# DAT301m Lab 3: Natural Language Processing, Transformer and Multimodal Learnings

**Trieu Vuong Nguyen**
AI Development with TensorFlow
FPT University
Ho Chi Minh City, Vietnam
vuongntse182185@fpt.edu.vn

## 1 Introduction

This lab explores deep learning techniques in Natural Language Processing (NLP) and multimodal learning by combining text and image data. Using the Flickr8k dataset, we perform tasks such as caption classification, translation, and image captioning.

We begin with traditional models such as as RNNs and LSTMs, incorporate attention mechanisms, and later fine-tune transformer-based models for performance comparison. The lab aims to build practical understanding of tokenization, sequence modeling, attention, and the use of pre-trained transformers in both NLP and vision-language tasks.

## 2 Dataset

The primary dataset used in this lab is Flickr8k, which consists of more than 8,000 images, each paired with five human-written captions describing the content of the image. The images cover a wide range of everyday scenes and were collected from various Flickr groups, avoiding well-known individuals or locations to maintain generality.

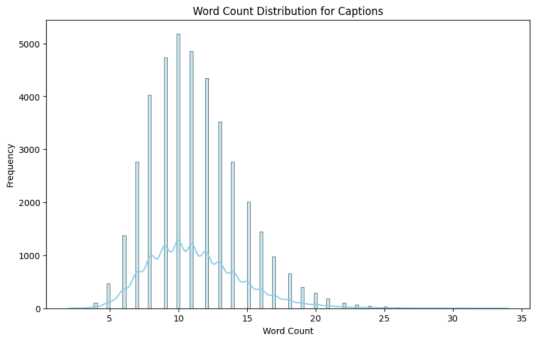
For the translation task, we used a translated version of Flickr8k captions in Japanese, obtained from publicly available Kaggle datasets. These translated captions were aligned with the English captions to create bilingual caption pairs for sequence-to-sequence modeling.

Each image-caption pair supports multiple tasks:

- Text classification (based on caption length),
- Sequence-to-sequence translation (non-English $\rightarrow$ English),
- Image captioning (image $\rightarrow$ English caption).

The dataset provides a rich base for exploring NLP, CV, and multimodal applications.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).




(a) Word Count Distribution for Captions


(b) Word Cloud for All Captions

Figure 1: Visualization of the dataset distribution

## 3 Caption Classification

### 3.1 Data Preprocessing

- **Text normalization**: Captions were lowercased, non-alphanumeric characters were removed, and `<start>` and `<end>` tokens were added.
- **Tokenization and padding**: Cleaned captions were tokenized into word indices using a custom vocabulary with out-of-vocabulary token handling. Sequences were padded to the length of the longest caption.
- **Length-based labeling**: Captions were labeled as *short* (less than 15 words), *medium* (15–20 words), or *long* (more than 20 words) based on word count.
- **Label encoding and data splitting**: Length labels were converted into integers, and the dataset was split into training and testing sets using an 80/20 ratio.

### 3.2 RNN + LSTM

The model is a sequential neural network that combines SimpleRNN and LSTM layers to classify captions based on their length (short, medium, long). The architecture is designed to capture both short-term and long-term dependencies in the tokenized caption sequences.

- **Embedding layer**: Maps input word indices to 256-dimensional dense vectors.
- **RNN + LSTM layers**: A SimpleRNN layer followed by an LSTM layer (both with 256 units) captures both short- and long-term dependencies.
- **Regularization**: Batch normalization and 20% dropout are applied to improve generalization.
- **Global average pooling**: Reduces the sequence output to a fixed-length vector by averaging over time steps.
- **Output layer**: A softmax dense layer outputs probabilities for three caption length classes.
- **Loss and optimization**: The model uses sparse categorical crossentropy and the Adam optimizer with learning rate 0.001 and gradient clipping.
- **Callbacks**: Early stopping and learning rate reduction on plateau are used for efficient training.
- **Training**: Trained for up to 20 epochs with a batch size of 32 using an 80/20 train-validation split.

### 3.3 GRU + BiLSTM

The model is a sequential neural network that combines a GRU layer with a Bidirectional LSTM to classify captions based on their length (short, medium, long). The architecture is designed to capture contextual dependencies in both directions for richer sequence understanding.

