Sexism Identification Using Annotator Ranking in Memes: A Multimodal Approach Using Transformers

Deobrat Kumar Jha[1,†], Mitesh Kumar Mandal[1,†] and Anand Kumar Madasamy[1]

[1]Department of Information Technology, National Institute of Technology Karnataka Surathkal, Mangalore 575025, India

Abstract

Memes are a popular medium for sharing information on social media, often embedding humor and interactive content. However, they can also propagate sexism, targeting specific genders, particularly females. This paper presents a multimodal approach to detect sexism in memes and classify the intent of sexist memes and sexism categorization. We leverage BERT for textual analysis, BLIP for multimodal processing, and Vision Transformers (ViT) for image feature extraction. Our model achieves approximately 68.49% accuracy in identifying sexist memes and 68.52% accuracy in determining the source intention and 49.31% accuracy in Sexism Categorization. This work contributes to creating safer digital spaces by automating the detection of biased content on social media.

Keywords

BERT, BLIP, Memes, Social Media, Sexism, Vision Transformers

1. Introduction

Social networking platforms have emerged as powerful tools for advocacy, awareness, and activism, amplifying issues like sexism through movements such as #MeToo and #Time'sUp. However, these platforms also harbor sexist ideologies and misogynistic rhetoric, necessitating automated tools for detection. This paper addresses the challenge of identifying sexism in memes, a prevalent form of multimedia content on social media, using multimodal machine learning techniques.

The EXIST 2024 campaign provided datasets for meme analysis, advancing sexism detection beyond text to include visual elements. Our work extends this by developing a model that detects sexist memes, classifies their intent (judgmental or direct) and Sexism Categorization (Five Categories) in Memes using BERT, BLIP, and ViT. We also employ the Learning With Disagreement (LeWiDi) framework to handle subjective annotations, enhancing model fairness.

Our research offers three main task directions: (1) to determine wheather a meme contains sexism, (2) to classify sexist memes as Judgmental or Direct and (3) Sexism Categorization in five categories. With extensive training and optimization, the performance of our system achieved 68.49% for sexism detection, 68.52% for intent identification of memes and 49.31% for sexism Categorization. These results attest to the power of our multimodal approach, supplementing the ongoing activity on the automated detection of bias on social media.

2. Literature Review

In recent years, memes have gained immense popularity as a means of communication on social media. They encapsulate sentiments, humor, and opinions, making them an effective tool for spreading messages, including both positive and negative discourse. The field of automated meme analysis has been explored extensively through sentiment analysis, multimodal learning, and computer vision

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

[†]These authors contributed equally.

deobratkumarjha.242it008@nitk.edu.in (D. K. Jha); miteshkumar.242it020@nitk.edu.in (M. K. Mandal); m_anandkumar@nitk.edu.in (A. K. Madasamy)

0009-0007-7764-2369 (D. K. Jha); 0009-0002-8382-8374 (M. K. Mandal); 0000-0003-0310-4510 (A. K. Madasamy)

techniques. This literature survey reviews key research contributions relevant to our work on detecting sexism in memes and the source intention of the memes.

A study is conducted on the sentiment analysis of text memes using various supervised machine learning models. Their research highlights that memes often contain sentiments toward specific issues, individuals, or entities, and their classification requires effective text-processing techniques. The study compared Naïve Bayes, Support Vector Machines (SVM), Decision Trees, and Convolutional Neural Networks (CNN) for analyzing Indonesian text memes. Among these models, Naïve Bayes demonstrated the highest accuracy of 65.4% in classifying meme sentiment. This study underscores the significance of textual feature extraction in meme classification and provides a baseline for incorporating machine learning techniques in sentiment analysis.

Another technique is presented in which a multimodal approach to sentiment analysis of memes that integrates both visual and linguistic components is used. Their study focused on analyzing Hinglish and English memes using a dataset of 3999 labeled memes. They explored the effectiveness of various models, including RoBERTa, CLIP, BERT, SVM, Multinomial Naïve Bayes, and VADER, in identifying sentiments as positive, negative, or neutral. The RoBERTa-CLIP combination yielded the highest accuracy of 82%, significantly outperforming traditional sentiment analysis models such as BERT (64%), SVM (42%), and Naïve Bayes (34%). Their findings emphasize the importance of incorporating both textual and visual features for a comprehensive understanding of meme sentiment, which is highly relevant to our approach in detecting sexism in memes.

The Vision Transformer (ViT) model and its role in image analysis was investigated in a study [1]. The study delves into the four core components of ViT—patch division, token selection, position encoding, and attention calculation—that enhance its capability in visual processing. A review of ViT applications across various domains, including medical image processing and object detection, provides insights into how advanced deep learning architectures can be leveraged for meme classification. Given that memes contain both text and images, ViT's powerful feature extraction mechanisms are crucial for understanding their context and detecting underlying biases.

