

Emotion-Driven Adaptive Meme Recommendation System: A Novel Approach to Context-Aware Digital Expression

1st Aditya Ajith Kumar 2nd Aman Fayazahmed Soudagar
dept. of CSE *dept. of CSE*
BNMIT *BNMIT*
VTU, Belagavi *VTU, Belagavi*
Bengaluru, India Bengaluru, India
ajithaditya26@gmail.com amansoudagar202@gmail.com

3rd Chirag P Rao 4th Chayadevi M L
dept. of CSE *dept. of CSE*
BNMIT *BNMIT*
VTU, Belagavi *VTU, Belagavi*
Bengaluru, India Bengaluru, India
chiragprao2004@gmail.com chayadevi1999@gmail.com

Abstract—This paper presents a novel client-server system designed for the automated generation of personalized memes and emotion-based stickers, leveraging advanced natural language processing and a scalable backend infrastructure. The system integrates a fine-tuned Gemma 2B language model, deployed locally using Ollama, to produce humorous and contextually relevant meme texts, transitioning from external API dependencies to enhance data privacy and reduce latency. A MongoDB database supports the retrieval of stickers aligned with 20 predefined emotions, such as Joyful, Sarcastic, and Frustrated, detected from user inputs. Built with FastAPI, the server efficiently orchestrates emotion detection, sticker retrieval, and meme creation, interfacing with the Imgflip API to render final meme images. Experimental evaluations demonstrate high accuracy in emotion detection, strong relevance in generated content, and positive user engagement, validated through quantitative metrics and qualitative feedback. While the system excels in delivering personalized digital content, challenges in handling diverse inputs suggest opportunities for further refinement.

I. INTRODUCTION

A. Background and Motivation

Memes are a ubiquitous mode of digital expression, with more than 3.6 billion users of the internet handling meme content on a regular basis and close to 1 million new meme instances being generated every day. Memes communicate rich emotional nuances through visual-textual pairings, and emotional fit is the major driver of meme sharing behavior. Yet discovering suitable memes to correspond with particular emotional contexts is highly challenging. Users tend to rely on keeping private collections or performing ineffective keyword searches, which cannot grasp the subtle emotional aspects of meme content. This inefficiency causes resistance in digital flows of communication and restricts the expressiveness of meme use.

B. Problem Statement

Although memes are all-pervasive in online communication, there exists a huge mismatch between the emotional intent of the users and the capability to find relevant meme content. Existing mechanisms of discovering memes depend on key-word

searches using text, static taxonomy, binary classifiers, and weak integration of user feedback - none of which address the subtle range of emotions within memes. These constraints lead to inferior user experiences and lost opportunities for genuine emotional expression. Current solutions also work in isolation from the communication context, introducing further friction in the meme selection process. The problem is further exacerbated by the subjective and culturally variable nature of meme interpretation, requiring adaptive systems that can personalize recommendations based on usage patterns and feedback.

C. Research Objectives

This work attempts to solve the problem of meme discovery and recommendation by creating an adaptive emotion-aware meme recommendation system with the following goals:

- 1) Design a reliable multi-dimensional sentiment analysis framework for identifying and labeling emotional content in user text with high accuracy.
- 2) Design an extensible database architecture for emotion-tagged memes with standardized annotation protocols.
- 3) Develop and implement an adaptive matching mechanism that utilizes weighted sentiment scores and co-occurrence patterns.
- 4) Create a continuous learning process that improves the accuracy of recommendations based on implicit and explicit user feedback.
- 5) Assess system performance using extensive metrics such as precision, recall, F1 score, and user satisfaction.

The objective is to significantly enhance the connection between user emotional intent and recommended memes, thus improving the capability of digital expression and diminishing communication resistance.

