MemeMatch: Context-Aware Multimodal Meme Retrieval and Recommendation via Structured Semantic Understanding

1st Tri An Le
AIT-Budapest
Budapest, Hungary
triandole@gmail.com

***Abstract*—This project presents an intelligent meme recommendation system that leverages state-of-the-art natural language processing and computer vision techniques to suggest relevant memes based on user queries or uploaded images. By integrating OCR (Optical Character Recognition), image captioning models, and semantic similarity analysis, the system can understand both textual and visual content, enabling context-aware meme recommendations. The backend is implemented using FastAPI, supporting both text-based and image-based inputs, and utilizes precomputed embeddings and sentiment analysis to enhance the relevance, accuracy, and diversity of recommendations.**

I. Introduction

Memes have emerged as a dynamic and influential form of communication in the digital age, capable of conveying complex ideas, emotions, and cultural commentary through a concise blend of text and imagery. Their virality and adaptability have made them a dominant mode of expression on social media platforms, contributing to public discourse and online identity formation. However, with the vast and rapidly growing volume of meme content, users often struggle to discover memes that are contextually appropriate, emotionally resonant, or relevant to specific scenarios.

To address this challenge, we present an intelligent meme recommendation system that leverages state-of-the-art natural language processing (NLP) and computer vision techniques. The system is designed to understand both the textual and visual components of memes, enabling personalized and context-aware recommendations based on user input in the form of text queries or uploaded images.

To frame the scope of our system more precisely, we introduce the concepts of *local context* and *global context*, which play a central role in our model design. The **local context** of a meme refers to the user-generated textual content—such as the overlaid text on the meme image and the meme's title. This content captures the immediate, situation-specific intent behind a meme's usage. In contrast, the **global context** refers to the underlying meme template—the structural and visual foundation that typically carries an established, shared cultural meaning or format.

By incorporating both local and global context into our system, we aim to more effectively capture the nuanced intent and communicative power of memes, ultimately enhancing the quality and relevance of the recommendations provided to users.

II. Previous Work

Research in meme analysis and recommendation has grown significantly in recent years, largely driven by the increasing availability of multimodal learning techniques. Early approaches to meme recommendation often relied on simple keyword matching, rule-based filters, or metadata tags, which struggled to capture the nuanced combination of text and image typically found in memes.

To address these limitations, recent work has explored deep learning methods that process both textual and visual information. Kiela et al. introduced the Hateful Memes Challenge, which framed meme understanding as a multimodal task and presented a benchmark dataset combining images and text for hate speech detection [5]. This highlighted the importance of fusing linguistic and visual features to fully grasp the meaning of a meme.

Building on this, Suryawanshi et al. proposed a multimodal sentiment analysis system for memes that uses convolutional neural networks (CNNs) for image feature extraction and recurrent neural networks (RNNs) for text processing [10]. Their results showed that sentiment features play a crucial role in understanding the emotional tone of meme content.

One of the most influential models in the field of vision-language learning is CLIP (Contrastive Language–Image Pre-training), developed by OpenAI [8]. CLIP learns joint embeddings of images and natural language using contrastive learning across large-scale internet data, allowing it to perform zero-shot image classification and cross-modal retrieval. CLIP's ability to bridge the gap between visual and textual modalities makes it a foundational model for applications like meme recommendation.

Additionally, Chandrasekaran et al. explored the role of humor, emotion, and sentiment in meme generation, emphasizing that a successful meme system must understand more than literal content—it must also grasp context, tone, and cultural reference [3].

Our work builds upon these foundations by integrating several key components: OCR to extract overlaid text, image captioning for semantic scene understanding, sentence-

transformer-based embeddings for textual similarity, and CLIP for multimodal embedding alignment. By combining these tools, our system aims to offer context-aware, sentiment-aligned, and visually-relevant meme recommendations that go beyond keyword matching or isolated feature extraction.