A study [2] explored the detection of extreme sentiments on social networks using BERT. The work builds on previous studies that classified social media posts as positive, negative, or neutral by refining sentiment classification using a semi-supervised approach. The study demonstrated that many posts classified as extremely positive or negative indeed carried heightened sentiments when analyzed with BERT, proving its effectiveness in fine-tuning sentiment detection. This work is relevant for identifying extreme opinions that contribute to sexism and hate speech in memes.

A scalable harmful meme detection framework using Graph Neural Networks (GNNs) was introduced in a study [3], incorporating both invariant and specific modality representations. The method enhances cross-modal interaction by projecting visual and textual data into distinct spaces to address the modality gap. This approach significantly improves harmful meme classification by dynamically balancing inter-modal and intra-modal relationships. The study highlights an effective method for detecting harmful memes, aligning with efforts to detect sexist content in memes.

Another study [4] focused on hate speech detection in social media memes using machine learning. It analyzed the challenges associated with detecting hate speech in visual content, emphasizing the need for automatic detection mechanisms to prevent hate speech propagation. The researchers utilized Facebook AI's hateful meme dataset to evaluate unimodal and multimodal approaches, highlighting the complexities in meme analysis. The findings underscore the necessity of robust multimodal techniques for effective hate speech and bias detection in social media content.

Recent advancements in natural language processing and deep learning have enabled significant progress in understanding and analyzing internet memes, which typically combine text and visual components to convey humor, opinions, or controversial content.

A deep learning-based approach was proposed that utilizes both textual and visual features to detect and classify memes. This multi-modal architecture not only outperformed traditional unimodal models but also tracked the evolution of memes during political events, shedding light on how memes transform and propagate online [5].

To address the complex semantics of memes, a large-scale dataset was introduced that focuses on

metaphor usage within meme content. The dataset includes annotations for sentiment, intent, metaphor type, and offensiveness. The study demonstrated that incorporating metaphor analysis significantly enhances the performance of models tasked with meme sentiment classification [6].

In efforts to detect offensive memes, a multi-step pipeline was developed. It involved extracting embedded text using OCR, classifying its offensive nature using a GRU-based model, and categorizing the level of offense. This approach highlights the importance of automated tools in identifying harmful content at scale [7].

Focusing on emotional interpretation, transformer-based models like BERT were applied to meme sentiment classification as part of a benchmark challenge. These models showed superior performance in detecting emotions such as sarcasm and humor compared to older LSTM-based models [8].

A novel framework enhanced meme detection by utilizing a vision transformer that emphasizes important visual regions. By introducing visual part utilization and attention mechanisms, this model excelled at distinguishing memes from non-meme images, especially in complex scenarios [9].

Beyond memes, robust deep learning techniques have improved hate speech detection. One study categorized tweets into subtypes such as racism and sexism using neural models, outperforming traditional machine learning methods [10]. However, another study revealed the limitations of current models, showing their vulnerability to adversarial attacks like text obfuscation or insertion of neutral words [11].

HateBERT is a domain-adapted BERT model retrained on 1.4M Reddit posts from communities banned for offensive content. It effectively captures the linguistic patterns of hate and abuse, outperforming general BERT models in tasks like offensive, abusive, and hate speech detection. Its robustness and cross-domain portability make it ideal for social media toxicity analysis in research contexts [12].

Additionally, distributed comment embeddings were employed to detect abusive language on online platforms. These embeddings capture semantic context while reducing dimensionality, leading to efficient and scalable classification of toxic content [13].

3. Methodology

This section describes the methodology used for meme classification, leveraging multi-modal deep learning techniques. The approach integrates visual and textual feature extraction using BLIP (Bootstrapping Language-Image Pretraining), BERT (Bidirectional Encoder Representations from Transformers) and ViT (Vision transformer) followed by a fusion mechanism employing attention-based. Then we used multi-layer perceptron (MLP) for the final classification.

3.1. Data Processing Workflow

The process begins with loading a JSON file containing descriptions of memes, as outlined in Algorithm 1 and illustrated in Figure 1. This file is checked for correct formatting and converted into a pandas DataFrame. The dataset is then examined to remove irrelevant columns that do not contribute to the analysis, streamlining further processing. Next, the data is divided into two categories: **Non-Tie Cases**, where a clear consensus among annotators exists, and **Tie Cases**, where conflicting rankings are provided. For non-tie cases, the most frequently chosen label is assigned using a majority vote approach. In tie cases, annotator ranking is employed to resolve conflicts based on annotator reliability or predefined criteria. The outcomes from both the majority vote and tie resolution processes are then merged into a single, consistent dataset. To ensure environmental compatibility, image paths are adjusted according to the directory structure. Finally, the fully cleaned and processed dataset is exported in CSV format for easy accessibility and compatibility with various machine learning frameworks.

