CoBICap: Context-Boosted Image Captioning with LSTMs and GRUs

Aadit Trivedi Avyesh Kapadia Atishay Jain Varun Patel
Georgia Institute of Technology
{atrivedi, akapadia31, atishay.jain, vpatel394}@gatech.edu

Abstract

This project focuses on creating an image captioning model based on the popular Flickr8k dataset. In the project, we compare the effectiveness of several model types, including Long Short-Term Memory networks (LSTMs), Bi-direction LSTMs, Gated Recurrent Units (GRUs) with attention, and Convolutional Neural Networks (CNNs) with Transformers. By training all models on Flickr8k and using Bilingual Evaluation Understudy (BLEU) scores to evaluate model accuracy, we find that the GRU model with attention mechanism outperforms others by producing more accurate and contextually relevant captions, thus demonstrating its superiority in handling the complexities of image captioning tasks. The attention-equipped GRU aligns dynamically with different image features, enhancing caption quality by focusing on relevant details during the generation process. The CNN with Transformers, despite their potential, performed moderately well, likely due to the limited size of the training dataset and inherent complexities of the model which may lead to overfitting. Furthermore, the more complex Bi-direction LSTMs were unexpectedly outperformed by the Vanilla LSTMs. This paper discusses the technical implementations, challenges, and the comparative performance of each model, providing insights into the future directions of employing advanced neural network architectures for real-world applications in accessibility technologies.

1. Introduction

The objective of our project was to create an accurate and reliable image captioning system, primarily motivated by our desire to improve accessibility and empower the visually impaired in their daily lifestyle. The most common and current image captioning practice involves extracting features using a convolutional neural network (CNN) and using a language model such as a recurrent neural network (RNN) or transformers [21] to selectively generate a caption. More recent research focuses on improving the quality of captions by employing an attention mechanism [7]. Some of the limits of these current practices involve poor context understanding and the generalization of real-world test data. Context is critical to an image-captioning task as it allows the model to better understand visual and textual relationships, improving the accuracy of the outputs. Additionally, generalization is a common pitfall of models in which models fail to produce quality captions on new and unique images which undermines its ability to be used reliably in the real-world. For our project, we used the Kaggle Flickr8k dataset [1], an industry standard dataset for image captioning. To better understand the image captioning task and attempt to incorporate context while avoiding generalization, we will employ and experiment with long short-term memory networks as part of our multi-step deep learning model that will generate a sequential caption based on image input.

2. Approach

We seek to conduct a comparative analysis of our various custom-built models on our image captioning task using existing and proven building blocks such as LSTMs and GRUs.

To extract the features from our image dataset, we used a convolutional neural network. Image feature extraction is done with a pretrained VGG16 model which is a 16-layer CNN. [18] We decided to develop our deep learning model that we train using long short-term memory (LSTM) networks due to their known forte of handling long sequential data, captions in our case, and dealing with sequential context. Our main variations of LSTM models include a vanilla LSTM and a bidirectional-based LSTM. The vanilla LSTM provides a strong baseline whereas the bidirectional-based LSTM offers stronger contextual capabilities due to its ability to process a sequence in both directions. We analyze the quality of our image captions as we progressively strengthen our LSTM, testing on standard benchmark metrics such as BLEU. [24]

One of our approaches utilized gated recurrent networks (GRU), which are similar to LSTMs in function but slightly different architecturally, featuring only update/reset gates instead of the typical 3 LSTM gates. Because GRU features

fewer parameters than LSTMs, they are simpler and ultimately train quicker. GRUs have shown to be more flexible regarding the dependencies of time scales, and have shown to converge faster in training because of the simpler architecture while still providing robust results. Our GRU implementation also featured an attention mechanism in the decoder, which focuses on certain parts of the image during caption generation. We discuss more in the third model section [3].

