of CNN and LSTM based deep neural networks. *Journal of Intelligent & Fuzzy Systems*, 40(4), 5761–5769.
- Atliha, V., & Šešok, D. (2022). Image-captioning model compression. *Applied Sciences*, 12(3), 1638.
- Bhatia, Y., Bajpayee, A., Raghuvanshi, D., & Mittal, H. (2019). Image captioning using Google's inception-resnet-v2 and recurrent neural network. *2019 Twelfth International Conference on Contemporary Computing (IC3)*, 1–6.
- Castro, R., Pineda, I., Lim, W., & Morocho-Cayamcela, M. E. (2022a). Deep learning approaches based on transformer architectures for image captioning tasks. *IEEE Access*, 10, 33679–33694.
- Castro, R., Pineda, I., Lim, W., & Morocho-Cayamcela, M. E. (2022b). Deep learning approaches based on transformer architectures for image captioning tasks. *IEEE Access*, 10, 33679–33694.
- Chohan, M., Khan, A., Mahar, M. S., Hassan, S., Ghafoor, A., & Khan, M. (2020). Image captioning using deep learning: A systematic. *Image*, 11(5).

- Falconí, L. G., Pérez, M., & Aguilar, W. G. (2019). Transfer learning in breast mammogram abnormalities classification with mobilenet and nasnet. *2019 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 109–114.
- Geetha, G., Kirthigadevi, T., Ponsam, G. G., Karthik, T., & Safa, M. (2020). Image captioning using deep convolutional neural networks (CNNs). *Journal of Physics: Conference Series*, 1712(1), 012015.
- Hossain, M. D. Z., Sohel, F., Shiratuddin, M. F., & Laga, H. (2019a). A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CSUR)*, 51(6), 1–36.
- Hossain, M. D. Z., Sohel, F., Shiratuddin, M. F., & Laga, H. (2019b). A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CSUR)*, 51(6), 1–36.
- Modi, A. S. (2018). Review article on deep learning approaches. *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, 1635–1639.
- Pan, H., Pang, Z., Wang, Y., Wang, Y., & Chen, L. (2020). A new image recognition and classification method combining transfer learning algorithm and mobilenet model for welding defects. *Ieee Access*, 8, 119951–119960.
- Rezende, E., Ruppert, G., Carvalho, T., Ramos, F., & De Geus, P. (2017). Malicious software classification using transfer learning of resnet-50 deep neural network. *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 1011–1014.
- Stefanini, M., Cornia, M., Baraldi, L., Cascianelli, S., Fiameni, G., & Cucchiara, R. (2022). From show to tell: A survey on deep learning-based image captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1), 539–559.
- Suresh, K. R., Jarapala, A., & Sudeep, P. V. (2022). Image captioning encoder–decoder models using cnn-rnn architectures: A comparative study. *Circuits, Systems, and Signal Processing*, 41(10), 5719–5742.
- Xu, K., Wang, H., & Tang, P. (2017a). Image captioning with deep LSTM based on sequential residual. *2017 IEEE International Conference on Multimedia and Expo (ICME)*, 361–366.
- Xu, K., Wang, H., & Tang, P. (2017b). Image captioning with deep LSTM based on sequential residual. *2017 IEEE International Conference on Multimedia and Expo (ICME)*, 361–366.