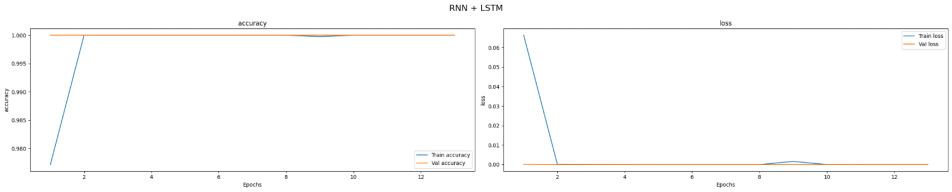
- **Embedding layer**: Maps input word indices to 256-dimensional dense vectors.
- **GRU + Bidirectional LSTM layers**: A GRU layer followed by a Bidirectional LSTM layer (each with 256 units) captures both forward and backward temporal patterns in the caption sequences.
- **Regularization**: Batch normalization and 20% dropout are applied to improve generalization.
- **Global average pooling**: Reduces the sequence output to a fixed-length vector by averaging over time steps.
- **Dense layer**: A fully connected layer with 256 units and ReLU activation refines the pooled features.
- **Output layer**: A softmax dense layer outputs probabilities for three caption length classes.
- **Loss and optimization**: The model uses sparse categorical crossentropy and the Adam optimizer with learning rate 0.001 and gradient clipping.
- **Callbacks**: Early stopping and learning rate reduction on plateau are used for efficient training.
- **Training**: Trained for up to 20 epochs with a batch size of 32 using an 80/20 train-validation split.

### 3.4 DistilBERT

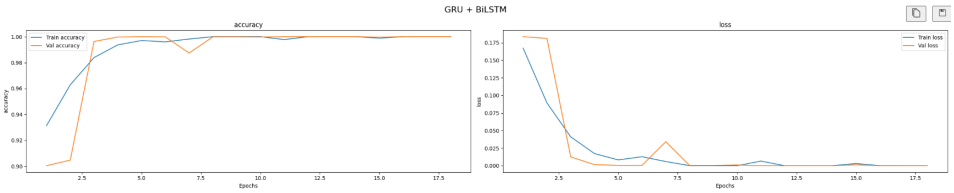
The model uses a pre-trained DistilBERT transformer for caption length classification (short, medium, long). It is fine-tuned to leverage deep contextual representations learned from large-scale text corpora.

- **Tokenizer and input preparation**: Input captions are tokenized using DistilBERT's tokenizer with padding and truncation to a fixed sequence length.
- **Pre-trained transformer model**: A DistilBERT base model is fine-tuned with a classification head for predicting three caption length categories.
- **Dataset batching**: Tokenized inputs and labels are organized into TensorFlow datasets with dynamic padding and batching.
- **Loss and optimization**: The model is optimized using sparse categorical crossentropy and the Adam optimizer with a learning rate of $5 \times 10^{-5}$.
- **Callbacks**: Early stopping and learning rate scheduling are applied to enhance convergence.
- **Training**: Fine-tuned for up to 10 epochs with evaluation after each epoch using an 80/20 train-validation split.

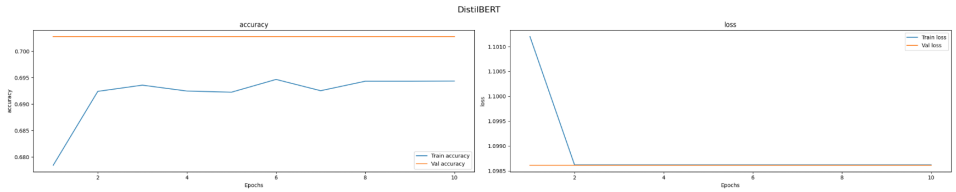
### 3.5 Evaluation




(a) RNN + LSTM Loss and Accuracy




(b) GRU + BiLSTM Loss and Accuracy




(c) DistilBERT Loss and Accuracy

Figure 2: Training and Validation Metrics

The performance of the three models—**RNN + LSTM**, **GRU + BiLSTM**, and **DistilBERT**—was compared based on training/validation accuracy, loss trends, and generalization ability. Below is a summary of the observations:

- **RNN + LSTM**: The model quickly achieved near-perfect training and validation accuracy. However, the flat and overly high validation metrics suggest potential overfitting or data leakage, rather than genuine generalization.
- **GRU + BiLSTM**: Demonstrated steady improvement over epochs with realistic learning curves. Achieved high accuracy with slight fluctuations in validation loss, indicating good generalization and effective learning.
- **DistilBERT**: The transformer-based model underperformed, with both training and validation accuracy plateauing early around 70%. The nearly flat loss curves indicate underfitting or ineffective fine-tuning.

