Multimodal Movie Genre Classification using Image Posters and Plot Summaries

Supkhankulov Andrey

May 2025

Abstract

This report details the development and evaluation of a multimodal deep learning model for movie genre classification. The model leverages both visual information from movie posters and textual information from plot summaries. It employs pre-trained ResNet50 and BERT models as backbones, with several enhancements such as SwiGLU activation in projection and classifier heads, partial fine-tuning of backbones, Focal Loss for handling class imbalance, label smoothing for regularization, and a learning rate warmup schedule. The model was trained and evaluated on a modified version of the MM-IMDb dataset. This document describes the problem, related work in multimodal learning, the model architecture, dataset characteristics, experimental setup, and achieved results, comparing them with state-of-the-art approaches on the MM-IMDb dataset. Project code: `https://github.com/sup41kkk/NLP-course`.

1 Introduction

Movie genre classification is a challenging task that can benefit significantly from incorporating information from multiple modalities, such as visual cues from posters and textual narratives from plot summaries. Accurate genre prediction enhances recommendation systems, content categorization, and user experience. While unimodal approaches exist, multimodal models that can effectively fuse information from different sources often achieve superior performance. This project focuses on developing a robust multimodal classifier for movie genres. The uniqueness of the approach lies in the specific combination of modern neural network components, including the use of SwiGLU activation functions for improved feature transformation, a carefully designed fine-tuning strategy for pre-trained image (ResNet50) and text (BERT) encoders, and advanced training techniques like Focal Loss to address genre imbalance, label smoothing to prevent overconfidence, and a learning rate warmup schedule for more stable convergence. The goal is to effectively combine visual and textual features to predict multiple genres for a given movie.

1.1 Team

Supkhankulov Andrey was responsible for data preprocessing, model architecture design, implementation, training, evaluation, and report preparation.

2 Related Work

The task of multimodal classification, particularly for media like movies, has seen various approaches. Key areas of related work include multimodal representation learning, graph neural networks for relational data, and attention mechanisms.

Multimodal Representation Learning: Combining information from different modalities (e.g., text and image) is crucial. Joint representation is a popular method where modality vectors are combined, often by concatenation, into a single vector to learn a shared semantic subspace, providing richer contexts [Guo et al., 2019]. Fusion methods can be categorized based on when modalities are combined [Bayoudh et al., 2021]:

- **Early fusion**: Fuses data before feature extractors or classifiers to preserve original feature richness [Sun et al., 2018]. The Multimodal Bitransformer (MMBT) is an example that employs early fusion by extending BERT-style tokenization to the image modality [Kiela et al., 2020].
- **Late fusion**: Fuses data after extracting features from separate modalities [Bayoudh et al., 2021].
- **Hybrid fusion**: Uses both early and late fusion at different points in the architecture [Bayoudh et al., 2021].

Previous works have shown that multimodal representations outperform unimodal ones in tasks like classification, but often one modality (like text) contributes more significantly than others [Arevalo et al., 2017]. This suggests that improper usage of the image modality can be a limitation [Seo et al., 2022].

Graph Neural Networks (GNNs): GNNs are designed to generate node embeddings by passing messages between nodes in a graph $G = (V, E)$.

- **Vanilla GNNs** average neighbor messages [Kipf and Welling, 2017].
- **Graph Convolution Networks (GCNs)** improve upon this using symmetric normalization and spectral-based convolution [Kipf and Welling, 2017]. However, their reliance on a fixed graph can limit generalization [Wu et al., 2021].
- **GraphSAGE** is a spatial-based model that enables inductive generalization by concatenating a node's previous hidden state with an aggregated representation of its local neighbors [Hamilton et al., 2017].

Attention Mechanisms: Attention mechanisms compute a probability distribution over an encoder's hidden states based on a decoder's current state, assigning more importance to relevant parts of the input [Luong et al., 2015, Bahdanau et al., 2015].