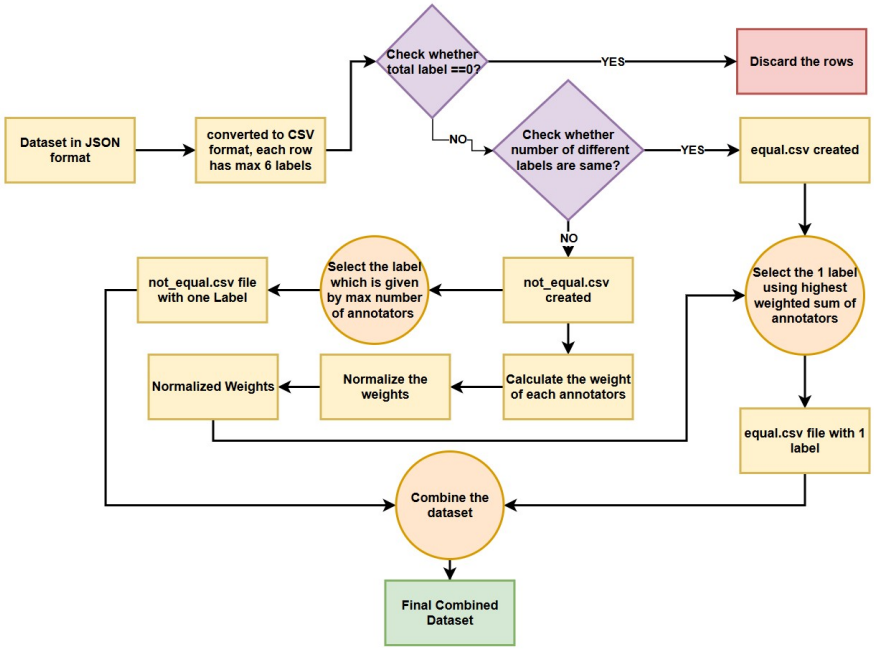

Figure 1: Overview of the Data Processing Steps

3.2. Methodology for Detecting Sexism in Memes (Subtask 1)

3.2.1. Overview

To detect whether a meme is sexist or not, we use both the image and the text in the meme. First, we extract features from the image and generate a caption using a model called BLIP. At the same time, the actual meme text is processed using BERT, which helps us understand the meaning of the text. BLIP takes care of the image part—it extracts the main details from the image and writes a short caption describing it. Both the original text and the generated caption are then passed through BERT to get meaningful text representations. To make these text features stronger, we apply attention pooling, which helps the system focus on the important words. Finally, we combine everything—the image features, the meme text, and the caption—into one complete set of features. This combined data goes into an MLP classifier, which predicts whether the meme is sexist. We train this whole model using standard methods like cross-entropy loss and the Adam optimizer to get the best performance

The flow diagram of the methodology for this subtask is shown in Figure 2. The subsequent sections elaborate on each stage of the process in greater detail.

3.2.2. Input Data Processing

The system takes two main inputs: the meme image and the meme text. Feature extraction is performed on the meme image, followed by caption generation. The texts are processed independently from the images.

3.2.3. Image Pre-processing using BLIP

All input images are preprocessed for compatibility with the BLIP model. The images are converted to RGB, resized, and normalized to the dimensions required by BLIP's Vision Transformer (ViT). These processed images are then converted into tensor representations to serve as input to the BLIP model.

Algorithm 1 Annotator Ranking and Label Assignment

Input: Dataset D with columns: annotator IDs, labels, final labels; tie-case dataset D_tie
Output: Ranked annotator dictionary, updated D_tie with resolved labels
Initialize empty dictionaries `correct_counts` and `total_counts`
for each record in D **do**
    Get `annotators`, `labels`, and `final_label`
    for each `annotator`, `label` pair **do**
        if `label` is valid (not '-' or null) and `final_label` is valid **then**
            Increment `total_counts[annotator]`
            if `label` = `final_label` **then**
                Increment `correct_counts[annotator]`
            end if
        end if
    end for
end for
Initialize empty dictionary `annotator_accuracy`
for each `annotator` in `total_counts` **do**
    `annotator_accuracy[annotator]` ←
        `correct_counts[annotator]`/ `total_counts[annotator]`
end for
Sort `annotators` by `accuracy` in descending order
Create `annotator_rank` dictionary: assign rank (1 to n) based on sorted order
for each record in D_tie **do**
    Get `annotators` and `labels`
    Create list of (`rank`, `label`) pairs for valid `labels` (not '-' or null), using `annotator_rank`
    if list is not empty **then**
        Sort pairs by `rank` (ascending)
        Assign first `label` (from lowest rank) to `final_label`
    else
        Assign `final_label` ← null
    end if
end for
Return `annotator_rank`, updated D_tie

3.2.4. Text Pre-processing using BERT

BERT handles meme text processing independently. The meme text is tokenized using the BERT tokenizer for uniform representation. Special tokens like [CLS] and [SEP] are included for better semantic encoding. These tokens are then converted to tensors along with attention masks to support variable-length inputs.