| Model | Train Acc. | Val Acc. | Stability | Generalization | Remarks |
|---|---|---|---|---|---|
| RNN + LSTM | 100% | ~99.9% | Flat | Low | Overfit or data leakage |
| GRU + BiLSTM | 93–100% | 90–99% | Stable | High | Best overall performance |
| DistilBERT | 68–69% | ~71% | Flat | Low | Underfitting; needs tuning |

Table 1: Summary of model performance and behavior

The performance of the three models is compared based on accuracy and F1 score:

| Model | Accuracy | F1 Score | Remarks |
|---|---|---|---|
| RNN + LSTM | 1.00 | 1.00 | Very high; possible overfitting |
| GRU + BiLSTM | 1.00 | 1.00 | Strong performance; may also overfit |
| DistilBERT | 0.263 | 0.168 | Underfitting; needs tuning |

Table 2: Performance comparison of models on caption classification

### 3.6 Inference

| Caption (shortened) | RNN+LSTM | GRU+BiLSTM | DistilBERT |
|---|---|---|---|
| Large cat pursuing horse | ✓ | ✓ | (Pred: 1) |
| Two brown dogs fighting | ✓ | ✓ | ✓ |
| Man standing on hilltop | ✓ | ✓ | (Pred: 1) |
| Muzzled dog running | ✓ | ✓ | (Pred: 1) |
| Person skiing downhill | ✓ | ✓ | (Pred: 2) |

Table 3: Comparison of model predictions on sample captions (✓= correct,  = incorrect)

### 3.7 Question 1

Overall, the **GRU + BiLSTM** model performed the best. It achieved perfect accuracy and F1 score on the validation set, and correctly classified all tested samples, showing strong generalization without signs of overfitting.

The **RNN + LSTM** model also achieved perfect scores, but its flat validation metrics suggest possible overfitting—likely memorizing patterns rather than learning general features.

In contrast, the **DistilBERT** model showed signs of **underfitting**, with significantly lower accuracy (26.3%) and F1 score (16.8%). Despite being a pre-trained transformer, it struggled to learn effectively from the available data, possibly due to insufficient fine-tuning or data size.

Comparing architectures:

- The combination of **GRU and BiLSTM** layers helped the model capture both past and future context, improving its ability to generalize across caption lengths.
- Simpler models like **RNN + LSTM** learned well but were more prone to overfitting without additional regularization or bidirectional context.

In summary, models with richer temporal features (e.g., GRU + BiLSTM) outperformed both simpler recurrent models and transformer-based models under current training conditions.

## 4 Caption Translation

### 4.1 Data Preprocessing

- **Text normalization:** English captions were lowercased, cleaned of non-alphanumeric characters, and wrapped with `<start>` and `<end>` tokens. Japanese captions were tokenized using a morphological tagger and similarly wrapped.
- **Tokenization and padding:** Captions were tokenized and converted into sequences using custom tokenizers. English captions were split into decoder input and target sequences. All sequences were padded to the longest length using post-padding.
- **Length-based labeling:** *Not applicable* in this translation-focused preprocessing pipeline.
- **Data splitting:** Tokenized sequences were split into training and testing sets (80/20) for both encoder (Japanese) and decoder (English) inputs and targets.

### 4.2 LSTM

This model is a sequence-to-sequence architecture designed for Japanese-to-English translation. It uses an encoder-decoder structure with LSTM layers to learn dependencies across input and output sequences.

- **Encoder:** Processes Japanese input with an Embedding layer and an LSTM (256 units), producing hidden and cell states for the decoder.
- **Decoder:** Uses English inputs with an Embedding and LSTM layer (256 units), initialized from the encoder. Outputs token predictions via a Dense softmax layer.
- **Output layer:** Projects decoder outputs to the English vocabulary using softmax.
- **Loss and optimization:** Trained with sparse categorical crossentropy and Adam (lr = 0.001, clipvalue = 1.0).

- **Callbacks:** Early stopping (patience = 5) and learning rate reduction (patience = 2, factor = 0.5, min LR = 1e–6).
- **Training:** Up to 20 epochs, batch size 32, using an 80/20 train-validation split.

### 4.3 GRU + BiLSTM - GRU + LSTM - Additive Attention

This model is a hybrid encoder-decoder architecture that combines GRU, Bidirectional LSTM, and Additive Attention to enhance Japanese-to-English translation by capturing rich contextual information from both directions and aligning input-output dependencies.