- **Self-attention** computes a weighted average of input vectors [Vaswani et al., 2017]. This is a core component of Transformers like BERT [Devlin et al., 2019].
- **Graph Attention Networks (GATs)** apply attention to neighbor nodes in a graph, assigning different importance to different neighbors by learning attention coefficients α_{ij} [Vaswani et al., 2017, Veličković et al., 2018].

MM-GATBT Approach: A recent advanced approach, MM-GATBT (Multimodal Graph Attention Network with Bitransformer), proposes constructing a multimodal entity graph where nodes represent movie entities and edges represent shared features (e.g., same producer, director) [Seo et al., 2022]. It uses GAT to learn image-based node embeddings that capture relational semantics and then fuses these with text embeddings using an MMBT-like architecture [Seo et al., 2022, Kiela et al., 2020]. This method aims to overcome limitations of models that do not capture interactions among entities across modalities [Seo et al., 2022] and has achieved state-of-the-art results on the MM-IMDb dataset [Seo et al., 2022]. Other notable multimodal models for this dataset include GMU [Arevalo et al., 2017], CentralNet [Vielzeuf et al., 2018], MFM [Braz et al., 2021], and ReFNet [Sankaran et al., 2022]. The approach in this project shares the goal of multimodal fusion but uses a different architecture without the explicit graph construction between movie entities.

3 Model Description

The model developed in this project is a Multimodal Classifier designed to predict movie genres based on image posters and textual plot summaries. It integrates features from both modalities using separate encoders and then fuses them for classification.

3.1 Architecture

The core components of the `MultimodalClassifier` are:

1. **Image Backbone (ResNet50)**:

   A pre-trained ResNet50 model (weights from ImageNet1K V2) is used as the image encoder. The final fully connected layer (`fc`) is replaced with an identity layer to extract feature vectors (2048 dimensions). For fine-tuning, a strategy of partial unfreezing is employed, where the last N blocks (e.g., `layer4`) of ResNet50 can be unfrozen, while earlier layers remain frozen to retain general features learned from ImageNet. In the experiments, the last block (`layer4`) was unfrozen.

2. **Text Backbone (BERT)**:

   A pre-trained BERT model (`bert-base-uncased`) serves as the text encoder. It processes tokenized plot summaries (input IDs and attention masks) and outputs contextualized embeddings. The `pooler_output` (768

dimensions) is typically used as the aggregate representation of the text. Similar to the image backbone, partial fine-tuning is applied. The last M encoder layers, the pooler layer, and the embedding layers of BERT are unfrozen to adapt them to the specific task of plot understanding for genre classification. In the experiments, the last 2 BERT encoder layers, the pooler, and embeddings were unfrozen.

3. **Projection Layers (SwiGLU)**:

   Both image and text features are passed through separate projection layers to map them to a common PROJECTION_DIM (set to 512). These projection layers utilize the SwiGLU activation function, which is a variant of Gated Linear Units (GLU) using SiLU (Sigmoid-weighted Linear Unit) as the activation. The SwiGLU layer is defined as:

$$\text{SwiGLU}(x; W_1, W_3, W_2) = \big(\text{SiLU}(xW_1) \odot (xW_3)\big) W_2, \tag{1}$$

   where W_1, W_3 project to a hidden dimension (controlled by `ffn_expansion_factor`, default $8/3$ of input) and W_2 projects to the output dimension. This structure allows for dynamic, input-dependent gating of information.