3.2.5. BLIP: Image Feature Extraction and Caption Generation

The BLIP model consists of two major components: a Vision Encoder (ViT) and a Text Decoder. The Vision Encoder extracts visual features from the meme image, and the Text Decoder generates a descriptive caption of the image. For extracting image features, the image is processed by ViT to produce visual embeddings, and the CLS token output from the final transformer layer is used as the image representation. For caption generation, the image is passed through BLIP's decoder to generate a caption, which represents the semantic meaning of the image and is then passed through BERT for embedding.

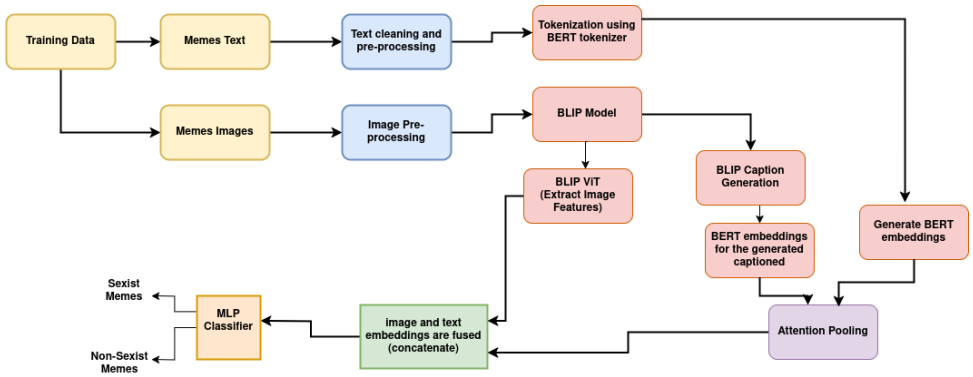

Figure 2: Overview of the Training Methodology for finding sexism in the memes (Subtask 1)

3.2.6. BERT: Textual Embedding for Meme Text and Generated Captions

BERT is applied to two textual inputs: the raw meme text and the generated captions. For meme text embedding, the raw meme text is tokenized using the BERT tokenizer, encoded using BERT, and the [CLS] token embedding is extracted. Similarly, for caption text embedding, the BLIP-generated caption is tokenized and encoded, and its [CLS] token embedding is extracted as the caption's representation.

3.2.7. Attention Pooling for Feature Aggregation

To improve text feature representation, attention pooling is used. This mechanism assigns different weights to token embeddings based on their relevance and aggregates these embeddings into a single vector for each text source. Attention pooling is applied to BERT's hidden states for both the meme text and the captions, resulting in one aggregated vector per input source.

3.2.8. Feature Fusion: Combining Image and Text Embeddings

All extracted features are concatenated into a single representation, comprising image features, meme text embedding, and caption embedding. This final representation captures visual semantics from BLIP-ViT, text semantics from the raw meme text, and generated caption semantics.

3.2.9. MLP Classifier for Final Prediction

A Multi-Layer Perceptron (MLP) processes the fused feature vector. The MLP consists of one or more fully connected layers, uses ReLU activation for non-linearity, and includes dropout layers to prevent overfitting. The final classification is performed via a softmax layer.

3.2.10. Training and Optimization

To ensure optimal training, several techniques are used. Cross-entropy loss is applied for multi-class classification. The Adam optimizer is used for efficient parameter updates. Training is conducted in mini-batches to stabilize learning and improve generalization.

3.3. Methodology for Finding the source intention behind the Sexism Memes (Subtask 2)

3.3.1. Overview

This part of the project focuses on building a multi-modal model to detect sexism in memes and also understand the type of intention behind them. We take both the meme text (caption) and the meme image into account. For text, we use the BERT model, and for image, we use the Vision Transformer (ViT).

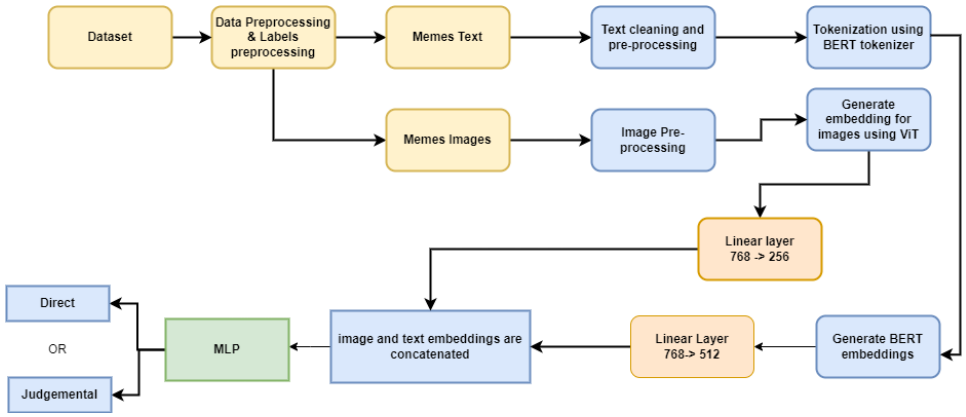

Figure 3: Overview of the Training Methodology for finding the source intention in the memes (subtask 2)