Our final approach was to use a CNN with a Transformer. A Transformer refers to deep learning model architecture that is specifically designed to process sequential data. This makes it great for natural language processing because language is inherently sequential—every word relies on the ones that came before it for meaning and the relative positions of words are extremely important even when they are extremely far apart in sequences. Thus, Transformers have become the basis for tasks such as machine translation and text summarization and what has led to generic use cases such as LLMs. Transformers, like the GRU approach, work by specifically relying on attention as the mechanism that captures dependencies between input and output sequences [8].

On a separate note, it is important to realize that in the dataset for image captioning that there can be inherent bias in the captions chosen for certain images, which can differ greatly across different image captioning datasets. Thus, we elected to use Flickr8k for each model that we created.

One predicted issue was the difficulty in the data transformation pipeline, starting from not one but two sources of input–the raw image and the caption–and passing them both through the model to arrive at a single output of one caption. This was a tricky thing to implement since the model must handle and synchronize two fundamentally different types of data. For example, each of these data types requires distinct preprocessing steps, and then they must be effectively integrated within the model's architecture. The image data, initially in the form of raw pixels, needs to be transformed through feature extraction processes to become meaningful feature vectors that represent the visual content, while the captions must be tokenized and converted into sequences of integers that are suitable for processing by neural network models.

There were a few unexpected issues with GPU training from an architectural standpoint. When training the GRU with attention model, we initially attempted to use Metal Performance Shaders (MPS), a GPU available on M1 Mac machines. However, after running into various unexpected problems, including tensors of incorrect dimension and unexpected functionalities during training, through some quick research we learned that MPS and GRU have several ongoing issues regarding using MPS to train GRU models. We elected to switch to CUDA using Google Colab to solve this issue.

Naturally, these intense and complex models require large amounts of time and vast resources to train, even through the use of GPU. The use of GRU was influenced by its relative simplicity compared to LSTM, which allowed it to train quicker.

3. Experiments and Model Architecture

3.1. LSTM

Our Vanilla-LSTM model has a dual-input encoder-decoder architecture. There are two distinct input pathways, one for image features and one for the textual data. Image input is vectorized in a 4096x4096 format. This data was regularized with dropout=0.4 to avoid the risk of overfitting. The dense layer and the ReLU activation are critical in detecting non-linear patterns in the images. The caption text is processed by an embedding layer prior to getting mapped to 256-dimension. As part of the caption pre-processing, we cleaned the caption data by removing noise and adding start and end sequence tokens. A LSTM layer then sequences through embedded words. The decoder brings these two pathways together, using a dense layer and ReLU activation which creates a probabilistic distribution for the next word in the generated caption.

3.2. Bi-directional LSTM

Bi-LSTM builds off Vanilla-LSTM by adding Bidirectional LSTMs. Bidirectional LSTMs can process textual data both in the forward and backward direction, providing more context and allowing for more complex text understanding and generation.

Let us take a look at the following equation which shows how output is calculated from a Bidirectional LSTM layer. y_t is the output sequence of the layer, $\overrightarrow{h}$ is the forward hidden sequence, and $\overleftarrow{h}$ is the backward hidden sequence. The forward and backward sequences are iterated from $t \in [1, N]$ where N is the size of the input sequence.

$$y_t = W_{\overrightarrow{h}y}\overrightarrow{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y$$

[22]

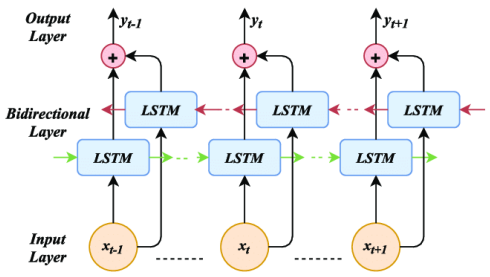


Figure 1: High-level architecture of Bidirectional LSTM layers [11]

The LSTM layer from Vanilla-LSTM has been replaced with a series of three Bidirectional LSTM layers, each having output dimension 512 due to the two 256-dimensional compositional LSTMs that comprise the Bidirectional LSTM.