III. Dataset

Our meme recommendation system is built on a large-scale, custom-curated dataset comprising approximately **301,000 memes** and **2,100 meme templates**. The meme data were collected from two primary sources: **Reddit** and **Imgflip**, covering a diverse array of meme topics and spanning the years **2018 to 2024**. The dataset also includes rich metadata associated with each meme, such as:

- **ID**
- **Score** (upvotes)
- **Title**
- **URL**
- **Reddit post link**
- **Creation timestamp**

We describe below the data collection pipelines for each source in detail.

A. Reddit Dataset

Approximately **150,000 memes** were gathered from Reddit using the **Python Reddit API Wrapper (PRAW)** in conjunction with the **Pushshift API**, a third-party service that indexes Reddit data and supports more granular querying (e.g., by time range).

Initially, we used the **PSAW (Pushshift.io API Wrapper)** package to query the Pushshift API. However, due to performance and scalability limitations, we transitioned to **PMAW (Pushshift Multithread API Wrapper)**, which leverages Python's multiprocessing capabilities to significantly speed up data collection.

Since the Reddit API enforces strict request limits (typically 1000 requests per day), using Pushshift via PSAW/PMAW enabled us to bypass these constraints and extract large volumes of historical Reddit posts. Each record included image URLs and metadata. The resulting dataset consists of:

- A folder of downloaded meme images
- A folder containing corresponding metadata files, organized by subreddit, collectively stored in a directory named `subreddits`

B. Imgflip Dataset

The remaining **151,000 memes** and **2,100 meme templates** were scraped from **Imgflip**, a popular online meme generator, using **Selenium**, a web automation framework.

The raw data collected from Imgflip contained many duplicate entries. To clean the dataset:

- We applied **image hashing techniques** (e.g., perceptual hash) to identify and remove duplicate memes and templates.
- Only unique images were retained to ensure high-quality and diverse meme content.

The final Imgflip dataset is organized into two main folders:

1) A folder containing multiple subfolders, each named after a meme template. These subfolders include memes that were generated using the corresponding template.
2) A folder containing the **meme templates** themselves, used to infer relationships between memes and their structural origins.

IV. Methodology

A. Compute Embeddings and Sentiment Scores

The first stage of our system involves preprocessing raw image-with-text memes and preparing structured representations such as semantic embeddings, sentiment scores, and usage labels. These representations form the foundation for accurate meme recommendations in the next stage.

Figure 1 illustrates an overview of the preprocessing pipeline.

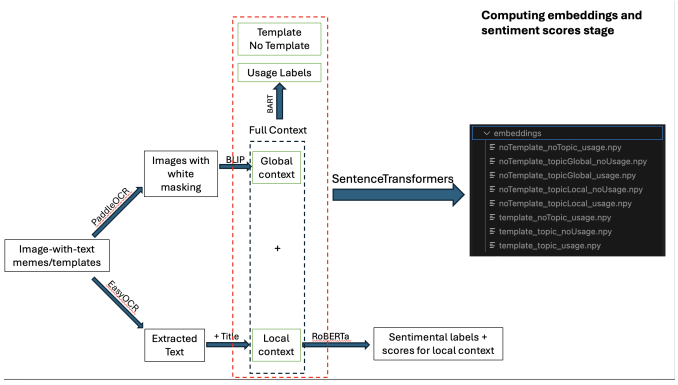

Fig. 1. Overview of our preprocessing and preparation pipeline.

1) Local Context Extraction: Extracting user-generated text overlays from meme images is central to analyzing memes' local context, which typically contains the meme's main message or punchline. Optical Character Recognition (OCR) tools like EasyOCR and Tesseract have been extensively utilized for this task [1], [9], but meme-specific characteristics such as irregular fonts, visual noise, and watermarks pose significant challenges.

To accurately extract and isolate user-generated text, we implemented the following steps:

1) We first employed EasyOCR [1] to extract raw text from meme images.
2) We similarly applied EasyOCR to extract default embedded text from the 1,145 most common meme templates from the IMGFlip dataset.
3) To distinguish between local (user-added) text and global (template-embedded) text, we developed a robust data cleaning pipeline, which involved:
   - Removing non-informative tokens such as watermarks, URLs, special characters, and repeated artifacts.