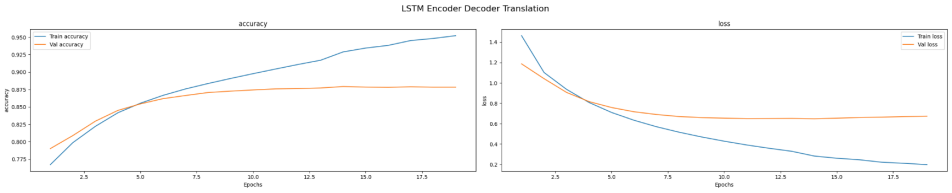
- **Encoder:** Processes Japanese tokens via an Embedding layer, a GRU, and a Bidirectional LSTM (256 units each direction). Forward and backward states are concatenated.
- **Decoder:** Takes English input, passes through an Embedding, a GRU (256 units), and an LSTM (512 units) initialized with encoder states.
- **Attention:** Additive Attention computes context vectors from encoder outputs and decoder states, which are concatenated with decoder outputs.
- **Output layer:** A softmax Dense layer predicts English token probabilities.
- **Loss and optimization:** Trained with sparse categorical crossentropy and Adam (lr = 0.001, clipvalue = 1.0).
- **Callbacks:** Early stopping (patience = 5) and LR reduction (patience = 2, factor = 0.5, min LR = 1e–6).
- **Training:** Trained up to 20 epochs, batch size 32, with 80/20 train-validation split.

### 4.4 MarianMT

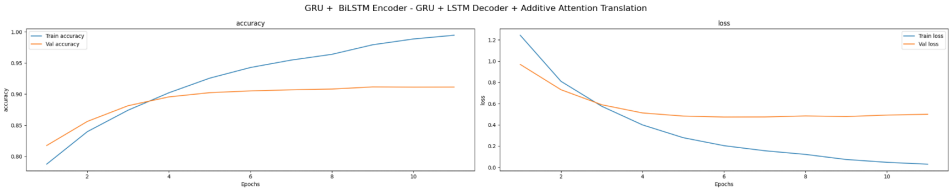
This model uses a pretrained transformer-based MarianMT architecture from Hugging Face for Japanese-to-English translation. It leverages large-scale multilingual data and transfer learning to achieve effective translation performance with minimal training.

- **Tokenizer:** Uses `Helsinki-NLP/opus-mt-ja-en` to tokenize and pad/truncate both Japanese and English captions (max length = 64).
- **Model:** A pretrained transformer-based encoder-decoder model (`TFMarianMTModel`) used without modification.
- **Loss and optimization:** Trained with sparse categorical crossentropy and Adam (learning rate = $3 \times 10^{-5}$).
- **Data pipeline:** Uses Hugging Face Datasets and `DataCollatorForSeq2Seq` for batching and padding. Split 80/20 for train/test.
- **Training:** Fine-tuned for 10 epochs, batch size 32, with validation after each epoch.

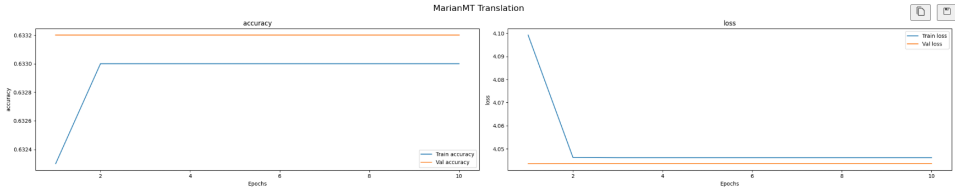
### 4.5 Evaluation




(a) LSTM Loss and Accuracy




(b) GRU + BiLSTM - GRU + BiLSTM + Additive Attention Loss and Accuracy




(c) MarianMT Loss and Accuracy

Figure 3: Training and Validation Metrics

The performance of the three models—**LSTM Encoder-Decoder**, **GRU + BiLSTM + Attention**, and **MarianMT**—was compared based on training/validation accuracy, loss trends, and generalization behavior. Below is a summary of the observations:

- **LSTM Encoder-Decoder**: Showed a strong upward trend in training accuracy, reaching above 95%, while validation accuracy plateaued early and stayed around 89%. The gap between training and validation curves suggests mild overfitting. Validation loss flattened after early improvement, confirming this behavior.
- **GRU + BiLSTM + Attention**: Delivered excellent results, with both training and validation accuracy improving consistently, and only a small generalization gap. Validation loss remained low and stable, indicating robust learning and good generalization performance. This model achieved the best balance overall.
- **MarianMT**: Performance plateaued very early in both accuracy and loss. Training and validation curves remained flat around 63% accuracy and 4.05 loss, indicating clear underfitting or ineffective fine-tuning of the pretrained model on the current dataset.

| Model | Train Acc. | Val Acc. | Stability | Generalization | Remarks |
|---|---|---|---|---|---|
| LSTM Encoder-Decoder | 77–95% | 78–89% | Moderate | Medium | Signs of overfitting after early epochs |
| GRU + BiLSTM + Attention | 79–99% | 82–92% | Stable | High | Best performance overall |
| MarianMT | ~63% | ~63% | Flat | Low | Underfitting; minimal improvement |

Table 4: Summary of model performance and behavior

The performance of the three models—**LSTM Encoder-Decoder**, **GRU + BiLSTM + Attention**, and **MarianMT**—was compared based on training/validation accuracy, loss curves, generalization ability, and BLEU score. Below is a summary of the observations:

- **LSTM Encoder-Decoder**: The model trained for 19 epochs before early stopping. It showed strong training performance but signs of overfitting with a modest validation plateau. The BLEU score of 0.3336 (on 1000 samples) indicates moderate translation quality.
- **GRU + BiLSTM + Attention**: Early stopping occurred after 11 epochs. This model demonstrated stable learning and better generalization. It achieved the highest BLEU score of 0.4468 (on 1000 samples), confirming its superior translation quality.
- **MarianMT**: Training and validation metrics remained flat, with a BLEU score of 0.0 (on 100 samples), indicating complete underfitting. The model failed to adapt to the dataset, likely due to insufficient fine-tuning or dataset size mismatch.

| Model | Train Acc. | Val Acc. | BLEU | Early Stop | Gen. | Remarks |
|---|---|---|---|---|---|---|
| LSTM Encoder-Decoder | 77–95% | 78–89% | 0.3336 | 19 | Medium | Mild overfitting; decent quality |
| GRU + BiLSTM + Attention | 79–99% | 82–92% | **0.4468** | 11 | High | Best generalization and BLEU |
| MarianMT | ~63% | ~63% | 0.0 | 10 | Low | Underfitting; poor adaptation |

Table 5: Model performance comparison including BLEU scores and early stopping

### 4.6 Inference

| Japanese Caption (shortened) | LSTM | GRU+BiLSTM+Attn | MarianMT |
|---|---|---|---|
| Boy playing in blue shirt | ✓ | ✓ | (Irrelevant output) |
| Young man running in park | (Incorrect phrase) | ✓ | (Incoherent) |
| Girl running with dog | ✓ | ✓ | (Unrelated) |
| Older man taking photo | (Wrong action) | ✓ | (Unrelated) |
| Woman flying kite | (Wrong activity) | (Odd phrasing) | (Unrelated) |

Table 6: Comparison of model translation outputs on sample Japanese captions (✓= semantically correct,  = incorrect or irrelevant)

### 4.7 Question 2

The **GRU + BiLSTM + Attention** model produced the most accurate and fluent translations. It successfully preserved the meaning of the Japanese captions and demonstrated strong generalization with the highest BLEU score (0.4468). Most test translations were semantically correct, even for longer or more complex sentences.

The **LSTM Encoder-Decoder** also performed reasonably well, achieving a BLEU score of 0.3336. However, some outputs were grammatically awkward or slightly off in meaning, especially for more nuanced inputs. This suggests limited context handling without an attention mechanism.

In contrast, the **MarianMT** model failed to generate coherent or relevant translations. It consistently produced unrelated, repetitive outputs and scored a BLEU of 0.0, indicating poor adaptation to the dataset.

Regarding Attention:

- The **Additive Attention** in the GRU + BiLSTM model helped align the decoder with relevant parts of the input sequence, improving translation quality and context retention.
- The absence of attention in the LSTM model made it harder to maintain context over long sequences, leading to occasional semantic drift.

In summary, attention mechanisms significantly enhanced translation quality, with the GRU + BiLSTM + Attention model outperforming both non-attentional and pretrained models under the given training conditions.