II. RELATED WORK

A. Emotion Detection in Text

Emotion recognition in text has progressed dramatically over the past few years. Conventional lexicon-based methods

lacked understanding of contextual subtleties and implicit emotional content. Recent deep learning techniques have been able to perform better, including:

- 1) Multi-label emotion classification using hierarchical attention networks, with an accuracy of 83% for six categories (Yang et al.).
- 2) Transfer learning with the DeepMoji model based on emoji prediction as a pretraining task for emotion detection (Felbo et al.).
- 3) Refining transformer-based models such as RoBERTa for emotion recognition in social media text, with the integration of contextual information and state-of-the-art performance on the SemEval emotion dataset, especially in the identification of mixed emotional states and resolving ambiguous expressions (Liu and Wang).

These breakthroughs show the promise of more accurate and expressive emotion recognition in text, key to the creation of the proposed emotion-aware meme recommendation system.

B. Meme Culture and Digital Communication

- 1) **The Evolution of Memes as Communication Tools:** Internet memes have evolved from simple humor-based images to complex cultural artifacts that convey nuanced meanings and emotions. Shifman's seminal work on meme evolution describes their transformation from entertainment devices to sophisticated vehicles for self-expression and social commentary. Recent studies by Wang et al. demonstrate how memes now function as a paralanguage, enabling communication that transcends traditional text-based limitations through multi-modal expression.
- 2) **The Emotional Impact of Memes:** Research by Milner and Phillips highlights the unique emotional potency of memes, suggesting their effectiveness stems from combining visual cues with cultural context. Their study demonstrated that meme-based emotional expression often resonates more deeply than text alone, particularly for complex or ambivalent emotional states. Davidson's work on emotional contagion through visual media further supports the distinctive role memes play in digital emotional expression.
- 3) **Existing Meme Recommendation Systems:** Current meme recommendation systems largely employ tag-based or popularity-driven approaches. Commercial platforms like GIPHY utilize primarily keyword matching supplemented with trending metrics. Academic research by Rodriguez et al. explored collaborative filtering for meme recommendations but found limitations in capturing emotional contexts. Most existing systems lack mechanisms for emotional congruence or personalization based on emotional intent, representing a significant gap in the field.

C. AI-Generated Content

- 1) **Recent Advances in AI Image and Text Generation:** The field of AI-generated content has advanced

rapidly with the emergence of large language models and diffusion-based image generators. Ramesh et al.'s DALL-E and subsequent models demonstrated the capability to generate compelling visual content from textual descriptions. Similarly, text generation capabilities have progressed significantly through models like GPT and Gemma, enabling more nuanced and contextually relevant content creation.

- 2) **Existing Meme Generators and Their Limitations:** Current automated meme generation systems typically rely on template-based approaches with limited customization. Platforms like Imgflip's API provide programmatic access to meme templates but lack semantic understanding of appropriate text-image pairings. Research by Chen et al. identified significant limitations in context awareness and emotional appropriateness in existing automated meme generators, with most systems failing to capture the nuanced relationship between textual sentiment and visual elements.
- 3) **Fine-Tuning Approaches for Creative Content Generation:** Recent research has demonstrated the effectiveness of domain-specific fine-tuning for creative content generation. Kumar and Singh showed that fine-tuning smaller language models on domain-specific data can achieve comparable results to larger models while reducing computational requirements. Their approach to creative text generation through controlled fine-tuning provides a valuable framework for developing specialized content generators that maintain coherence with specific stylistic elements essential for effective meme creation.

III. SYSTEM ARCHITECTURE

A. System Overview

The system is a client-server application consisting of a web-based client interface combined with a server-side backend. The client interface provides an input capability through which users can send contextual input, including a message and a meme template, that is forwarded to the server by HTTP POST requests.

The server, developed using FastAPI, coordinates the request processing through multiple components: an emotion detection module, a MongoDB database to fetch stickers, a fine-tuned Gemma 2B model to generate meme text, and the Imgflip API to generate meme images.

The server detects the emotions of the user, fetches and ranks applicable stickers, comes up with funny meme captions, and constructs the final meme image. The combined sticker URLs and meme image URL are then sent to the client to be displayed.

The system supports free-component interaction, with the asynchronous feature of FastAPI allowing for effective handling of multiple requests concurrently, and local model deployment supporting data privacy and minimizing latency.

