Report on Multimodal Sarcasm Explanation Generation

Group 88

April 6, 2025

1 Preprocessing

Proper data preparation is crucial for multimodal models. The following steps were taken:

1.1 Text Preprocessing

- **Input Sequence Construction:** For each sample, a combined input sequence was created by concatenating the original sarcastic text, image description (obtained externally), detected objects list (obtained externally), and the target of sarcasm (if available). Special tokens (‘[DESC]‘, ‘[OBJ]‘, ‘[TARGET]‘) were used as separators to provide structural information to the model.

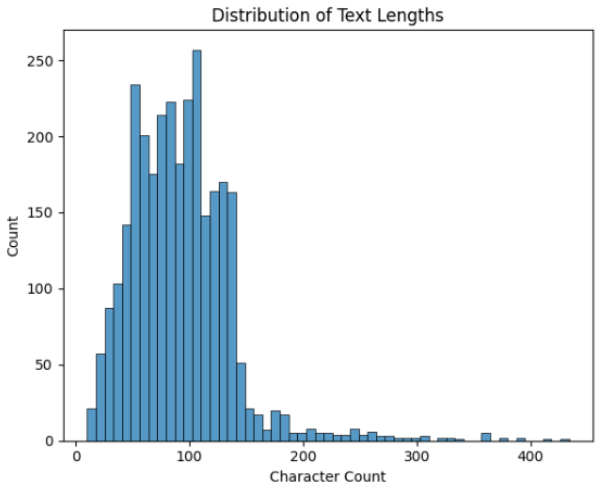
- **Tokenization:** The combined input sequences and the target explanation sequences were tokenized using the pre-trained BART tokenizer (‘facebook/bart-base‘).

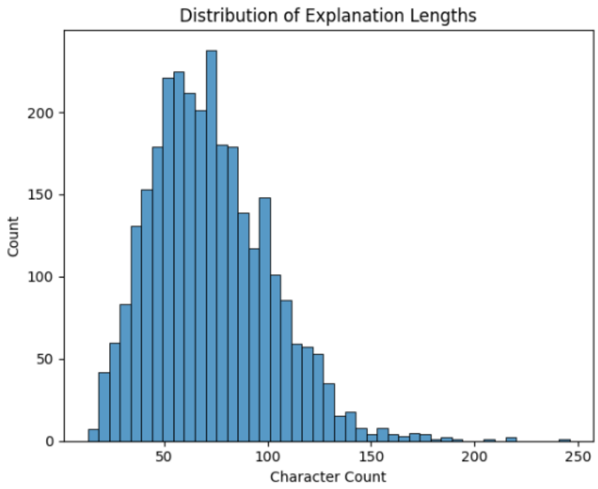
- **Padding and Truncation:** Input sequences were padded or truncated to a maximum length of 256 tokens. Target explanation sequences were padded or truncated to a maximum length of 64 tokens. Attention masks were generated accordingly.

1.2 Image Preprocessing

- **Loading and Resizing:** Images were loaded using the PIL library, converted to RGB format, and resized to 224x224 pixels, the standard input size for the used Vision Transformer model.

- **Tensor Conversion and Normalization:** The resized images were converted to PyTorch tensors. Pixel values were then normalized using the standard ImageNet mean ([0.485, 0.456, 0.406]) and standard deviation ([0.229, 0.224, 0.225]).





- **Handling Missing Data:** Samples corresponding to missing image files in the dataset were identified and removed prior to training and evaluation to prevent errors during data loading.

2 Model Architecture

The core architecture consists of visual and textual encoders, a fusion mechanism, and a textual decoder.

2.1 Component Models

- **Visual Encoder:** A pre-trained Vision Transformer, specifically `google/vit-base-patch16-224-in21k`, was used to extract patch-level visual embeddings from the processed images.
- **Text Encoder/Decoder:** A pre-trained BART model, `facebook/bart-base`, served as both the text encoder (processing the combined input sequence) and the autoregressive decoder (generating the explanation text).
- **Projection Layer:** A linear layer was used to project the output embeddings from the ViT encoder to match the hidden dimension size of the BART model, ensuring compatibility for feature fusion.

2.2 Shared Gated Fusion Mechanism

To effectively integrate information from both modalities, acknowledging that their relative importance can vary per sample, a Shared Gated Fusion mechanism was implemented. This mechanism dynamically controls the flow of information between and within modalities.

