INFO 202: Final Project - Minkush Jain

Project Title: Automatic Image Captioning Using NLP and Deep Learning

Introduction

Automatic image captioning is an interdisciplinary problem that combines computer vision and natural language processing (NLP) to generate descriptive captions for images. This project aims to develop a deep learning-based system that generates coherent and contextually relevant captions for images, leveraging concepts from the class such as word embeddings, tokenization, vector representations, and data preprocessing. The Flickr 8k dataset, which consists of 8,000 images paired with descriptive captions, serves as the foundation for training and evaluating the model. TensorFlow and Keras were utilized for building and training the model. This write-up describes the step-by-step process, technical specifications, experimentation, and alignment with class concepts.

Process and Technical Implementation

The implementation of this project involved several stages: data preprocessing, model architecture design, training, evaluation, and optimization.

1. Data Preprocessing

The Flickr 8k dataset contains both image files and text descriptions. Before feeding this data into the model, extensive preprocessing was performed on both modalities:

- **Image Preprocessing:** Images were resized to a uniform shape (299x299 pixels) and passed through a pre-trained Convolutional Neural Network (CNN), InceptionV3, to extract feature vectors from the final pooling layer. These feature vectors serve as a compact representation of each image.

- **Text Preprocessing:** The captions were tokenized, lowercased, and stripped of punctuation to standardize input. Special tokens such as <start> and <end> were added to indicate the beginning and end of a sentence. Word frequencies were analyzed to construct a vocabulary, limiting it to the top 5,000 most frequent words to manage complexity.
- **Word Embeddings:** Pre-trained GloVe embeddings were used to map each word in the captions to a 100-dimensional vector. This representation captures semantic relationships between words, such as synonyms and contextual similarities.

2. Model Architecture

The model architecture consisted of two main components:

- **Feature Extraction (CNN):** The pre-trained InceptionV3 model was used to extract high-level features from images. These features were then passed through a fully connected layer to reduce dimensionality, resulting in a fixed-size feature vector for each image.

- **Language Model (LSTM):** The language generation component was built using a Long Short-Term Memory (LSTM) network, which processed sequences of word embeddings. The LSTM was trained to predict the next word in a caption based on the current word and the image features.

The two components were combined by feeding the image features into the LSTM as an initial state, followed by sequentially feeding in the tokenized words of the caption. The output was a probability distribution over the vocabulary for each predicted word.

3. Training the Model

The model was trained using categorical cross-entropy loss, which evaluates how well the predicted word probabilities align with the actual captions. The Adam optimizer was used to minimize the loss, with an initial learning rate of 0.001. The training process was computationally intensive, requiring a batch size of 64 and running for 30 epochs.

To prevent overfitting, techniques such as dropout and early stopping were applied. Additionally, the dataset was split into training, validation, and test sets in a ratio of 80:10:10 to ensure robust evaluation.

4. Experimentation

Several experimentation techniques were applied to improve caption generation:

1. **Greedy Search:** The simplest decoding method, where the model selects the word with the highest probability at each step. While computationally efficient, this approach often produces suboptimal captions due to the lack of consideration for long-term context.

2. **Beam Search:** This method keeps track of multiple possible sequences at each step, allowing the model to explore a broader search space. A beam width of 3 was used, which improved the quality of generated captions by considering multiple hypotheses.

3. **BLEU Score Evaluation:** To evaluate the quality of generated captions, the BLEU (Bilingual Evaluation Understudy) score was calculated. BLEU compares the n-grams of

the generated captions with the reference captions, providing a measure of how similar they are. BLEU-1 (unigrams), BLEU-2 (bigrams), and BLEU-4 (four-grams) were computed to assess different aspects of fluency and coherence.

Incorporation of Class Concepts

This project incorporates several key concepts covered in class:

1. **Word Embeddings:** Pre-trained GloVe embeddings were used to represent words as dense vectors, enabling the model to understand semantic relationships. This relates directly to the class discussions on lexical relations and semantic similarity.

2. **Tokenization and Sequence Processing:** The text preprocessing pipeline included tokenization, padding, and sequence truncation, ensuring that input captions were consistent in length. These preprocessing steps were emphasized in the class topics on structured data processing.

3. **Feature Vectors and Vectors in Space:** The use of CNNs to extract feature vectors from images aligns with the class concept of representing information in vector space. These vectors serve as a bridge between visual and textual data.

4. **Evaluation Metrics:** The use of BLEU score to evaluate the model's performance is consistent with class discussions on how to measure the effectiveness of language models and retrieval systems.

5. **Handling Ambiguity:** The class discussion on the vocabulary problem and ambiguity in communication was relevant here, as the model often struggled with selecting the most appropriate word. Techniques like beam search helped address this challenge.