- Filtering out global template-based text patterns by cross-referencing with text extracted from the IMGFlip dataset.

The cleaned OCR-extracted text was then combined with meme titles provided by Reddit metadata to constitute the comprehensive **local context**. This combination ensures that our analyses capture both explicit user-edited messages and descriptive contextual information supplied by meme creators.

Figure 2 demonstrates a raw OCR output example, while Figure 3 illustrates the cleaned and finalized local context, ready for analysis.

Fig. 2. Raw OCR output example

Fig. 3. Cleaned and finalized Local Context

2) Global Context Extraction: The global context reflects the underlying visual structure and thematic essence of meme templates, independent of user-added textual modifications. Recognizing the significance of visual semantics, we adopted advanced image captioning methods based on transformer architectures, specifically leveraging BLIP [6]. This model combines vision-language pretraining with robust generative capabilities, enabling rich and semantically coherent image descriptions.

To isolate the global visual template context, the following steps were executed:

1) User-generated text regions were identified and masked using PaddleOCR [4], ensuring the subsequent visual analysis only captured the underlying meme template.
2) The masked images were input into the BLIP image captioning model to produce descriptive captions representing the global visual semantics.

We intentionally used two distinct OCR systems to support the different goals of local and global context processing. For local context extraction, we selected EasyOCR due to its high efficiency and low computational cost, which made it well-suited for large-scale meme processing. Despite being lightweight, EasyOCR delivered sufficient accuracy for our task, reliably extracting diverse fonts and noisy overlaid text commonly found in memes. In contrast, for global context recovery, the objective was not to extract text content but to *precisely localize* user-generated text regions for masking. To achieve this, we employed PaddleOCR, which offers superior bounding box precision and more accurate region detection—especially for small or irregularly positioned text. This allowed us to generate cleaner masks and ensure that subsequent image captioning by BLIP reflected only the original, unaltered visual template.

3) Sentiment Modeling: Memes often carry complex emotional undertones. To capture these, we applied two transformer-based sentiment pipelines:

- `cardiffnlp/twitter-roberta-base-emotion-multilabel-latest` for multi-label emotion classification [2]
- `cardiffnlp/twitter-roberta-base-sentiment-latest` for sentiment polarity classification [7]

Both models are RoBERTa-based and trained on Twitter data, making them well-suited to meme content, which often mirrors the informal, expressive tone of social media language. Input consisted of the combined OCR-extracted text and Reddit title, truncated to the 512-token limit.

The emotion classifier returns scores across 11 emotions (e.g., *anger*, *joy*, *trust*), while the sentiment classifier outputs polarity labels (*positive*, *neutral*, *negative*). The dual-model approach provides a nuanced emotional profile. The final output for each meme is a vector of 14 sentiment scores stored in a structured CSV. Figure 4 shows some examples for the sentiment analysis.

Text / Emotion	"3DS owners: OH NO! We'll have to pirate games now!" Wii owners:"	"Fat free milk water. No difference whatsoever"
Anger	0.479241	0.593545
Anticipation	0.199450	0.054121
Disgust	0.647908	0.673541
Fear	0.066868	0.010768
Joy	0.032594	0.033441
Love	0.002387	0.003806
Optimism	0.016223	0.021999
Pessimism	0.075464	0.022189
Sadness	0.217513	0.673541
Surprise	0.060239	0.022189
Trust	0.005590	0.004430
Negative	0.768447	0.079837
Neutral	0.215292	0.720003
Positive	0.016261	0.200160

Fig. 4. Sentiment and emotion scores of local context

4) Usage Label Prediction via Zero-Shot Classification: To model meme functionality and intended usage, we curated a set of 28 custom usage labels that span diverse communicative functions (e.g., satire, motivation, complaint, political commentary).

We merged the local and global context into a single string and used the `facebook/bart-large-mnli` model for zero-shot classification. The full context served as input, and the predefined usage labels as candidate classes. Figure 5 illustrates an example meme with predicted usage labels.

Fig. 5. Example of predicted usage labels.