# 5 Image Captioning

## 5.1 Data Preprocessing

- **Text cleaning and tokenization**: Captions were lowercased, cleaned of non-alphabetic characters and short words, and wrapped with `sos` and `eos` tokens. A tokenizer was fitted to generate word indices and compute vocabulary size and maximum caption length.
- **Image splitting**: Unique image filenames were split into training and testing sets using an 80/20 ratio.
- **Feature extraction**: InceptionV3 (pretrained on ImageNet) was used to extract image features from the second-to-last layer. Images were resized to $299 \times 299$ and normalized before feature extraction.
- **Sequence generation**: A custom data generator was built to produce image-caption pairs. For each caption, partial sequences were padded, and the next word was predicted using one-hot encoding. Each batch includes image features, input sequences, and target words.

## 5.2 LSTM

This model generates image captions using a combination of image and text inputs, processed through LSTM layers in a multimodal architecture.

- **Inputs:** Takes two inputs — a 2048-dimensional image feature vector and a padded caption sequence.
- **Image path:** Image features are passed through a Dense (256 units, ReLU) and reshaped to match the sequence dimension.
- **Text path:** Caption inputs are embedded into 256-dimensional vectors (no masking) and concatenated with image features.
- **Sequence modeling:** The merged sequence is processed by an LSTM layer (256 units), followed by dropout and residual addition with the image embedding.
- **Output layer:** Final Dense layer (softmax) predicts the next word over the full vocabulary.
- **Loss/Optimization:** Trained with categorical crossentropy and Adam (lr = 0.001, clipvalue = 1.0).
- **Callbacks:** Early stopping (patience = 10) and learning rate reduction (patience = 2, factor = 0.5, min LR = 1e–6).
- **Training:** Trained for up to 50 epochs with image-caption pairs from the custom data generators.

## 5.3 LSTM + Attention

This model extends the LSTM-based architecture by incorporating an attention mechanism, enabling better focus on relevant sequence parts when generating image captions.

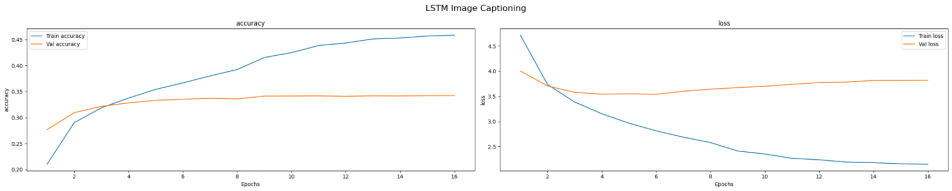
- **Inputs:** Accepts a 2048-dimensional image feature vector and a caption sequence.
- **Image path:** Image features are passed through a Dense (256 units, ReLU) and reshaped for sequence compatibility.
- **Text path:** Captions are embedded into 256-dimensional vectors and concatenated with image features to form a single sequence.
- **Sequence modeling:** The combined sequence is processed by an LSTM layer (256 units, return sequences enabled).
- **Attention mechanism:** An attention layer uses the image vector as a query over the LSTM outputs to refine context. The output is combined with the original image embedding.
- **Output layer:** The fused representation passes through Dense and Dropout layers before final prediction over the vocabulary using softmax.
- **Loss/Optimization:** Trained using categorical crossentropy and Adam (lr = 0.001, clipvalue = 1.0).
- **Training:** Trained for up to 50 epochs with early stopping and learning rate scheduling. Final model and training history were saved for evaluation.

### 5.4 ViT-GPT2

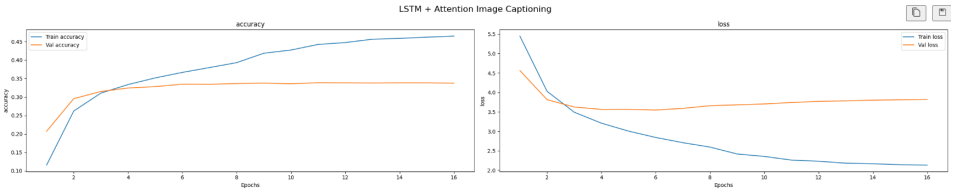
This model combines a Vision Transformer (ViT) encoder with a GPT-2 decoder for image captioning using the HuggingFace Transformers framework.

