Multimodal Sentiment Analysis with Vision Transformers and BERT

Brent Weiffenbach [1] **Mauricio Mergal** [1] **Jeffrey Li** [1] **Tanveer Kaur** [1]

Abstract

Image-Text Sentiment Analysis requires understanding emotional cues from multiple sources to determine an overall sentiment. In this project, we implement a multimodal sentiment classification model from scratch in PyTorch. The approach uses a Vision Transformer to encode vision features, and BERT to encode textual features. The encoded representations are positionally aligned and passed to a fusion module to extract a multimodal feature space. A fully connected layer with a softmax activation function acts as a prediction head to predict positive, negative, or neutral sentiment for the image-text pair.

1. Introduction

Multimodal Sentiment Analysis (MSA) aims to bridge the gap between the separate aspects that make up our everyday conversations like tone, imagery, and context by integrating this type of information to produce a more accurate understanding of sentiment. In this project, we will focus on building a model capable of jointly analyzing images and text to predict sentiment labels (positive, neutral, negative). Our approach leverages a transformer-based architecture to learn shared representations between visual and textual modalities, allowing the model to capture the subtle cross-modal relationships that single-modal models often overlook.

To train and evaluate our model, we will use the MVSA-Multiple dataset (Niu et al., 2016), which consists of approximately 19,600 annotated image–text pairs collected from Twitter, each labeled by three independent annotators.

Building on the insights from prior work such as CTMWA (Zhang et al., 2024), we aim to implement our own MSA model from scratch. Rather than relying on pre-trained encoders, we will design and train each component ourselves. This approach will allow us to study how multimodal representations can be effectively learned without external pretraining and to evaluate whether such a model can achieve performance comparable to the baseline set by CTMWA at $\sim$74% accuracy.

[1]Worcester Polytechnic Institute. Correspondence to: Brent Weiffenbach <rweiffenbach@wpi.edu>, Mauricio Mergal <mjmergal@wpi.edu>, Jeffrey Li <jcli@wpi.edu>, Tanveer Kaur <tfnu@wpi.edu>.

CS/DS 541 Deep Learning Final Project, Worcester Polytechnic Institute.

2. Related Work

Image-Text Sentiment Analysis (ITSA) has become increasingly important as social media platforms are flooded posts that combine images and text to express emotion. Early approaches often relied on breaking down the image into objects or scenes using computer vision tools, then matching those objects to keywords in the text. While this worked okay, it was prone to errors—if the object detector missed something, the whole analysis could go wrong.

More recently, the rise of Transformer models has changed the game. Instead of relying on separate, error-prone steps, these models can learn to understand both image and text simultaneously from raw pixels and words. One standout example is CLMLF, which uses contrastive learning to pull image and text features closer together if they express the same sentiment. These approaches are powerful because they capture the shared meaning between modalities. But here's the catch: they often miss the fact that sometimes, the real emotion comes from just one modality. A picture might be neutral, but the caption could be sarcastic. Or vice versa. The model needs to know which part of the puzzle matters most for each specific post.

This is where the Crossmodal Translation-Based Meta Weight Adaption (CTMWA) model, introduced by Zhang et al. (Zhang et al., 2024), really stands out. Their approach doesn't just look for shared meaning; it also tries to figure out which modality—image or text—is doing the heavy lifting for each individual post. They do this in two clever ways. First, they build a "translation" network that can turn an image into a text-like representation, and vice versa. This helps the model handle situations where one modality is missing (like a post with only a picture). Second, and most importantly, they use meta-learning to train a special module that learns to assign weights to the image and text predictions. Think of it as the model asking itself, "For this particular post, should I trust the image more, or the text?" By training on a small set of manually annotated

examples, this weight module learns to dynamically adjust its focus based on the situation.

Recent work by Ren (Ren, 2024) has also explored multimodal sentiment analysis using BERT and ResNet architectures, demonstrating the effectiveness of combining transformer-based text encoders with convolutional image encoders. However, our approach differs by using Vision Transformers for both modalities, enabling a more unified architecture.

In our project, we're starting by replicating the CTMWA model (Zhang et al., 2024) as a benchmark. Our goal isn't to reinvent the wheel right away, but to first understand exactly how this state-of-the-art method works—from data preparation to training. Only by building a solid, reproducible baseline can we confidently design and test our own fusion transformer later on.