For our LSTM models, our training sessions were 10 epochs with batch size 32. Increasing the epoch count did not significantly improve the accuracy of the model likely due to the limited size of dataset. We noticed overfitting even when we decreased the number of epochs, so we settled on 10 epochs as the ideal hyperparameter, and batch size 32 worked best for training efficiency given our dataset size.

Our LSTM models employ categorical cross-entropy loss function, which we thought to be effective since it would quantify discrepancies between probabilities of token classes function, and the Adam optimizer. Categorical cross-entropy provided us with an optimal balance of training efficiency and accuracy.

3.3. Attention-GRU

GRU (gated recurrent unit) is another decoding architecture similar to LSTM but different in a few aspects, such as the presence of 2 gates (update/reset) instead of 3 (input/output/forget). The update gate determines how much of the past information (from previous time steps) needs to be passed along to the future–ultimately how much the old state contributes to the new state–and the reset gate decides how much of the past information to forget. This model also makes use of the attention mechanism, which helps the model focus on specific parts of the image features while generating each word in the caption.

Firstly, images, which are of various sizes, are decoded and standardized to 255 x 255 pixels. Features are then extracted from the model using InceptionV3 (a common pre-trained CNN for feature extraction) configured to output tensors of features as opposed to class predictions outright. Then, the captions are tokenized (assigned to a unique integer) and padded to facilitate batch processing. The extracted features are then processed through an encoder, which has a dense layer and a ReLU layer. The decoder makes use of an attention model, which features two dense layers (one for image features and one for a hidden state), a tanh activation, and then a final dense layer followed by softmax. For each word being predicted in the caption, the decoder uses an attention mechanism to focus on specific parts of the image features, based on the current context and the previous hidden state of the Decoder. The decoder combines the context vector from the attention mechanism with its current state and the previous word's embedding to predict the next word in the sequence. This process uses a GRU layer followed by dense layers to generate predictions for the next word. The decoder will generate one word at a time starting with the special start token and going until the special end token.

The attention model/mechanism is essential for the decoder, which ultimately produces a caption, to select and focus on different parts of the image at different stages of the caption generation. This ability is important for generating contextually relevant and coherent captions that accurately reflect the actual content of the inputted image. At each decoding step, the attention mechanism takes the current hidden state of the decoder and all the image features to compute attention scores. These scores determine how much each part of the image features will contribute to the context vector for that iteration. Ultimately, by focusing on specific elements within an image when generating each word, the model is able to produce more accurate and coherent captions.

To analyze model performance, we calculated loss with Sparse Categorical Cross Entropy (SCE) loss, which is efficient in the realm of multi-class classification; we can consider the vocabulary to be assigned unique integers based on our tokenization, and SCE is efficient because it expects integers as model outputs.

3.4. CNN + Transformer

Even though LSTMs and RNNs also attempt to capture sequential relationships, transformers have the ability to be trained in parallel, which allows much greater efficiency for both training and inference. This is what allows Transformer based models to reach the complexities and depth of something like BERT or GPT. Furthermore, the transformer's use of attention leads to the potential to greatly increase model performance. The inclusion of this attention-based system seems to suggest that a Transformer would be excellent for such a task as it would lead to the best understanding of the visual and textual dependencies needed on each other.

Just like with the other models, the first step was to take the Flickr8k dataset and preprocess it with steps such as caption normalization and length filtering. The images also had to be preprocessed with steps such as resizing.

The next step was to choose a CNN to use for the image feature extraction. Although a Transformer is excellent at sequential dependencies, it is not great at spatial or positional dependencies and in order to capture that a CNN makes the most sense. We decided that it makes the most sense to start with a pretrained CNN, debating between using InceptionV3 and EfficientNet. Although InceptionV3 seems to perform better at a very large scale, EfficientNet, as the name suggest, seems to outperform other CNNs at its complexity level, indicating that it is more resource-

efficient for the image captioning task.

After the image features were extracted, we processed the output with a Transformer encoder. This encoder layer is where multiple layers of multi-head self-attention, layer normalization, and feed-forward neural networks were used to capture dependencies between different parts of the image using its features. We then utilized a Transformer decoder, once again with multiple layers of masked multi-head self-attention, layer normalization and feed-forward neural networks. Cross-attention was also used to attend to the encoder outputs, which allowed the decoder to focus on the relevant image features while generating each word in the caption. Finally, a dense layer with softmax formed as the last layer.