As shown in Figure 3, we first preprocess the dataset to separate meme texts and images. The text goes through cleaning, tokenization using BERT, and embedding generation. Similarly, the image is preprocessed and passed through ViT to get image embeddings. These embeddings are then reduced using linear layers.

After that, both text and image embeddings are combined and passed through a simple MLP model to predict the intention behind the meme. The output is one of the two categories:

- **DIRECT**
- **JUDGEMENTAL**

This combined method helps the model understand both what is written and what is shown in the meme, which improves accuracy in detecting sexism. The subsequent sections elaborate on each stage of the process in greater detail.

3.3.2. Input Data Processing

Dataset Preparations is the same as Task 1. So, we only need to change the label since both are binary classifications.

3.3.3. Model Architecture

In our Model architecture, we first process the text and image separately using separate tokenizer and embedder. Later we concatenate them and feed them into a Multi-Layer Perceptron for classification.

BERT (Bidirectional Encoder Representations from Transformers) for text analyzing We apply BERT-base-uncased, which is a transformer model that utilizes attention mechanisms to process text data into a contextualized representation.

- The text input is tokenized and sent through BERT's embedding, mapping each word to a high-dimensional space.
- BERT's transformer layers produce contextualized embeddings, which create semantic relationships between words.
- We extract the last hidden state corresponding to the [CLS] token, which is a 768-dimensional vector.
- This 768-dimensional text feature is then passed through a fully connected (FC) layer that reduces the dimensionality to 512.

Vision Transformer (ViT) for Memes image analysis We make use of ViT-base-patch16-224-in21k, a model that perceives images as a succession of patches.

- Each image was delineated into 16×16 pixel patches, and each patch was reflected into an embedding space of high-dimensional objects.
- To account for the spatial information inherent in the patches, a position embedding was added.
- The transformer layers of ViT process these embeddings in a manner analogous to how BERT processes words.
- At the end of the model, the hidden state corresponding to the classification token (a 768-dimensional vector) is extracted.
- This image feature embedded in 768-dimensions is then processed by a fully connected (FC) layer that reduces the dimensionality from 768 to 256.

Concatenation of features/ embeddings extracted from above methods for text and image

- The text and image features of size 512 and 256 dimension are combined together into a vector of size 768 dimension.
- This fused representation is then run through a Multi Layer Perceptron for classification.
- The output is a vector of logits with two values corresponding to the probability scores of the DIRECT and JUDGEMENTAL classes.
- The predicted class is the class that has the highest score.

3.3.4. Process of Training

Loss Function For our study, we use CrossEntropyLoss, which is a standard loss function used for multi-class classification. Formally, it simply computes the difference between predicted probabilities for each class and true class labels.

Optimization We selected the AdamW optimizer, at a learning rate of $2e^{-5}$, as it is typically the optimizer of choice for fine-tuning pre-trained transformers. In addition to this, we employ weight decay as a regularizer to prevent overfitting.

Training Loop We trained the model for 5 epochs with a batch size of 16. In each iteration:

1. Text and image inputs are inputted through the separate models.
2. Features are extracted, then transformed, and finally concatenated and fused.
3. The resultant fused representation is classified, and a loss value is computed.
4. The loss value is used to propagate a weight update according to both the loss function and optimizer.

We monitored potential improvements in model performance using the average loss per epoch.

3.4. Methodology for Sexism Categorization in Memes (Subtask 3)

3.4.1. Overview

This subtask deals with classifying sexist memes into five specific categories. Our model uses both the meme image and its text to understand the meaning better and make the final prediction.

As shown in Figure 4, we first separate the image and text from the training data. The meme text is translated to English (if needed), cleaned, and preprocessed. On the image side, we use the BLIP model to generate captions after doing basic image preprocessing.

Both the original meme text and the captions generated from the image are then passed through HateBERT. First, we tokenize the inputs, and then generate embeddings for both the text and the captions. These embeddings are combined to create one final feature.