4. **Feature Fusion and Self-Attention**:

   The projected image and text features are concatenated, resulting in a fused feature vector of dimension PROJECTION_DIM $\times$ 2. This fused representation is then processed by a self-attention mechanism (specifically, `nn.MultiheadAttention` with 8 heads and dropout) to model interactions between the combined multimodal features. A LayerNorm is applied after the attention layer, incorporating a residual connection:

$$\text{AttnOutput} = \text{LayerNorm}\big(\text{SelfAttention}(F_{\text{fused}}) + F_{\text{fused}}\big). \tag{2}$$

5. **Classifier Head (SwiGLU)**:

   The output from the self-attention layer is passed to the classifier head. This head first applies LayerNorm, then another SwiGLU FFN layer (reducing dimensionality, e.g., to $\frac{1}{2}\times$fused_dim), followed by dropout, and finally a linear layer that outputs logits for each of the NUM_CLASSES (26 genres in this case):

$$\text{Logits} = \text{Linear}\Big(\text{Dropout}\big(\text{SwiGLU}(\text{LayerNorm}(\text{AttnOutput}))\big)\Big). \tag{3}$$

The model uses a dropout rate of 0.3 in the classifier head for regularization. The total number of parameters in the configured model is approximately 173.5 million, with about 94.1 million being trainable due to the partial unfreezing strategy.

4 Dataset

The primary dataset used for this project is the Multimodal IMDb (MM-IMDb) dataset, originally presented by Arevalo et al. [Arevalo et al., 2017]. This dataset is commonly used for multimodal classification tasks, specifically movie genre prediction. It can be accessed via Hugging Face Datasets: `https://huggingface.co/datasets/sxj1215/mmimdb`.

4.1 Dataset Characteristics

Each entry in the MM-IMDb dataset corresponds to a movie and typically includes:

- **Image**: A movie poster.
- **Text**:
  - A plot summary or description.
  - Additional features such as director, producer, writer, etc. (though the notebook primarily focuses on plot for text encoding).
- **Labels**: A list of genres associated with the movie. This is a multi-label classification task, as a movie can belong to multiple genres.

The notebook loads the 'sxj1215/mmimdb' version from Hugging Face, which contains a 'train' split of 15,552 samples. This 'train' split was further divided into training and validation sets with a ratio of 0.15 for validation, resulting in 13,219 samples for training and 2,333 for validation. A random seed (42) was used for reproducibility.

During genre extraction from the dataset messages, 26 unique genres were dynamically identified and used for the 'MultiLabelBinarizer'. These genres are: 'action', 'adventure', 'animation', 'biography', 'comedy', 'crime', 'documentary', 'drama', 'family', 'fantasy', 'film-noir', 'history', 'horror', 'music', 'musical', 'mystery', 'news', 'reality-tv', 'romance', 'sci-fi', 'short', 'sport', 'talk-show', 'thriller', 'war', 'western'.

The PDF paper by Seo et al. [Seo et al., 2022] on MM-GATBT, which also uses MM-IMDb, mentions a dataset size of 23,351 movie entities, split into 15,552 training and 7,799 testing samples [Arevalo et al., 2017, Jin et al., 2021]. They preprocess by dropping "News" and "Adult" labels, leading to 15,513 training and 7,779 testing entities, and test on 23 distinct labels [Arevalo et al., 2017]. The notebook's dynamic genre extraction differs slightly.

4.2 Data Preprocessing

1. **Text Preprocessing**:
   - Plot summaries are extracted from the `messages` field using regular expressions. If no plot is found, the default text *"No plot found."* is used.

- The extracted plot is tokenized with the BERT tokenizer (`bert-base-uncased`).
- Token sequences are padded or truncated to length

$$\text{MAX\_TEXT\_LENGTH} = 128.$$

2. **Image Preprocessing**:

   Movie posters are resized to

$$\text{IMG\_SIZE} = (224,\ 224).$$

   - **Training images** are augmented with:
     - `RandomResizedCrop`(scale=0.7–1.0)
     - `RandomHorizontalFlip`(probability=0.5)
     - `ColorJitter`(brightness=0.3, contrast=0.3, saturation=0.3, hue=0.15)
     - `RandomRotation`(degrees=20)
     - `RandomAffine`(translate=0.1, scale=0.9–1.1, shear=10°)
   - **Validation images** are simply resized to IMG_SIZE.
   - All images are converted to RGB, then to tensors, and normalized using ImageNet statistics:

$$\text{mean} = [0.485, 0.456, 0.406], \quad \text{std} = [0.229, 0.224, 0.225].$$

   - If an image fails to load (e.g., is corrupted), a zero tensor of shape $(3, 224, 224)$ is returned.