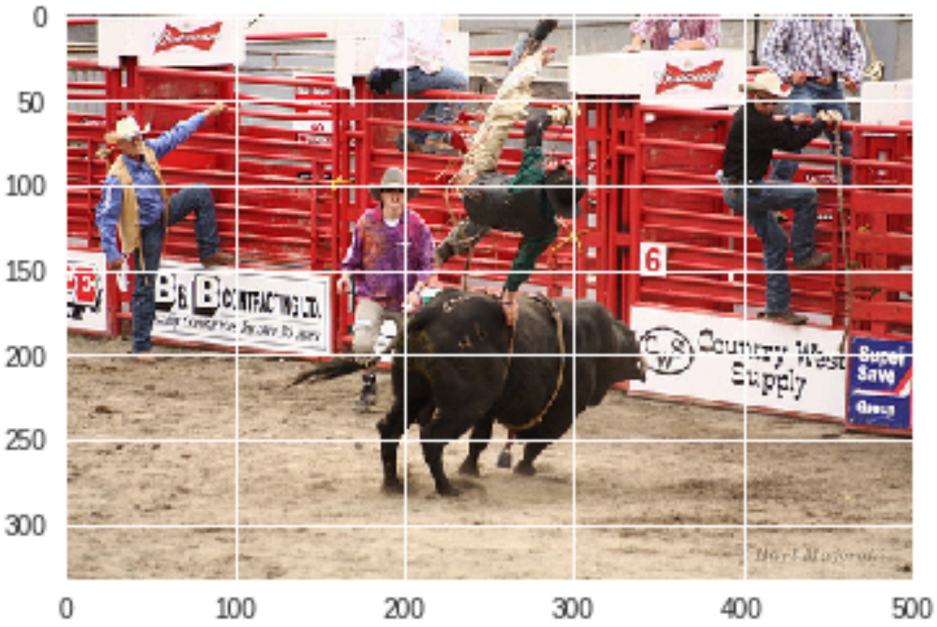
Results

- **Greedy Search Performance:** Generated captions were coherent but often lacked variety and nuance. The BLEU-4 score for greedy search was approximately 0.32.

- **Beam Search Performance:** Captions generated using beam search were more diverse and contextually accurate, achieving a higher BLEU-4 score of 0.42. This highlights the advantage of considering multiple hypotheses during decoding.

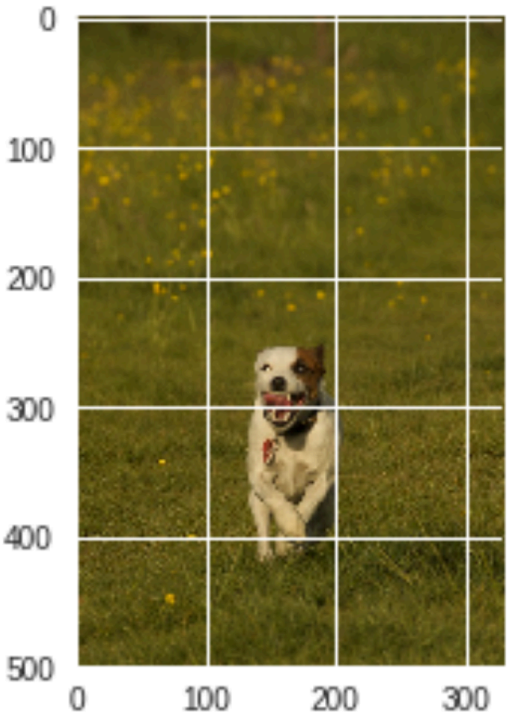
- **Qualitative Results:** The model performed well on simple, clear images (e.g., "a dog playing with a ball") but struggled with complex scenes or abstract concepts.

Output Examples:



man in black shirt is riding bull

two children are playing with plastic toy



white dog is running through the grass

Challenges and Future Improvements

1. **Dataset Limitations:** The Flickr 8k dataset is relatively small for a deep learning project, limiting the model's ability to generalize to unseen images. Future work could use larger datasets such as MS COCO for improved performance.

2. **Handling Ambiguity:** While beam search improved results, the model still occasionally produced grammatically correct but semantically irrelevant captions. Exploring advanced techniques like transformer-based models or incorporating attention mechanisms could address this.

3. **Bias in Captions:** The model occasionally reflected biases present in the dataset, such as stereotypical descriptions of people or activities. Addressing this would require curating a more diverse and balanced dataset.

Timeline Reflection

The project followed a structured timeline:

- **November 12–20:** Data preprocessing and initial exploration of the dataset.

- **November 21–30:** Model architecture design, training, and implementation of greedy search decoding.
- **December 1–7:** Experimentation with beam search and BLEU score evaluation.
- **December 8–10:** Final testing, result analysis, and preparation of the write-up.