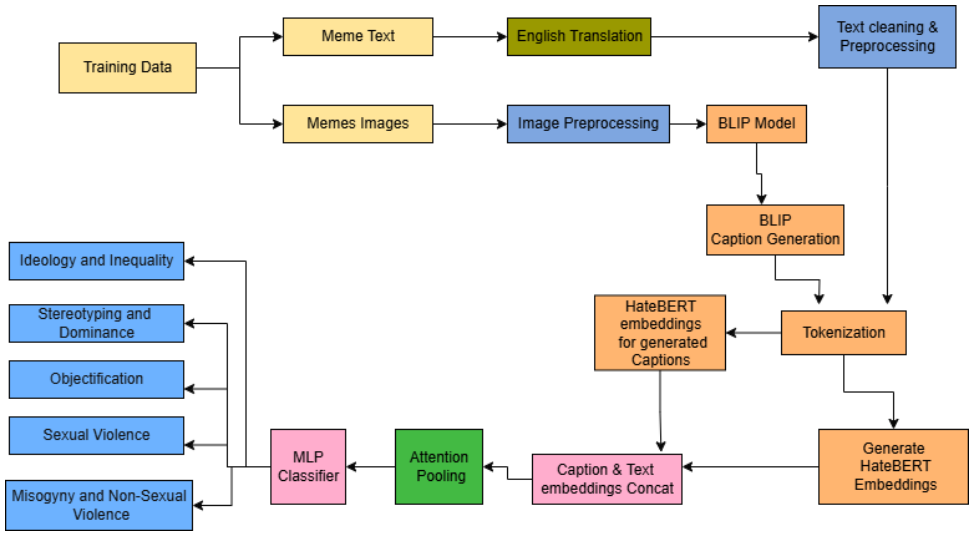

Figure 4: Overview of the Training Methodology for finding sexism in the memes (Subtask 3)

This combined feature is then passed through attention pooling and a Multi-Layer Perceptron (MLP) classifier, which predicts one of the five categories:

- **Ideology and Inequality**
- **Stereotyping and Dominance**
- **Objectification**
- **Sexual Violence**
- **Misogyny and Non-Sexual Violence**

Using both the image and text like this helps the model understand the meme more completely, especially in cases where just one of them is not enough. The subsequent sections elaborate on each stage of the process in greater detail.

3.4.2. Input Data Processing

The system takes two main inputs: the meme image and the meme text. Feature extraction is performed on the image, followed by caption generation. The texts are processed independently from the images.

3.4.3. Image Pre-processing using BLIP

All input images are preprocessed for compatibility with the BLIP model. The images are converted to RGB, resized, and normalized to the dimensions required by BLIP's Vision Transformer (ViT). These processed images are then converted into tensor representations to serve as input to the BLIP model.

3.4.4. Text Pre-processing using HateBERT

HateBERT handles meme text processing independently. The meme text is tokenized using the HateBERT tokenizer for uniform representation. Special tokens like [CLS] and [SEP] are included for better semantic encoding. These tokens are then converted to tensors along with attention masks to support variable-length inputs.

3.4.5. BLIP: Image Feature Extraction and Caption Generation

The BLIP model has two major components: a Vision Encoder (ViT) that extracts visual features from the meme image, and a Text Decoder that generates a descriptive caption of the image. For extracting image features, the image is processed by ViT to produce visual embeddings, and the CLS token output

from the final transformer layer is used as the image representation. For caption generation, the image is passed through BLIP's decoder to generate a caption, which represents the semantic meaning of the image and is then passed through HateBERT for embedding.

3.4.6. HateBERT: Textual Embedding for Meme Text and Generated Captions

HateBERT is applied to two textual inputs. For meme text embedding, the raw meme text is tokenized using the HateBERT tokenizer, encoded using HateBERT, and the [CLS] token embedding is extracted. Similarly, for caption text embedding, the BLIP-generated caption is tokenized and encoded, and its [CLS] token embedding is extracted as the caption's representation.

3.4.7. Attention Pooling for Feature Aggregation

To improve text feature representation, attention pooling is used. This mechanism assigns different weights to token embeddings based on their relevance and aggregates these embeddings into a single vector for each text source. Attention pooling is applied to HateBERT's hidden states for both the meme text and the captions, resulting in one aggregated vector per input source.

3.4.8. Feature Fusion: Combining Image and Text Embeddings

All extracted features are concatenated into a single representation comprising image features, meme text embedding, and caption embedding. This final representation captures visual semantics from BLIP-ViT, text semantics from the raw meme text, and generated caption semantics.

3.4.9. MLP Classifier for Final Prediction

A Multi-Layer Perceptron (MLP) processes the fused feature vector. The MLP consists of one or more fully connected layers, uses ReLU activation for non-linearity, and includes dropout layers to prevent overfitting. The final classification is performed via a softmax layer.

3.4.10. Training and Optimization

To ensure optimal training, several techniques are used. Cross-entropy loss is applied for multi-class classification. The Adam optimizer is used for efficient parameter updates. Training is conducted in mini-batches to stabilize learning and improve generalization.