We also applied positional encoding to the input of both the encoder and the decoder to ensure that we retained positional data about the image features and the words in the caption. Lastly, we utilized teacher forcing during training to improve convergence. Teacher forcing was done by providing the ground truth caption as input to the decoder at each time step, which helped the model to learn how to generate accurate captions.

Note: All of our code detailed in this section is hosted at this GitHub repository

4. Results

4.1. BLEU score

The BLEU (Bilingual Evaluation Understudy) score is a widely adopted metric for assessing the quality of machine-generated text in natural language processing (NLP) tasks like machine translation and image captioning. Its core principle is to measure the similarity between the machine output and human-written reference texts by quantifying the overlap of words and their order. The higher the BLEU score, the closer the machine translation matches the human references, indicating better quality. Typically, we use BLEU-1 scores and BLEU-2 scores [10].

BLEU-1 refers to the "unigram" precision score. It measures the overlap of single words (unigrams) between the generated text and the reference text. It essentially counts the number of individual words from the generated caption that appear in the reference caption, then divides this number by the total number of words in the generated caption. BLEU-1 can provide insight on the accuracy of individual words in the translation or caption but doesn't consider word order or the correctness of phrases. Similary, BLEU-2 is calculated by counting the matches of bigrams (2-word phrases) in the generated text that appear in the reference, then dividing by the total number of bigrams in the generated text.

There are a few limitations of BLEU–for instance, it focuses more on the accuracy of generated captions–as opposed to capturing the semantic accuracy/fluency of the caption. In the context of our models, it is a strong choice to evaluate the quality of our models because it provides a straightforward, quantitative measure to evaluate the performance of image captioning models, allowing for objective comparisons.

4.2. Model BLEU Score Results

Model	BLEU-1	BLEU-2
Vanilla-LSTM	0.559054	0.332799
Bi-LSTM	0.543261	0.315804
Attention-GRU	0.616079	0.516973
CNN + Transformer	0.575922	0.341121

Table 1. BLEU Scores for Different Models

As shown above in Table 1, Based on our different model architectures, it seems that GRU with attention significantly outperforms the other models in terms of both BLEU-1 and BLEU-2 scores. The Attention-GRU model achieves a BLEU-1 score of 0.616079 and a BLEU-2 score of 0.516973. The CNN + Transformer model is just in suit, with a BLEU-1 score of .575922 and a BLEU-2 score of .341121. The Vanilla-LSTM model performs moderately well with a BLEU-1 score of 0.559054 and a BLEU-2 score of 0.332799. The Bi-LSTM model scores slightly lower than the Vanilla-LSTM on both metrics, with a BLEU-1 score of 0.543261 and a BLEU-2 score of 0.315804.

Image 1: two dogs play with each other on the sidewalk
Image 2: man standing on rocky peak
Image 3: two girls playing on lawn

Three example outputs using the Vanilla-LSTM model.

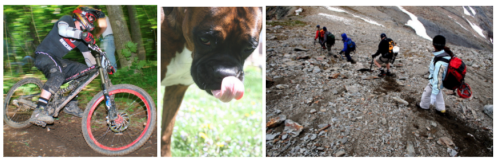
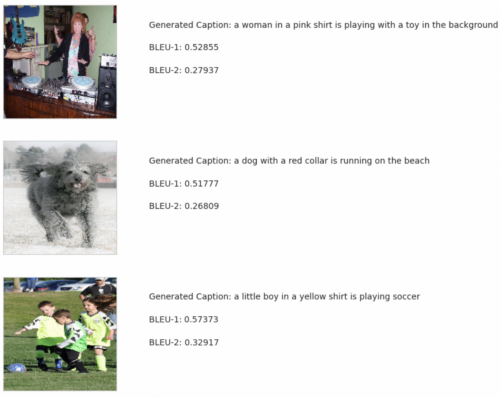


Image 1: man rides bike on dirt road
Image 2: closeup of brown dog licking its nose
Image 3: two hikers descend on rocky ground

Three example outputs using the Bi-LSTM model.