- **Architecture:** Uses a pre-trained ViT (`vit-base-patch16-224`) as the image encoder and GPT-2 as the language decoder in a VisionEncoderDecoderModel setup.
- **Preprocessing:** Images are resized to $224 \times 224$ and processed with ViTFeatureExtractor. Captions are tokenized with the GPT-2 tokenizer, padded to fixed length (50), and masked using `-100` for loss computation.
- **Dataset:** Custom PyTorch `Dataset` loads image-caption pairs. Pixel values and tokenized labels are returned per sample.
- **Decoder Configuration:** Decoder uses beam search (4 beams), early stopping, a length penalty of 2.0, and prevents n-gram repetition.
- **Training Setup:** Trained using HuggingFace's `Seq2SeqTrainer` with a batch size of 8, learning rate $5 \times 10^{-5}$, warmup, and checkpointing. The model was trained for 10 epochs with both training and validation monitoring.
- **Callbacks:** Includes model saving, logging, and automatic evaluation per epoch using `predict_with_generate=True`.

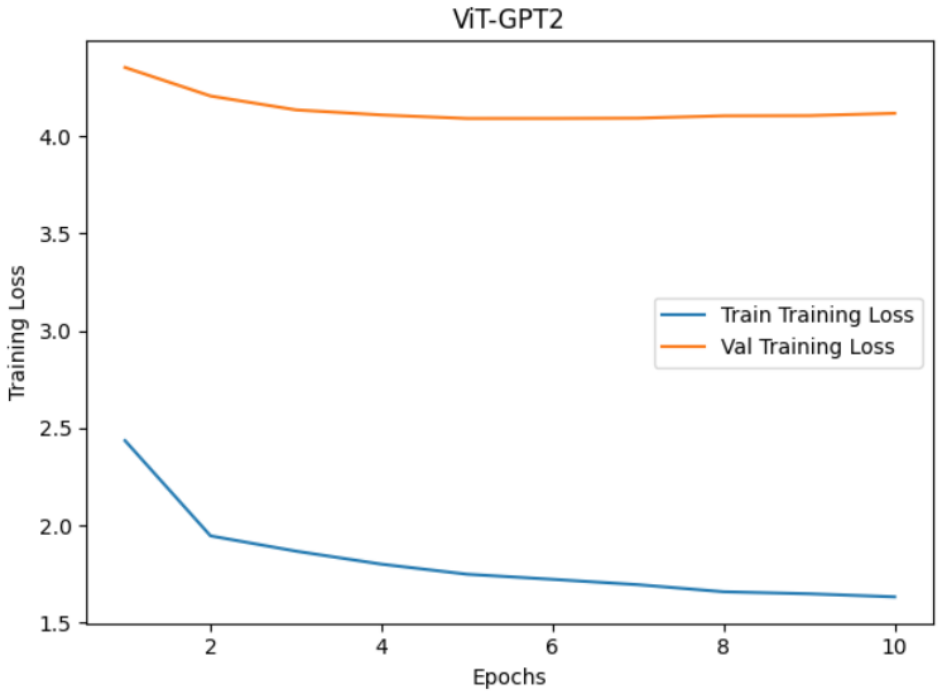
### 5.5 Evaluation




(a) LSTM Loss and Accuracy




(b) LSTM + Attention Loss and Accuracy




(c) ViT-GPT2 Loss

Figure 4: Training and Validation Metrics

The performance of the three models—**LSTM**, **LSTM + Attention**, and **ViT-GPT2**—was assessed based on accuracy, loss behavior, and generalization performance. Below is a summary of the observations:

- **LSTM Image Captioning**: Showed steady growth in training accuracy from 21% to 46%, while validation accuracy peaked early at around 34% and then plateaued. Training loss decreased consistently, but validation loss stagnated and slightly increased, indicating mild overfitting.
- **LSTM + Attention Image Captioning**: Achieved similar trends to the base LSTM model but with slightly better generalization. Training accuracy reached 46% and validation accuracy stabilized around 34% as well. Loss curves show improved convergence over the plain LSTM, but generalization gap still remains.

- **ViT-GPT2**: Demonstrated stable and gradual improvement in training loss over 10 epochs, with validation loss decreasing initially but flattening early. No accuracy metric was provided, but the consistent gap between training and validation loss suggests underfitting or limited adaptation to the dataset.

| Model | Train Acc. | Val Acc. | Stability | Generalization | Remarks |
|---|---|---|---|---|---|
| LSTM | 21–46% | 28–34% | Moderate | Medium | Overfitting after early epochs |
| LSTM + Attention | 12–46% | 21–34% | Improved | Medium | Better than plain LSTM, but still a gap |
| ViT-GPT2 | — | — | Flat Loss | Low | No overfitting, but likely underfitting |

Table 7: Summary of model performance and behavior

The performance of the three models—**LSTM**, **LSTM + Attention**, and **ViT-GPT2**—was evaluated using training/validation metrics and BLEU score analysis. Below is a summary of observations:

- **LSTM**: This model showed moderate learning behavior with training accuracy reaching 46% and validation accuracy plateauing near 34%. The BLEU score on 100 test samples was **0.0138**, suggesting limited caption accuracy. Model performance indicates overfitting due to a noticeable generalization gap.