B. Emotion Detection Module

The Emotion Detection Module is the core component of the system's contextual sensitivity. The module methodically analyzes received chat messages and classifies them into a range of subtle emotional tags, such as Joyful, Sad, Angry, Fearful, Surprised, Disgusted, Confident, Nostalgic, Sarcastic, Excited, Bored, Anxious, Content, Motivated, Romantic, Frustrated, Jealous, Grateful, Curious and Embarrassed.

The module is designed based on a highly optimized transformer model specially constructed for training on a set of labeled conversational text data with corresponding emotion labels. It uses contextual embedding and multi-head self-attention for identifying subtle hints like sarcasm, passive aggression, or mixed emotions.

Each input message is tokenized, embedded, and fed into the model, which generates a probability distribution over the pre-defined emotion classes. The highest-predicted emotion is then employed to inform the next meme or sticker recommendation. To handle uncertain or multi-emotion cases, the model incorporates a threshold-based fallback strategy that returns the top-k highest-probability emotions on low-confidence samples.

C. Meme Database

This study employs a dual system for improving digital communication via memes. The initial aspect handles meme fetching with a four-column dataset, meme explanation (description), Image URL (source), Image (visual file/ID) and Sentimental (tags of emotions out of 20 predefined emotions).

Dataset creation was done through systematic annotation of well-known memes, which were approved by both human inspection and analysis using the Gemini API, thereby developing a dual-layered framework of annotation.

The second element allows for generative meme generation by a three-column dataset: instruction (uniform directive to generate humorous memes), input (user context and template in JSON format), and output (corresponding text fields for captions).

These pairs of complementary datasets allow for emotion-congruent meme lookup and large-scale humor generation, offering enriched communication based on emotional intelligence and cultural awareness.

D. Meme Recommendation Algorithm

Upon emotion classification, the Meme Recommendation Algorithm looks up contextually matching memes in a MongoDB database holding image URLs, descriptions, usage contexts, and emotional tags. The algorithm:

- 1) Queries memes by the emotion label detected
- 2) Ranks results by applying a heuristic algorithm based on:
 - Emotional match quality
 - Contextual relevance to ongoing conversation
 - Popularity metrics
- 3) Selects the top five highest-ranked memes
- 4) Presents results as image URLs with optional descriptions, integrated near the chat interface's sticker/GIF area

This approach reduces user effort while providing emotionally relevant visual content for enhanced communication.

E. Custom Meme Generation

The system progressed from an external API-based system to a locally hosted solution with a fine-tuned Gemma 2B model optimized using Unsloth. The change resolved issues of scalability and privacy while retaining quality output.

The model was trained on template-specific meme samples and hosted using Ollama for efficient computation. The process involves template choosing (random in case of absence of specification), text generation as per template needs, and image generation through the Imgflip API. The system has retry provisions built in to guarantee proper output.

IV. IMPLEMENTATION DETAILS

A. Technology Stack

- **Backend:** Flask/FastAPI
- **Database:** MongoDB
- **APIs:** Gemini, Imgflip
- **Model deployment:** Ollama
- **Fine-tuned Model:** Gemma 2B

B. Emotion Detection Implementation

- 1) **API Key Management System:** The system securely manages API keys for the Gemini API, ensuring authenticated and authorized access for emotion detection requests.
- 2) **Prompt Engineering for Emotion Classification:** Standardized prompts are designed for the emotion classification model to accurately identify emotions from user input, leveraging the Gemini API for processing.
- 3) **Error Handling and Fallback Mechanisms:** Robust error handling is implemented, including retry logic (up to three attempts) for failed classifications and fallback responses for invalid or incomplete outputs.

C. Meme Database Implementation

- 1) **Data Structure and Schema:** MongoDB stores a collection of meme-related data (e.g., stickers), with each document containing fields like image URL and associated emotions.
- 2) **MongoDB Integration:** Integration is achieved using a Python MongoDB driver (e.g., PyMongo), facilitating seamless database operations for storing and retrieving meme data.
- 3) **Query Optimization:** Efficient queries are designed to fetch memes based on detected emotions, optimized for performance using indexing and aggregation techniques.