3. **Label Preprocessing**:

   - Genre labels are extracted from the `messages` field as lists of strings.
   - These lists are binarized into multi-hot vectors via `sklearn.preprocessing.MultiLabelBinarizer`, fitted on the full set of genres.

4.3 Dataset Class and DataLoader

```
class MMIMDbDataset(torch.utils.data.Dataset):
    def __init__(self, df, tokenizer, transform):
        self.df = df.reset_index(drop=True)
        self.tokenizer = tokenizer
        self.transform = transform
        self.mlb = MultiLabelBinarizer().fit(df['genres'].tolist())

    def __len__(self):
        return len(self.df)

    def __getitem__(self, idx):
        row = self.df.iloc[idx]
```

```
        # Text
        plot = row['plot'] or "No plot found."
        tokens = self.tokenizer(
            plot,
            padding='max_length',
            truncation=True,
            max_length=128,
            return_tensors='pt'
        )
        # Image
        img = load_image(row['image_path'])
        if img is None:
            img_tensor = torch.zeros(3, 224, 224)
        else:
            img_tensor = self.transform(img)
        # Labels
        label = self.mlb.transform([row['genres']])[0]
        return {
            'input_ids': tokens.input_ids.squeeze(),
            'attention_mask': tokens.attention_mask.squeeze(),
            'image': img_tensor,
            'labels': torch.tensor(label, dtype=torch.float)
        }

# Transforms
train_transform = transforms.Compose([
    transforms.RandomResizedCrop(224, scale=(0.7,1.0)),
    transforms.RandomHorizontalFlip(0.5),
    transforms.ColorJitter(0.3,0.3,0.3,0.15),
    transforms.RandomRotation(20),
    transforms.RandomAffine(degrees=0, translate=(0.1,0.1), scale=(0.9,1.1), shear=10),
    transforms.ToTensor(),
    transforms.Normalize([0.485,0.456,0.406],[0.229,0.224,0.225])
])
val_transform = transforms.Compose([
    transforms.Resize((224,224)),
    transforms.ToTensor(),
    transforms.Normalize([0.485,0.456,0.406],[0.229,0.224,0.225])
])

# DataLoaders
train_loader = DataLoader(
    train_dataset,
    batch_size=16,
    shuffle=True,
    num_workers=2,
```

```
    pin_memory=True,
    drop_last=True
)
val_loader = DataLoader(
    val_dataset,
    batch_size=16,
    shuffle=False,
    num_workers=2,
    pin_memory=True
)
```

4.4 Experiments

4.4.1 Metrics

The model was evaluated using:

- **Macro F1-score**: Compute F1 for each class independently, then take the unweighted average.

$$\text{Macro-F1} = \frac{1}{C} \sum_{i=1}^{C} \frac{2\,\text{TP}_i}{2\,\text{TP}_i + \text{FP}_i + \text{FN}_i}.$$

- **Micro F1-score**: Compute global counts over all classes:

$$\text{Micro-F1} = \frac{2 \sum_i \text{TP}_i}{2 \sum_i \text{TP}_i + \sum_i \text{FP}_i + \sum_i \text{FN}_i}.$$

  In multi-label settings this equals overall accuracy.

Early stopping and the learning-rate scheduler were both driven by validation Macro F1. For reference, Seo et al. [Seo et al., 2022] also report Weighted F1 and Samples F1 on the MM-IMDb benchmark.