3. Proposed Method

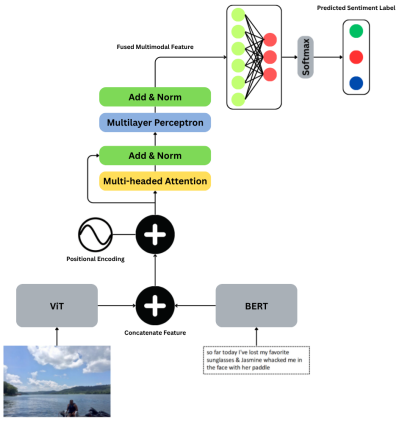

Figure 1. Overview of the proposed multimodal sentiment analysis architecture. The model encodes image and text inputs using Vision Transformer and BERT, respectively, then fuses the representations for sentiment classification.

3.1. Vision Encoder

The encoder is implemented using a Vision Transformer (ViT) architecture, which will process image patches as a sequence of tokens rather than using traditional convolutional operations. Each input image is first divided into fixed-sized patches, which are flattened and projected into the latent embeddings space using a learnable linear projection. Positional encodings are also added to retain spatial information, and the resulting sequence of embeddings is passed through a stack of transformer encoder layers. Each transformer layer consists of multi-head self-attention and feed-forward sublayers, both followed by residual connections and layer normalization. Through self-attention, the ViT model can create global dependencies between image regions, allowing it to capture context beyond the local texture patterns, which is particularly useful for our purposes where cues may arise from the contextual relationships found in the overall layout of the image rather than small-scale details. The output from the [CLS] token serves as the final visual embedding. Dropout and weight decay are applied for regularization. During training, gradients from the sentiment classification loss propagate back through both the fusion layers and the ViT encoder, enabling the model to learn patch representations.

3.2. Text Encoder

The text encoder is implemented using BERT architecture. BERT is a transformer-based language model that processes text bidirectionally, meaning it uses context from both the left and the right. It uses Masked Language Modeling (MLM) which works by hiding some of the words in a sentence and making BERT guess them, which helps the model learn how words relate to each other in context, and Next Sentence Prediction (NSP), which trains BERT to decide whether one sentence logically follows another, teaching it to understand the flow and structure of longer text, to learn robust semantic and syntactic representations that transfer effectively to downstream tasks such as sentiment analysis. We use the bert-base-uncased variant, which consists of 12 transformer encoder layers with a hidden size of 768. Input text is tokenized using BERT's WordPiece tokenizer with a maximum length of 128 tokens. The encoder outputs a contextual embedding sequence ([B, T, 768]) and a pooled [CLS] embedding representing the overall sentence meaning. To ensure compatibility with the 256-dimensional output of the Vision Transformer, we apply a linear projection layer that maps the 768-dimensional BERT sentence embedding into a 256-dimensional representation. A lightweight classification head then produces logits for the three sentiment categories (positive, neutral, negative). This setup enables seamless integration into our multimodal fusion model.

3.3. Fusion Transformer

The fusion transformer is responsible for modeling the relationship between the image and text. Often an image that could have a neutral tone may be heavily impacted by the image, or a sentence that could be perceived negatively might be sarcastic or a joke given an attached image. The implementation for our fusion transformer is inspired by an implementation from a 'Topic-oriented Model' on GitHub. The fusion module in this case mirrors a transformer encoder block. The intuition behind using a multi-headed attention layer should help the network learn how parts of one modality matter for understanding the other, with a residual layer and normalization to stabilize training. The MLP

adds some amount of non-linearity before going to the classification head. Finally, a fully connected layer takes the fused multimodal feature and passes it through to the final layer of size three, which represents the classification of positive, negative, or neutral.

4. Experiment

This section outlines the experimental design for our MVSA project, including the data preparation, model training strategy, and baseline comparisons. To reiterate, our primary objective is to assess whether a multimodal transformer-based model implemented entirely from scratch can achieve comparable performance to existing pre-trained architectures on the MVSA-Multiple dataset (Niu et al., 2016).

4.1. Dataset

We use the MVSA-Multiple dataset for our experiments. This dataset consists of approximately 19,600 image-text pairs collected from Twitter, each labeled with a sentiment category (positive, neutral, or negative) by three independent annotators. Started off with the raw MVSA-Multiple dataset, our first major task was to organize the data. We wrote a script to segregate the mixed .jpg and .txt files into dedicated `images/` and `texts/` folders. Then we used a pre-trained CLIP-ViT-B/32 model to extract 512-dimensional features for every valid image and text pair.