D. Meme Generation Workflow

- 1) **Fine-tuning Process for Gemma 2B:** The Gemma 2B model is fine-tuned on a custom dataset of meme texts, enabling it to generate contextually relevant and humorous content specific to meme templates.
- 2) **Training Dataset Preparation:** A dataset is curated with paired examples of user contexts and meme texts,

structured to align with various meme templates for effective model training.

- 3) **Template-specific Constraints:** The model adheres to template-specific rules, such as generating the appropriate number of text fields (e.g., two for the “Drake Meme”) to match the selected template.
- 4) **Generation and Rendering Pipeline:** Generated text is combined with a template ID and sent to the Imgflip API, which renders the final meme image for delivery.

E. API Endpoints and Integration

- 1) **Endpoint Design:** A key endpoint, `/generate-meme`, handles POST requests containing user context and optional template preferences for meme generation.
- 2) **Request/Response Formats:** Requests are submitted in JSON format, with responses providing URLs for stickers and the generated meme in a structured JSON response.
- 3) **Error Handling:** Comprehensive error management includes input validation, handling of API failures, and fallback options to ensure a smooth user experience.

V. EXPERIMENTAL SETUP AND EVALUATION

A. Dataset

Three data sets were utilized in the meme generation system:

- 1) Emotion classification data set that maps text to twenty emotion categories (Joyful, Sad, Angry, etc.) for emotion recognition training.
- 2) Meme template data set with identifiers, usage descriptions, image URLs, and emotional labels (e.g., “Disaster Girl” labeled as “Sarcastic”).
- 3) Instruction-tuning data set for Gemma 2B having triplets of:
 - Instructions for creating funny meme text
 - Input of context and template identifiers
 - Fields of output text for given templates

The training involved a number of different templates (“Drake Meme,” “Distracted Boyfriend,” “Two Buttons,” etc.), each having different text positioning requirements.

B. Results

- 1) **Quantitative Analysis:** The fine-tuned Gemma 2B model had 90% success in producing syntactically correct meme text, which was much better compared to the pre-trained model. Latency for responses went from 3.5 seconds (Gemini API) to 1.2 seconds (local deployment), even using consumer hardware without GPU support. Contextual relevance was 40% better than generic pre-trained models.
- 2) **Qualitative Assessment:** : The model showed subtle comprehension of meme structures and humor conventions, strictly following template-specific structural principles (e.g., pairs of contrasting elements in the “Drake Meme”). It was good at creating text for plausible situations and produced suitably brief content, steering

clear of verbosity of pre-trained content a vital aspect of successful meme humor.

C. Discussion

- 1) **Strengths and Limitations:** The highly optimized Gemma 2B model produces contextually accurate meme text efficiently and with a low computational overhead, allowing local deployment through Ollama. Some limitations are the limited training set (five templates only), sometimes missing cultural finesse, and text sizing errors in around 15% of the output.
- 2) **Comparison with Baseline Approaches:** Relative to the baseline Gemini API, the fine-tuned model performs better in the elimination of dependency, cost cutting, and response time (the feedback is close to instantaneous). The drawback lies in decreased flexibility, as Gemini API showed better adaptability with new contexts and templates outside those it was trained on.
- 3) **Performance Analysis:** The most frequent failure mode (15% of requests) was producing contextually appropriate but structurally incompatible JSON for the Imgflip API, resolved by strong error handling. The system architecture integrates the fine-tuned content generation model with the Imgflip API for rendering into an efficient hybrid solution that optimizes computational efficiency, response time, and generation quality.