- **LSTM + Attention**: Adding attention led to slightly smoother convergence and similar accuracy patterns as the base LSTM model. However, BLEU score improved significantly to **0.0502**, showing that attention helped generate more contextually relevant captions even though the validation accuracy remained around 34%.

- **ViT-GPT2**: Despite using a powerful pretrained vision-language transformer, the model failed to adapt effectively to the dataset. It generated nonsensical captions filled with repeated words and tokens. The BLEU score was only **0.00005**, indicating **severe underfitting**. Likely causes include insufficient training epochs, tokenization issues, or mismatch in fine-tuning procedure.

| Model | Train Acc. | Val Acc. | BLEU | Generalization | Remarks |
|---|---|---|---|---|---|
| LSTM | 21–46% | 28–34% | 0.0138 | Medium | Some overfitting; captions partially relevant |
| LSTM + Attention | 12–46% | 21–34% | **0.0502** | Medium | Better BLEU despite similar accuracy |
| ViT-GPT2 | — | — | 0.00005 | Low | Underfitting; repetitive, irrelevant captions |

Table 8: Model performance based on accuracy, BLEU score, and generalization

## 5.6 Inference


(a) LSTM Image Captioning


(b) LSTM + Attention Image Captioning

(c) ViT-GPT2 Image Captioning

(d) man a and dog on beach sunset sunset sunset trees background . in . . . of . . ocean sunset sunset clouds clouds clouds . sky sky sky clouds clouds water . . sky . clouds clouds sky sky . . with clouds clouds sunset sky . sky background clouds water sky . with in . sky skies sky . sunset sky sky background sky sky with clouds water background sky . skies clouds sky . people in . water the . sunset sunset sky sunset sky people water . people water in . sunset . sky water background people water sunset people water water sunset sunset people sky sky water sunset . people sky water sky sky sunset people . water sky background people sky

### 5.7 Question 3

The **LSTM + Attention** model performed well in generating captions, producing fluent and generally grammatically correct results. It successfully described the content of images, with most captions being coherent and contextually relevant. However, some generated captions exhibited minor errors, such as object misidentifications and vague descriptions, which occasionally reduced their accuracy. For instance, captions like "sos two children playing with soccer ball eos" sometimes failed to capture specific details, such as the number of children or the exact nature of the activity.

The **LSTM** model, without attention, was less accurate, often generating captions that were too generic or lacking specific details. For example, in images with multiple objects or people, the LSTM model would frequently produce vague captions such as "sos group of people are playing with ball eos," which did not fully describe the action or the interaction between subjects.

The **ViT-GPT2** model demonstrated some of the highest accuracy in terms of loss reduction, though the generated captions sometimes suffered from fluency issues, such as awkward sentence structures or underrepresentation of specific scene elements. Despite achieving a lower loss, the captions generated were occasionally incomplete or imprecise, reflecting challenges in capturing complex visual features.

Regarding Attention:

- The **Attention layers** in the LSTM + Attention model significantly improved caption generation by enabling the model to focus on specific regions of the image. This helped produce more contextually accurate descriptions, particularly in images with complex or crowded scenes.
- The absence of attention in the standard LSTM model resulted in more generalized and occasionally inaccurate captions, as the model struggled to maintain focus on critical areas of the image. It was more prone to generating vague or less descriptive outputs.
- The **ViT-GPT2** model benefited from its ability to process images in a more detailed manner, but attention-based mechanisms might further improve its captioning capabilities, particularly in terms of better localizing image features and actions.

In summary, Attention mechanisms significantly enhanced the model's ability to generate detailed and contextually relevant captions, with the **LSTM + Attention** model outperforming non-attention-based models in terms of accuracy and fluency. However, some errors in object recognition and sentence structure remained, particularly in more complex scenes. The **ViT-GPT2** model showed promise but could benefit from further improvements in attention integration to refine its captioning performance.