4.5 Experiment Setup

- **Hardware**: CUDA-enabled GPU.
- **Software**: PyTorch, Hugging Face Transformers, Datasets library, scikit-learn.
- **Model Configuration**:
  - Image Encoder: ResNet50 (last block unfrozen).
  - Text Encoder: BERT-base-uncased (last 2 encoder layers, pooler, embeddings unfrozen).
  - Projection Dimension:

$$\text{PROJECTION\_DIM} = 512.$$

  - Max Text Length:

$$\text{MAX\_TEXT\_LENGTH} = 128.$$

  - Image Size:

$$\text{IMG\_SIZE} = (224,\ 224).$$

- **Training Hyperparameters**:
  - Batch Size: 16.
  - Number of Epochs: 20.
  - Learning Rate (LR): initial

$$\text{LR}_0 = 1 \times 10^{-4}.$$

  - Weight Decay: 1×10^{-5}.
  - Optimizer: AdamW.
  - Loss Function: Focal Loss with $\alpha = 0.25$ and $\gamma = 2.0$:

$$\mathcal{L}_{\text{FL}} = -\alpha\,(1 - p_t)^{\gamma}\,\log p_t,$$

    where p_t is the model's estimated probability for the true class.
  - Label Smoothing: $\epsilon = 0.1$. Targets are smoothed as

$$y_i' = y_i\,(1 - \epsilon) + (1 - y_i)\,\epsilon.$$

  - Warmup: first 3 epochs with linear warmup from

$$\text{LR}_{\min} = 1 \times 10^{-6} \quad \text{to} \quad \text{LR}_{\max} = 1 \times 10^{-4}.$$

  - LR Scheduler: `ReduceLROnPlateau` on validation Macro F1 with patience = 2 and factor = 0.2, activated after warmup.
  - Early Stopping: halt if validation Macro F1 does not improve by at least MIN_DELTA = 0.001 for 5 consecutive epochs; the best checkpoint is saved.
- **Random Seed**: 42 for all random operations.

4.6 Baselines

While this project implements a specific multimodal architecture, the performance can be contextualized by comparing with unimodal baselines and other multimodal approaches reported in literature for the MM-IMDb dataset. Seo et al. [Seo et al., 2022] provide results for such baselines (referred from table [103] in their PDF):

- **Unimodal**:

  - EfficientNet (Image-only) [Tan and Le, 2019]
  - BERT (Text-only) [Devlin et al., 2019]
- **Multimodal**:
  - GMU (Gated Multimodal Units) [Arevalo et al., 2017]
  - CentralNet [Vielzeuf et al., 2018]
  - MMBT (Multimodal Bitransformer) [Kiela et al., 2020]
  - MFM (Multimodal Fusion Network) [Braz et al., 2021]
  - ReFNet [Sankaran et al., 2022]
- **Graphical**:
  - GAT w/ EfficientNet (Image-based graph features)
  - MM-GATBT (Proposed SOTA by [Seo et al., 2022])

The implemented model in this project is a multimodal classifier without an explicit graph structure between movie entities, differing from the MM-GATBT approach.

5 Results

The model was trained for 20 epochs, incorporating the techniques described. The training progress and validation performance were tracked.

5.1 Training Performance

The training loop progressed as follows:

- **Epoch 1-3 (Warmup)**: Learning rate increased linearly from 1.00×10^{-6} to 6.70×10^{-5}. Validation Macro-F1 improved from 0.0000 to 0.3948. The model checkpoint was saved at epoch 3.
- **Epoch 4**: Target LR of 1.00×10^{-4} was set. Validation Macro-F1 was 0.3324.
- **Epoch 5**: Validation Macro-F1 improved to 0.4188. Model saved.
- **Epoch 6**: Validation Macro-F1 improved to 0.4739. Model saved.
- **Epoch 8**: Validation Macro-F1 improved to 0.4792. Model saved.
- **Epoch 11**: Validation Macro-F1 improved to 0.4803. Model saved.
- **Epoch 12**: Validation Macro-F1 improved to 0.4960. Model saved.
- **Epoch 16**: LR reduced to 2.00×10^{-5} by the scheduler. Validation Macro-F1 improved to 0.5104. Model saved.