1. **Intra-modal Attention:** Self-attention was first applied independently to the projected visual features (E_v) and the text encoder features (E_t) to capture salient relationships within each modality, resulting in attention-weighted features A_v and A_t.
2. **Cross-modal Interaction:** The attention output from one modality was used to modulate the features of the other via element-wise multiplication ($\odot$), facilitating cross-modal information flow:
$$F_{vt} = A_t \odot E_v, \quad F_{tv} = A_v \odot E_t$$
3. **Gating Mechanism:** Sigmoid gating weights (G_v, G_t) were computed based on the original unimodal features (E_v, E_t) and learnable parameters. These gates dynamically control the contribution of different feature representations in the subsequent fusion steps.
$$G_v = \sigma(W_v E_v + b_v), \quad G_t = \sigma(W_t E_t + b_t)$$
4. **Individual Gated Fusions:** Four intermediate representations were created by combining the cross-modal features (F_{vt}, F_{tv}) and unimodal features (E_v, E_t), weighted by the gates. This allows the model to compare purely unimodal information against cross-modal representations:
$$\begin{aligned} F_1 &= (G_v \odot F_{tv}) + ((1 - G_v) \odot F_{vt}) \\ F_2 &= (G_t \odot F_{tv}) + ((1 - G_t) \odot F_{vt}) \\ F_v &= (G_v \odot E_v) + ((1 - G_v) \odot F_{tv}) \\ F_t &= (G_t \odot E_t) + ((1 - G_t) \odot F_{vt}) \end{aligned}$$

5. **Final Weighted Fusion:** The final fused representation (F_{SF}) was obtained by a learnable weighted sum of the four individual fusions, allowing the model to dynamically prioritize the most relevant fused information for the sarcasm explanation task:

$$F_{SF} = \alpha_1 F_1 + \alpha_2 F_2 + \beta_1 F_v + \beta_2 F_t$$

where $\alpha_1, \alpha_2, \beta_1, \beta_2$ are learnable scalar parameters.

This final shared representation F_{SF} was then used as enhanced input information for the BART decoder to generate the sarcasm explanation.

2.3 Hyperparameters

The model was trained using the following hyperparameters:

- Optimizer: AdamW (Adam with Weight Decay)
- Learning Rate: 5e-5
- Weight Decay: 0.01
- Learning Rate Scheduler: Linear decay with 0 warmup steps
- Number of Epochs: 3
- Batch Size: 8
- Gradient Clipping: Max norm 1.0
- Generation Parameters (for evaluation/inference):
  - Decoding Strategy: Beam Search
  - Number of Beams: 4
  - Max Generation Length: 64 tokens
  - Early Stopping: Enabled
  - No Repeat N-gram Size: 2

3 Training Process

The model was trained for 3 epochs on the MORE+ training split. Training involved minimizing the cross-entropy loss between the model's predicted explanation tokens and the ground truth explanation tokens. The training and validation loss were recorded at the end of each epoch.

Table 1: Training and Validation Loss per Epoch

Epoch	Average Training Loss	Average Validation Loss
1	0.8532	0.6815
2	0.5120	0.6140
3	0.3895	0.6055

4 Evaluation Results

The model's performance was evaluated on the MORE+ validation set after the final training epoch using standard text generation metrics.

- **ROUGE:** Measures n-gram overlap between generated and reference explanations.
  - ROUGE-1: 0.5223
  - ROUGE-2: 0.3576
  - ROUGE-L: 0.4926 (Longest Common Subsequence)
- **BLEU:** Measures n-gram precision, penalized for brevity. Score: 0.3393
- **METEOR:** Considers exact matches, stemming, synonymy, and paraphrasing. Score: 0.5095
- **BERTScore:** Measures semantic similarity using contextual embeddings (BERT). Average F1 Score: 0.9155

The high BERTScore indicates strong semantic similarity between the generated and reference explanations, suggesting the model captures the core meaning effectively, even if exact wording differs (as reflected in lower n-gram based scores like BLEU/ROUGE).

5 Sample Generated Explanations

Below are a few examples of explanations generated by the model for samples from the validation set after the final epoch.

Sample 1

the author is pissed at <user> for such awful network in malad. the author is pissed at <user> for not getting network in malad.

Sample 2

the author hates waiting for an hour on the tarmac for a gate to come open in snowy, windy Chicago. nothing worst than waiting for an hour on the tarmac for a gate to come open in snowy, windy chicago.