Actual Caption: girl in pink coat plays in the leaves
Predicted Caption: little girl in pink coat throwing leaves

An example output of the GRU with attention model, followed by the attention model highlights during the iterative parts of the captioning process.



Three example outputs of the CNN with transformer model, with their respective BLEU-1 and BLEU-2 scores.

5. Discussion

5.1. Dominance of GRU

GRU with attention, despite being a simpler model inherently and taking less time to train, shows much more accurate results and outperforms the other two LSTM models. This is likely due to the use of the attention mechanism, which enhances the GRU's capability by allowing it to focus on specific parts of the input sequence or, in the case of image captioning, specific regions of the image.

The attention mechanism enables the model to weigh different parts of the input differently, focusing on those aspects that are most relevant to generating the next word in the caption. This is particularly useful in image captioning, where different objects or areas of the image may be more relevant at different points in the caption. Specifically, if captions are generated based these particular points, a caption-generating model will thus benefit greatly from an attention feature in the decoder. This attention-equipped GRU can produce more accurate and contextually appropriate captions. This dynamic focusing aligns the model's "view" with that of a human describing the image, leading to higher quality outputs.

Furthermore, in long sequences, attention helps manage the loss of context that might occur with basic RNNs or even LSTMs and GRUs alone. It effectively brings earlier parts of the input back into consideration, which might be critical for understanding or describing later parts. The success of the attention mechanism can also be seen in the CNN with Transformer model, which was the second-most successful model and the only other one to use attention, as discussed in section 5.2.

Additionally, GRU being an inherently simpler model, for instance with only two layers as opposed to the three of LSTMS, allowed for increased data efficiency during learning and training even with the limited size of the Flickr8k dataset, the size of which was a limiting factor in the more robust models, as we discuss in the following sections. The simpler nature of the GRU archiecture also led us to decreasing the learning rate; since it is already simpler and can converge faster during training, we lowered the learning rate from .0005 to .0001 to show an increase in model success and BLEU scores. Decreasing the learning rate more to .00001 did not yield successful results, likely once more due to the limited size of the dataset.

5.2. CNN with Transformer Performance

The results of the CNN with Transformer are not outstanding, and it does not seem to outperform the GRU with the attention model as we had hypothesized. There can be a variety of reasons that lead to this. For example, the complexity of Transformers usually makes them prone to overfitting, which could have happened here as the training losses seem low. However, it is more likely that the smaller size of the dataset did not allow us to fully unleash capabilities of a Transformer, which performs more exceedingly as it scales. While the CNN with Transformer model did surpass the performance of the LSTM models, it fell short of achieving the effectiveness of the GRU with attention, which remains the most suitable for this application given the current dataset constraints and the model's capacity to manage sequential dependencies more efficiently.

As mentioned previously, Transformers truly excel in environments where vast amounts of data are available, as their large number of parameters are aptly suited to captur-

ing intricate patterns in large-scale data. The limited dataset size of Flickr8k, which was 8000 images, despite the diverse data (such as 5 human-written captions per image) may have restricted our ability to fully harness and demonstrate the capabilities of the CNN with Transformer architecture, potentially leading to its underperformance compared to the simpler, more data-efficient models like the GRU with attention.

Given the complexity and propensity for overfitting for the Transformer, we focused on fine-tuning its attention mechanism and experimenting with different numbers of attention heads. Initial configurations used standard settings, but adjustments were made to increase the number of attention heads from 8 to 16, allowing the model to attend to more aspects of the input data simultaneously. While this did slightly improve BLEU scores, as aforementioned increasing the complexity of the model with limited dataset size had accordingly limited effects.