4. Experimental Results

This section discusses the Dataset, Experimental setup and results obtained after evaluating the fine-tuned model on the training and validation data.

4.1. Dataset Description

The dataset we have used for the given tasks is provided by CLEF 2025 [14], named EXIST-Datasets. The EXIST 2025 Dataset aims to provide the research community with the first comprehensive multimedia dataset encompassing tweets, memes, and videos— for sexism detection and categorization in social media.

4.1.1. Meme Distribution

- **Total Memes**: 4044
- **English Memes (en)**: 2010
- **Spanish Memes (es)**: 2034

4.1.2. Label Distributions

Table 1
Label Distributions

Category	Values	No. of Annotators
Subtask 1: Sexism Identification in Memes	YES, NO	6
Subtask 2: Source Intention in Memes	DIRECT, JUDGEMENTAL	6
Subtask 3: Sexism Categorization in Memes	IDEOLOGICAL AND INEQUALITY, STEREOTYPING AND DOMINANCE, OBJECTIFICATION, SEXUAL VIOLENCE, MISOGYNY AND NON-SEXUAL VIOLENCE	6
Annotator Gender	Female, Male	6
Annotator Age Distribution	18–22 Years, 23–45 Years, 46+ Years	6
Ethnicity Distribution	Hispano or Latino, White or Caucasian, Multiracial, Black or African American, Middle Eastern, Asian, Other	6
Education Distribution	Less than high school diploma, High school degree or equivalent, Bachelor's degree, Master's degree, Doctorate, Other	6

4.2. Experimental Setup

We ran our experiments on Kaggle using a P100 GPU (16 GiB), with 29 GiB RAM and 57.6 GiB disk. We used PyTorch, Transformers, and Torchvision. Full setup details are shared in our GitHub repo.

4.3. Experimental Results

We evaluated our model on the **hard test** set provided in the EXIST 2025 shared task, which includes challenging memes exhibiting various degrees of subtle, implicit, or overt sexist content, in line with the annotation taxonomy (e.g., ideological inequality, stereotyping, objectification, etc.). The results obtained for Subtask 1, Subtask 2, and Subtask 3 are presented in Table 2.

The significant gap between training and validation accuracy across all subtasks indicates potential overfitting, warranting the application of further regularization strategies or architectural adjustments. To ensure robust performance during inference, we saved and reloaded the checkpoint corresponding to the highest validation accuracy in each subtask for final testing.

Table 2
Model Accuracy Metrics for All Subtasks with Annotator Ranking and Random Tie Breaking

Subtask	Metric	Annotator Ranking	Random Tie Breaking
Subtask 1	Training Accuracy	99.36%	99.61%
	Validation Accuracy	68.49%	65.74%
Subtask 2	Training Accuracy	99.32%	99.20%
	Validation Accuracy	68.52%	63.21%
Subtask 3	Training Accuracy	97.72%	99.21%
	Validation Accuracy	49.31%	45.48%

5. Conclusion and Future Scope

In this work, we proposed a robust multimodal classification model that effectively integrates textual and visual modalities using a combination of BERT, BLIP, HateBERT, and ViT architectures. By leveraging the strengths of both language and vision transformers, our approach demonstrates a comprehensive understanding of meme content, which is often ambiguous and context-dependent.

Across the three subtasks, our model consistently achieved high training accuracies, with subtask 1 and subtask 2 reaching 99.36% and 99.32% respectively, and subtask 3 achieving 97.72%. In terms of validation performance, subtask 2 yielded the highest accuracy at 68.52%, closely followed by subtask 1 at 68.49%, while subtask 3 lagged behind with 50.18%, likely due to increased task complexity or class imbalance.

These results highlight the capability of our multimodal framework to learn rich representations from both textual and visual features. However, the observed gap between training and validation accuracy suggests potential overfitting, indicating room for improvement through further regularization, data augmentation, or architectural enhancements. Overall, the promising validation results, especially in subtask 2, confirm the effectiveness of our integrated model in handling nuanced and multimodal data like internet memes.

5.1. Future Work

Regularization to Mitigate Overfitting. While the proposed model achieved excellent training accuracy—99.36% for Subtask 1 (Sexism Identification), 99.32% for Subtask 2 (Source Intention Classification), and 97.72% for Subtask 3 (Sexism Categorization)—a significant drop was observed during validation, with corresponding accuracies of 68.49%, 68.52%, and 49.31%. This performance gap indicates overfitting, where the model captures patterns specific to the training data but fails to generalize to unseen samples. Future work will focus on implementing advanced regularization techniques such as optimized dropout, fine-tuned weight decay, and strategic data augmentation. These measures aim to improve the model's generalization capabilities and robustness, especially in handling complex or subtle manifestations of sexism in meme content.