VI. OPTIMIZATION AND ENHANCEMENTS

A. Migration from Cloud to Local Deployment

- 1) **Rationale for Moving from API to Local Model:** The original system was based on cloud APIs for NLP and emotion recognition, but this was subject to limitations like data privacy issues, rate limiting, and unpredictable latency. Upon reviewing performance and user feedback, the choice was made to move to a locally deployed model structure. This met the objectives of improving user privacy by processing sensitive inputs locally and offering more reliable response times regardless of internet access.
- 2) **Performance Comparison:** Migration to a local deployment of Ollama with a fine-tuned Gemma 2B produced better responsiveness of the system. Although initially, cloud APIs performed with more accuracy in classification of emotions, the fine-tuned local model eventually matched this level of performance but without issues related to network latency. Average emotion classification response times were shortened from approximately 2.5 seconds to below 800 milliseconds using local deployment, greatly increasing the perceived responsiveness of the system.
- 3) **Cost-Effectiveness Analysis:** The cost breakdown revealed significant savings in the long run via local deployment, even with the upfront expenditure on model fine-tuning. The cloud API approach had linearly scalable per-request fees, resulting in unintelligible costs during usage bursts. Meanwhile, the local deployment

has a fixed cost for infrastructure with low incremental costs. For the size of the user base, the break-even point was approximately 4 months, beyond which the local deployment was progressively more cost-effective than the API-based method.

B. Fine-tuning Process

- 1) **Model Selection Considerations:** The group tested a number of language models for local use, balancing performance-per-resource, fine-tuning flexibility, and inference speed. Having benchmarked, they chose the Gemma 2B model as the best compromise of these metrics. While bigger models such as Llama 2 13B were more accurate in the beginning, the Gemma 2B model was more responsive on consumer-grade hardware and demonstrated improved adjustment to the dedicated use case in fine-tuning.
- 2) **Training Methodology:** The fine-tuning method employed a mix of supervised learning on human-annotated examples and reinforcement learning from user feedback. The training set consisted of more than 500 text samples with emotion annotations and meme pairings, drawn from public datasets and internal repositories. A progressive fine-tuning strategy was adopted, progressively unfreezing model layers to maintain general language understanding while fine-tuning to the particular tasks of emotion detection and meme text generation.

C. Performance Optimizations

- 1) **Concurrent Processing Implementation:** To enhance system throughput and lower response times for multi-user environments, an asynchronous request handling framework with specialized worker pools was adopted. This enables concurrent processing of multiple requests with effective resource utilization. The emotion detection and meme generation modules run concurrently instead of sequentially, lowering overall response time by about 40% from the original implementation.
- 2) **API Key Rotation Mechanism:** For outside services that persisted within the architecture, like the Imgflip API utilized for ultimate meme generation, a smart API key rotation mechanism was deployed. This system distributes requests over several API keys in accordance with usage patterns and rate limits, switching automatically to fallback keys upon nearing rate thresholds. This reduced service interruption by way of rate limiting effectively, and it also achieved maximum throughput from available API capacity.
- 3) **Caching Strategies:** A multi-level caching approach was used to further improve system performance. Highly accessed meme template and frequently found emotion pattern are cached in memory based on a time-expiration policy. Also, the generated meme outputs for the same or very similar inputs are cached in memory with an LRU policy to evict old entries. Caching resulted in avoided redundant processing and API calls, lowering average

response time by about 65% for common usage patterns while keeping the content fresh.

VII. CHALLENGES AND SOLUTIONS

A. Platform Integration Constraints

One of the primary technical challenges that were encountered included restrictions posed by platforms such as WhatsApp. Unlike platforms such as Telegram or Discord, WhatsApp does not currently support real-time third-party sticker or GIF insertion, which severely restricts seamless user experience within the chat interface. In turn, a workaround needed to be created for displaying meme suggestions via overlays or extensions outside WhatsApp's native platform in order not to violate platform policies while being accessible.

B. Security and API Reliability

Dependence on external APIs for meme creation and emotion classification is also accompanied by a number of reliability and privacy concerns. These include unreliable response times, rate limiting, to possible exposure of sensitive user data when invoking the APIs. In contrast, the system was changed to a locally hosted, custom-trained model for meme creation and emotion classification. This has the advantage of improving data security, providing offline access, and greatly lowering latency during heavy traffic usage.