5.3. Underperformance of Bi-Directional LSTM

One surprising result was the slight underperformance of the bi-direction LSTM in comparison to the vanilla LSTM. This was unexpected because for many natural language processing tasks, understanding context from both directions can provide significant benefits. However, a potential reason for the underperformance could be the overhead caused by the increase number of parameters. Because Bi-direction LSTMs consider LSTMs from both directions, they effectively double the number of parameters required. This increase in model complexity, especially with limited data or limited computational resources. Furthermore, with a more complex model, bi-directional LSTMs require more data to generalize effectively without overfitting. The extra parameters might lead to a weaker performance compared to a simpler model that matches the data constraints better, which could have happened in the vanilla LSTM vs bi-directional LSTM case.

6. Conclusion

Our research was aimed at developing a high-accuracy image captioning system using LSTMs and GRUs. Although we expected Bi-LSTM to be the most successful due to the bidirectional context processing, we found that using a simpler model architecture with attention proved to be the most optimal within the scope of our study, as shown by the success of GRU with attention. The simpler, two-layer architecture allowed for the high data efficiency given the limited dataset, and there was the added precision of the attention mechanism in focusing on salient features of the image relevant to the generated captions.

However, it is important to recognize some of the limitations of our work. First and foremost, we were working on a Google Colab environment with free GPU access which obviously serves as a limitation on our computational flexibility and performance. Additionally, our model's performance is bounded by the Flickr8k dataset–specifically the size. The potential of our complex models, such as the bi-directional LSTM and CNN with Transformer were bounded by the training size.

Our future research should focus on expanding dataset sizes, exploring hybrid models that can leverage both the simplicity of GRUs and the comprehensive attention mechanisms of Transformers, and implementing these models in real-world devices to truly transform how visually impaired individuals interact with their environment.

7. Further Research

There are a number of various directions to take this research going forwards. First and foremost, there is potential to explore new image feature extraction techniques. The potential of Vision Transformers (ViTs) should be experimented with for feature extraction. ViTs have proven success by segmenting an image into pieces and applying attention to these pieces which improves the model's ability to capture universal attention [6].

We could also integrate some sort of knowledge graph or informational database to our captioning sequence model to generate accurate information. This would help the captions be more factually accurate. For instance an incorrectly generated caption saying "rain is melting into thin snow" would be corrected to "rain is freezing into thin snow." Using a knowledge graph like ConceptNet [19] would help our model better understand factual relations between items and correct incorrectly generated captions like shown before. Along similar lines, simply using a larger dataset (larger than the 8000 of Flickr8k) would likely help our specific models generalize and learn better during the training of our data.

The next step to better address the needs of the visually impaired would be continuing to refine our model to reach higher accuracy scores and mounting our model on an end-to-end usable system that serves of value to the visually impaired. This could take the form of smart glasses or a smart watch that actively monitors surroundings and has some sort of audio cue to relay, or explain, the surroundings to its user. We could then take advantage of these devices to capture data for further training purposes post-annotation as the expansion of our dataset will only improve our model's functionality. Work still needs to be done to solve this impending accessibility issue, but our work in this paper marks an important starting step.

References

[1] adityajn105. Flickr 8k dataset. https://www.kaggle.com/datasets/adityajn105/flickr8k/data, June 2020. Retrieved April 24, 2020. 1

Student Name	Contributed Aspects	Details
Aadit Trivedi	Data Processing / Implementation / Analysis	Co-worked on data processing for image features and caption sequences. Co-implemented Vanilla-LSTM, ran experiments, and analyzed results. Implemented Bi-LSTM, ran experiments, and analyzed results.
Avyesh Kapadia	Data Processing / Implementation / Analysis	Co-worked on data processing for image features and caption sequences. Implemented GRU with attention, ran experiments and conducted analysis. Co-performed analysis and comparison among all models.
Atishay Jain	Implementation / Analysis	Co-performed literary survey of research space and evaluated usability of various datasets. Co-implemented Vanilla-LSTM, ran experiments, and analyzed results. Implemented CNN + Transformer, ran experiments, and analyzed results.
Varun Patel	Data Processing	Co-performed literary survey of research space and evaluated usability of various datasets. Found dataset and wrote a script to scrape data from the repository.