Dataset Expansion for Greater Robustness. Despite the diversity of the EXIST 2025 dataset, which consists of 4044 annotated memes, its size and cultural scope remain limited for developing a model with broad generalization. The relatively low validation accuracy in Subtask 3 (Sexism Categorization) particularly highlights this limitation. Future efforts will concentrate on expanding the dataset to include memes from a broader spectrum of languages, cultures, and social environments. This expansion is expected to capture a wider variety of sexist expressions and implicit biases, enabling the model to perform more reliably on real-world data from diverse online platforms.

Improving Multimodal Fusion Techniques. The current model uses a straightforward fusion strategy, combining image embeddings from ViT/BLIP and text embeddings from BERT/HateBERT via concatenation, followed by classification using a multi-layer perceptron (MLP). Although this approach yielded decent results—68.49% and 68.52% validation accuracy for Subtasks 1 and 2, respectively—it may not fully exploit the intricate relationships between visual and textual modalities, which are often essential in meme interpretation. Future research will explore more sophisticated fusion mechanisms, such as co-attention networks, cross-modal transformers, and gated fusion layers, to enable deeper interaction and contextual alignment between modalities.

Web Application for Practical Deployment. To broaden the real-world impact of this research, we plan to develop a web-based application that embeds the trained model into an accessible interface. This tool would allow users—including educators, moderators, and researchers—to upload memes and receive real-time predictions for sexism identification, source intention classification, and categorization. Given the promising validation accuracies achieved, the system demonstrates strong potential for practical use. A web application will enhance the visibility and usability of the model while supporting initiatives aimed at fostering safer and more inclusive digital environments.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] C. Yang, F. Zhu, J. Han, S. Hu, Scalable harmful meme detection using graph neural networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 46 (2024) 789–802.

[2] E. Asmawati, A. Saikhu, D. Siahaan, Sentiment analysis of text memes using supervised machine learning, Procedia Computer Science 190 (2021) 234–241.

[3] M. L. Jamil, S. Pais, J. Cordeiro, G. Dias, Detecting extreme emotions in social networks using bert, IEEE Access 11 (2023) 45678–45690.

[4] T. Zhou, Y. Niu, H. Lu, C. Peng, Y. Guo, H. Zhou, Vision transformers for image analysis: A comprehensive review, IEEE Transactions on Neural Networks and Learning Systems 34 (2023) 5123–5140.

[5] M. K. H. Tariq, Y. Zhu, M. Shrivastava, A. Thakur, Multi-modal meme classification with image-text joint embedding and transformer-based models, in: Proceedings of the Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations (CONSTRAINT), 2021, pp. 107–113.

[6] S. Ghosh, S. Srivastava, A. Malu, Y. Al-Onaizan, M. Shrivastava, Mami: A multimodal metaphor annotation dataset for internet memes, in: Findings of the Association for Computational Linguistics: EMNLP, 2022, pp. 3063–3074.

[7] M. Y. Khan, A. Munir, M. Sohail, A. Khan, C. Menon, F. Ahmad, Offensive memes detection using deep learning and ocr, in: International Conference on Information Technology and Systems (ICITS), 2022, pp. 497–506.

[8] S. Suryawanshi, B. R. Chakravarthi, R. Priyadharshini, J. McCrae, Multimodal meme emotion classification using deep learning, in: Proceedings of the Second Workshop on Multimodal Artificial Intelligence, 2021, pp. 21–29.

[9] Y. Qin, C. Liu, X. Liang, X. Li, L. Nie, Part-aware visual meme understanding via vision transformers, IEEE Transactions on Image Processing 32 (2023) 4107–4119.

[10] P. Badjatiya, S. Gupta, M. Gupta, V. Varma, Deep learning for hate speech detection in tweets, Proceedings of the 26th International Conference on World Wide Web Companion (2017) 759–760.

[11] M. Samory, T. Mitra, The unreliability of hate speech detection, in: Proceedings of the ACM on Human-Computer Interaction, volume 4, 2020, pp. 1–20.

[12] T. Caselli, V. Basile, J. Mitrović, M. Granitzer, HateBERT: Retraining BERT for abusive language detection in english, in: Proceedings of the Fifth Workshop on Online Abuse and Harms, Association for Computational Linguistics, Online, 2021, pp. 17–25. URL: https://aclanthology.org/2021.woah-1.3.

[13] X. Zhou, M. Zafar, M. A. Wani, I. Traore, I. Ghafir, Distributed comment embeddings for detecting toxic content on social media, Journal of Information Security and Applications 63 (2022) 103033.

[14] E. . Team, Exist 2025 dataset for sexism detection in social media, in: Proceedings of the Conference and Labs of the Evaluation Forum (CLEF), 2025, pp. 1–10.