- **Epoch 17**: Validation Macro-F1 improved to **0.5230**. This was the best Macro-F1 achieved. Model saved.
- **Epochs 18-20**: Validation Macro-F1 did not surpass 0.5230. The early stopping counter increased, but did not trigger termination within 20 epochs.

The training finished after 20 epochs. The best validation Macro-F1 score achieved was **0.5230** at epoch 17. The corresponding Micro-F1 score at this epoch was 0.6390. The training loss generally decreased over epochs, starting from 0.0464 and reaching around 0.0043 in the later epochs. Validation loss initially decreased from 0.0252 (epoch 1) to 0.0166 (epoch 3), then fluctuated, increasing to 0.0858 by epoch 20, indicating some overfitting as the Macro-F1 also plateaued and slightly decreased from its peak.

The learning rate plot (not shown here, but generated by the notebook) would show the initial warmup phase for 3 epochs, then a constant LR of 1.00×10^{-4} until epoch 15, followed by a reduction to 2.00×10^{-5} for epochs 16-20. The loss history plot would show training loss consistently decreasing and validation loss decreasing initially then increasing. The F1 score plot would show validation Macro-F1 and Micro-F1 scores peaking around epoch 17.

5.2 Comparison with Other Approaches

The table below shows the results achieved by the implemented model compared to baselines and state-of-the-art (SOTA) results on the MM-IMDb dataset as reported by Seo et al. [Seo et al., 2022] (Table 1 in their paper, referred as [103], [105] in their text). Note that "Our Model (Notebook)" is the model implemented in this project, which is different from "MM-GATBT (ours)" from the cited paper.

The implemented model achieves a Macro-F1 of 0.523 and a Micro-F1 of 0.639.

- Compared to unimodal baselines, our model significantly outperforms the image-only EfficientNet (Macro-F1 0.314) and performs slightly lower than the text-only BERT (Macro-F1 0.587) in terms of Macro-F1. Its Micro-F1 (0.639) is comparable to BERT's (0.645).
- Compared to other multimodal approaches reported by Seo et al. [Seo et al., 2022], our model's Macro-F1 (0.523) is lower than GMU (0.541), CentralNet (0.561), MMBT (0.618), and MFM (0.616). Its Micro-F1 (0.639) is comparable to GMU (0.630) and CentralNet (0.639), but lower than MMBT (0.669), MFM (0.675), and ReFNet (0.680).
- The SOTA model MM-GATBT [Seo et al., 2022] reports significantly higher scores (Micro-F1 0.685, Macro-F1 0.645). This highlights the benefit of its graph-based architecture which captures inter-entity relationships.

Model Type	Model	Micro F1	Macro F1	Weighted F1
Our Model (Notebook)	**Multimodal Classifier**	**0.639**	**0.523**	-
Unimodal	EfficientNet ([Tan and Le, 2019])	0.395	0.314	0.457
	BERT ([Devlin et al., 2019])	0.645	0.587	0.645
Multimodal	GMU ([Arevalo et al., 2017])	0.630	0.541	0.617
	CentralNet ([Vielzeuf et al., 2018])	0.639	0.561	0.631
	MMBT ([Kiela et al., 2020])	0.669	0.618	-
	MFM ([Braz et al., 2021])	0.675	0.616	-
	ReFNet ([Sankaran et al., 2022])	0.680	0.587	-
Graphical	GAT w/ EfficientNet	0.500	0.394	0.506
	MM-GATBT ([Seo et al., 2022])	**0.685**	**0.645**	**0.683**

Table 1: Comparison of F1 scores on the MM-IMDb dataset. '-' indicates not reported in the source table. "Our Model (Notebook)" results are from the validation set of the implemented notebook. Other results are from Table 1 of [Seo et al., 2022].

The results suggest that while the implemented multimodal fusion provides an improvement over image-only models, further architectural refinements or techniques like those in MM-GATBT are needed to reach SOTA performance, especially in Macro-F1 which is sensitive to performance across all classes. The increasing validation loss towards the end of training also suggests that further regularization or earlier stopping might be beneficial.