4.2. Training Configuration

Our model jointly processes text and image inputs through both text and visual encoders, with a fusion mechanism integrating the modalities. Training is conducted using PyTorch at an initial learning rate of 2×10^{-4}, batch size of 32, and cross-entropy loss. Regularization techniques include dropout and weight decay. Random seeds are fixed during this experiment to ensure reproducibility.

4.3. Baseline Comparison

We use CTMWA (Zhang et al., 2024) as a strong baseline for the model, as it is also designed to be used with this dataset and task in mind. To compare the results of both models we will use the same test split of MVSA-Multiple using accuracy and macro F1-score as the primary metrics. Additional analysis may include, if time allows, per-class precision, recall, and F1, as well as confusion matrices to help identify common misclassifications. This comparison allows us to quantify how close a from-scratch approach can come to a state-of-the-art pre-trained baseline and to understand the contribution of architectural design choices versus pretraining.

5. Results

Currently, we have the data loader working, which reads each sample's image, text caption, and sentiment annotations from the MVSA folder, and then used PyTorch's DataLoader to automatically batch, shuffle, and feed these paired text–image examples into our BERT and ViT encoders for testing. This also creates a conda env with the modules that we need, which will be useful for the future. Additionally, we have completed forward passes on both visual and text encoders using the MVSA dataset. The models load properly, the inputs have the correct shapes, the outputs have the right dimensions and there are no errors.

Figure 2 shows the output tensor shape from the Vision Transformer (ViT) encoder, confirming correct processing of image inputs. Figure 3 displays the output tensor shape from the BERT encoder, verifying proper handling of text inputs.

```
vit_test1653387.out
1  Using device: cuda
2  Dataset size: 19600
3  ViT Model initialized successfully
4  Model parameters: 29,885,443
5
6  Input image shape: torch.Size([2, 3, 224, 224])
7  Projection shape: torch.Size([2, 256])
8  Patch sequence shape: torch.Size([2, 196, 768])
9  Logits shape: torch.Size([2, 3])
10 Logits (sentiment predictions): tensor([[ 0.7867,  1.0441, -0.7993],
11         [ 0.8951,  1.0066, -0.5711]], device='cuda:0')
12
13 ✓ ViT architecture validated successfully!
14
```

Figure 2. Output tensor shape from the Vision Transformer (ViT) encoder on MVSA image samples.

```
bert_test1657223.out
1  Using device: cuda
2  Dataset size: 19600
3  BERT Text Encoder initialized successfully
4  Model parameters: 109,482,240
5  Input IDs shape: torch.Size([2, 33])
6  Attention mask shape: torch.Size([2, 33])
7  Pooled output shape: torch.Size([2, 768])
8  Hidden state shape: torch.Size([2, 33, 768])
9  Projected embedding shape: torch.Size([2, 256])
10 Logits shape: torch.Size([2, 3])
11 Logits (sentiment predictions): tensor([[0.2614, 0.0285, 0.3414],
12         [0.1771, 0.0119, 0.2079]], device='cuda:0')
13
14 ✓ BERT text encoder validated successfully on MVSA!
```

Figure 3. Output tensor shape from the BERT encoder on MVSA text samples.

6. Discussion

Our work started with setting up the data and environment in WPI's Turing cluster environment. We implemented the Vision Transformer and BERT encoders and verified the forward passes on the MVSA dataset. Next steps involve

integrating the fusion transformer module, adding the classification head, and finally training the complete model. While training the model we will monitor validation accuracy and loss to change hyperparmeters as needed to reduce overfitting and improve generalization. Once the model is trained, we will compare its performance against the CTMWA baseline and test our own image-text pairs outside the dataset to evaluate real-world applicability.

References

Niu, T., Zhu, S. A., Pang, L., and Saddik, A. El. Sentiment analysis on multi-view social data. In *MultiMedia Modeling (MMM)*, pp. 15–27, Miami, 2016.

Ren, J. Multimodal sentiment analysis based on bert and resnet. *arXiv preprint arXiv:2412.03625*, 2024.

Zhang, B., Yuan, Z., Xu, H., and Gao, K. Crossmodal translation based meta weight adaption for robust image-text sentiment analysis. *IEEE Transactions on Multimedia*, 26:9949–9961, 2024. doi: 10.1109/TMM.2024.3405662.