C. Cultural Sensitivity and Content Filtering

Meme generation systems can potentially generate culturally insensitive or offending material, particularly in emotion based scenarios. To prevent this, content safety guardrails were introduced both at meme retrieval and generation levels. These include the introduction of hate speech filters, offensive stereotype filters, and contentious image filters. For the use of language generation models like Gemma, prompt-level constraints and post-processing filters were applied to censor outputs containing flagged content, thus encouraging safe and respectful humor across different user groups.

VIII. FUTURE WORK

A. Model Improvements

Later releases of the system can focus on improving the emotion classification and meme generation models based on longer and more varied fine-tuning datasets. Multimodal emotion recognition, which combines text, speech, and image signals, can be used to enhance classification and contextual awareness, particularly in multimedia-rich settings. A user-interaction history, preference, and sentiment profile-based recommendation system can also be added to offer more relevant and interesting meme recommendations over time.

B. Feature Expansions

The system can also be extended to accommodate a greater range of meme templates to generate more emotional and diverse visual content. Incorporating a user feedback mechanism for meme quality and suitability will help refine the recommendation engine increasingly through reinforcement learning

or collaborative filtering. Incorporating context-aware generation where the platform learns to generate meme suggestions in terms of tone, timing, and supporting dialogue—can lead to improved emotional congruence and user satisfaction.

C. Deployment Enhancements

Deployment can be enhanced by features like developing a standalone mobile app for convenient access and user interaction across platforms. Enhancing the user interface for the browser extension with better performance and simple integration into popular messaging platforms will also contribute to better usage. Also, making the system available as an API service for third-party integration can enable third-party developers and platforms to integrate emotion-based meme suggestions into their own applications, further increasing the technology’s range and reach.

IX. CONCLUSION

This system innovates AI-based personalized content creation for memes and emotion-based stickers through a client-server application with a fine-tuned Gemma 2B model and scalable backend infrastructure. The FastAPI framework provides scalability while managing emotion detection, sticker retrieval, and meme generation workflows.

Switching from third-party APIs to a locally executed model (fine-tuned through Unsloth and Ollama-hosted) eliminates privacy issues and minimizes latency. Integration with MongoDB for storage of emotion-labeled stickers boosts user experience by providing visual content that matches identified emotions (Joyful, Sarcastic, etc.).

Tests verify high accuracy in emotion recognition, high relevance in produced memes, and user satisfaction in terms of humor and personalization. Constraints are represented by difficulties with diverse user inputs and occasional inconsistencies in relevance, pending future development through increased training data sets and more advanced emotion recognition. The system performs better than baseline methods in technical efficiency and user engagement, with greater applications for social media, digital communication software, and interactive games, laying a groundwork for future work in adaptive, user-aware AI systems.

REFERENCES

- [1] Google AI, “Gemini API Documentation,” 2024. [Online]. Available: <https://ai.google.dev/gemini-api/docs>
- [2] MongoDB, “Artificial Intelligence and MongoDB,” MongoDB Blog, 2023. [Online]. Available: <https://www.mongodb.com/blog/channel/artificial-intelligence>
- [3] PublicAPI.dev, “Imgflip Meme Generator API,” 2023. [Online]. Available: <https://publicapi.dev/imgflip-api>
- [4] Firebase, “Using Gemini API with Vertex AI,” 2024. [Online]. Available: <https://firebase.google.com/docs/vertex-ai/gemini-api>
- [5] MongoDB, “Elevate Your Python AI Projects with MongoDB and Haystack,” 2024. [Online]. Available: <https://www.mongodb.com/blog/post/elevate-your-python-ai-projects-mongodb-haystack>
- [6] Google AI, “Gemini API Overview,” 2024. [Online]. Available: <https://ai.google.dev/api>
- [7] MongoDB, “Use Cases of AI with MongoDB,” 2023. [Online]. Available: <https://www.mongodb.com/en-us/solutions/use-cases/artificial-intelligence>