6 Conclusion

This project successfully implemented a multimodal classifier for movie genre prediction, integrating visual features from ResNet50 and textual features from BERT. The architecture incorporated SwiGLU activations, partial fine-tuning of backbones, and several advanced training strategies including Focal Loss, label smoothing, learning rate warmup, and early stopping based on validation Macro F1-score.

The model was trained on the MM-IMDb dataset, achieving a best validation Macro F1-score of 0.5230 and a Micro F1-score of 0.6390. While these results demonstrate effective multimodal fusion compared to unimodal image-based approaches, they are below the performance of text-only BERT and other more advanced multimodal and graph-based models like MM-GATBT on this dataset. The increasing validation loss in later epochs also indicated a need for potentially stronger regularization or more fine-tuned early stopping.

Future work could explore incorporating graph-based relational information between movie entities, similar to the MM-GATBT approach, or experimenting with different fusion mechanisms and attention patterns to further enhance the

model's ability to capture complex inter-modal and intra-modal relationships for improved genre classification.

References

[Arevalo et al., 2017] Arevalo, J., Solorio, T., Montes-y Gómez, M., and González, F. A. (2017). Gated multimodal units for information fusion. *arXiv preprint arXiv:1702.01992*.

[Bahdanau et al., 2015] Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[Bayoudh et al., 2021] Bayoudh, K., Knani, R., Hamdaoui, F., and Mtibaa, A. (2021). A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets. *The Visual Computer*, pages 1–29.

[Braz et al., 2021] Braz, L., Teixeira, V., Pedrini, H., and Dias, Z. (2021). Image-text integration using a multimodal fusion network module for movie genre classification. In *2021 11th International Conference of Pattern Recognition Systems (ICPRS)*, pages 200–205. IEEE.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

[Guo et al., 2019] Guo, W., Wang, J., and Wang, S. (2019). Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394.

[Hamilton et al., 2017] Hamilton, W. L., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in neural information processing systems (NIPS)*, pages 1024–1034.

[Jin et al., 2021] Jin, W., Sanjabi, M., Nie, S., Tan, L., Ren, X., and Firooz, H. (2021). MSD: saliency-aware knowledge distillation for multimodal understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic, 7-11 November 2021*, pages 3557–3569.

[Kiela et al., 2020] Kiela, D., Bhooshan, S., Firooz, H., Perez, E., and Testuggine, D. (2020). Supervised multimodal bitransformers for classifying images and text. *CoRR*, abs/1909.02950.

[Kipf and Welling, 2017] Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

[Luong et al., 2015] Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 conference on empirical methods in natural language processing (EMNLP)*, pages 1412–1421.

[Sankaran et al., 2022] Sankaran, S., Yang, D., and Lim, S. (2022). Refining multimodal representations using a modality-centric self-supervised module. *CoRR*, abs/2201.00872.

[Seo et al., 2022] Seo, S. B., Nam, H., and Delgosha, P. (2022). MM-GATBT: Enriching multimodal representation using graph attention network. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, pages 106–112.

[Sun et al., 2018] Sun, H., Dhingra, B., Zaheer, M., Mazaitis, K., Salakhutdinov, R., and Cohen, W. W. (2018). Open domain question answering using early fusion of knowledge bases and text. pages 4231–4242.

[Tan and Le, 2019] Tan, M. and Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning (ICML)*, pages 6105–6114. PMLR.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems (NIPS)*, 30.

[Veličković et al., 2018] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. In *International Conference on Learning Representations (ICLR)*.

[Vielzeuf et al., 2018] Vielzeuf, V., Lechervy, A., Pateux, S., and Jurie, F. (2018). Centralnet: A multilayer approach for multimodal fusion. *CoRR*, abs/1808.07275.

[Wu et al., 2021] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2021). A comprehensive survey on graph neural networks. *IEEE Trans. Neural Networks Learn. Syst.*, 32(1):4–24.