Table 2. Contributions of team members.

[2] Rana Adnan Ahmad, Muhammad Azhar, and Hina Sattar. An image captioning algorithm based on the hybrid deep learning technique (cnn+gru), 2023.

[3] archis2004. Image to caption using attention and gru with gtts. https://www.kaggle.com/code/archis2004/image-to-caption-using-attention-and-gru-with-gtts?scriptVersionId=173022896, 2024. 2

[4] Baeldung. Bidirectional vs. unidirectional lstm. https://www.baeldung.com/cs/bidirectional-vs-unidirectional-lstm, 2024. Accessed: 2024-04-30.

[5] Roberto Cahuantzi, Xinye Chen, and Stefan Güttel. *A Comparison of LSTM and GRU Networks for Learning Symbolic Sequences*, page 771–785. Springer Nature Switzerland, 2023.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 6

[7] Abdul Daffa Fauzulhaq, Wayan Parwitayasa, Joni Agung Sugihdharma, Muhammad Fajar Ridhani, and Nadi Yudistira. Mami: Multi-attentional mutual-information for long sequence neuron captioning. *arXiv*, 2023. 1

[8] Saeed Ghamshadzai. Image captioning with transformers on flickr8k. https://www.kaggle.com/code/saeedghamshadzai/image-captioning-transformers-flickr8k/notebook, 2023. 2

[9] Taisiya Glushkova, Chrysoula Zerva, and André F. T. Martins. Bleu meets comet: Combining lexical and neural metrics towards robust machine translation evaluation, 2023.

[10] Google. Evaluate your models with bleu score - cloud automl translation. https://cloud.google.com/translate/automl/docs/evaluate#bleu, 2024. 4

[11] Isibor Ihianle, Augustine Nwajana, Solomon Ebenuwa, Richard Otuka, Kayode Owa, and Mobolaji Orisatoki. A deep learning approach for human activities recognition from multimodal sensing devices. *IEEE Access*, 8:179028–179038, 2020. 3

[12] Rashid Khan, Bingding Huang, Haseeb Hassan, Asim Zaman, and Zhongfu Ye. A Comparative Study of Pre-trained CNNs and GRU-Based Attention for Image Caption Generation. *arXiv e-prints*, page arXiv:2310.07252, October 2023.

[13] Rashid Khan, M Shujah Islam, Khadija Kanwal, Mansoor Iqbal, Md. Imran Hossain, and Zhongfu Ye. A deep neural framework for image caption generation using gru-based attention mechanism, 2022.

[14] Gordon Euhyun Moon and Eric C. Cyr. Parallel training of gru networks with a multi-grid solver for long sequences, 2022.

[15] Matt Post. A call for clarity in reporting bleu scores, 2018.

[16] Harshit Rampal and Aman Mohanty. Efficient cnn-lstm based image captioning using neural network compression, 2020.

[17] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, March 2020.

[18] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 1

[19] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge, 2018. 6

[20] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. From show to tell: A survey on deep learning-based image captioning, 2021.

[21] Yunhao Xu, Li Li, Hongliang Xu, Siwei Huang, Fei Huang, and Jianwei Cai. Image captioning in the transformer age. *arXiv*, 2204.07374, April 2022. 1

[22] Zhiwei Yu, Vikram Ramanarayanan, David Suendermann-Oeft, Xinhao Wang, Klaus Zechner, Lei Chen, Jian Tao, Andrei Ivanou, and Yao Qian. Using bidirectional lstm recurrent neural networks to learn high-level abstractions of sequential features for automated scoring of non-native spontaneous speech. In *Proceedings of the ASRU*. Carnegie Mellon University, 2015. 2

[23] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Gated recurrent units (grus). https://d2l.ai/chapter_recurrent-modern/gru.html, 2024.

[24] zohaib123. Image caption generator using cnn and lstm. https://www.kaggle.com/code/zohaib123/image-caption-generator-using-cnn-and-lstm, April 2023. Retrieved April 24, 2023. 1