5) Case-Based Embedding Generation: Finally, we generated high-dimensional semantic embeddings to handle different types of user queries. These embeddings capture the meme's meaning and usage in context.

We used the `all-mpnet-base-v2` model from SentenceTransformers, chosen for its strong performance on sentence similarity tasks. Each meme or template was encoded into a vector space where semantic similarity could be computed via cosine distance.

To accommodate different query types (e.g., with/without template, usage intent, or topic focus), we defined eight embedding cases:

- **noTemplate_noTopic_usage:** Used when the user specifies usage intent but no topic or template. E.g., "Give me a meme for humor."
- **noTemplate_topicGlobal_noUsage:** Topic is inferred from the template's visual content. No usage specified. E.g., "Give me Spongebob memes."
- **noTemplate_topicLocal_noUsage:** Topic is inferred from OCR text and title only. No usage specified. E.g., "Give me memes about college students."
- **noTemplate_topicGlobal_usage:** Topic inferred from image and usage specified. E.g., "Give me Spongebob memes to make fun of my friends."
- **noTemplate_topicLocal_usage:** Topic inferred from user text and usage is specified. E.g., "Give me memes to make fun of dating app."
- **template_topic_noUsage:** Template is specified along with a topic, but usage is not. E.g., "Give me some Minion meme templates."
- **template_topic_usage:** All three inputs are specified—template, topic, and usage. E.g., "Give me some Minion meme templates to make fun of college exams."

Each case is encoded independently, enabling flexible recommendation logic depending on the user's input structure. The use of these emdeddings will be discussed further in the next section.

B. Feature 1: Natural Language Query

This feature enables users to request memes or meme templates using free-form natural language. The system processes the query and returns the top-N most relevant recommendations based on semantic similarity.

Figure 6 and Figure 7 illustrate the two-stage query processing pipeline.

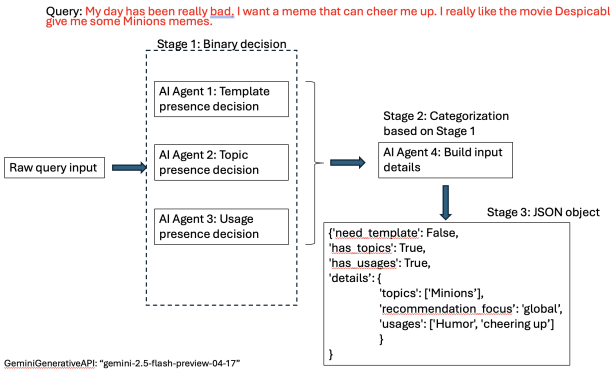

Fig. 6. Stage 1: Natural Language Query Parsing and Categorization

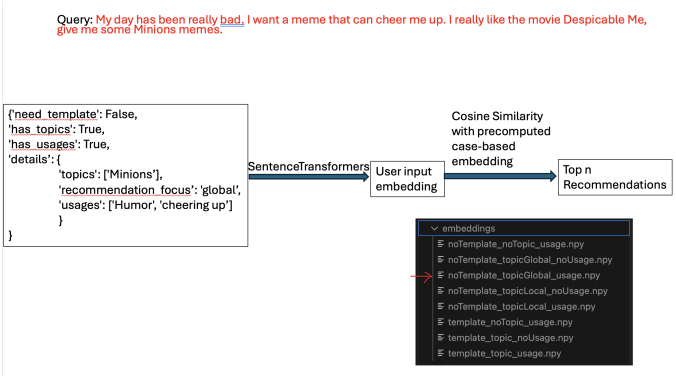

Fig. 7. Stage 2: Semantic Search and Ranking

1) Stage 1: Query Parsing and Categorization: In the first stage, the user's raw query is processed by a series of four AI agents powered by GeminiGenerativeAPI (`gemini-2.5-flash-preview-04-17`). Each agent has a dedicated role:

- **Agent 1:** Determines whether the user is requesting memes or meme templates.
- **Agent 2:** Detects whether specific topics or keywords are mentioned.
- **Agent 3:** Checks for any intended usage or purpose behind the requested memes.

Each of these three agents outputs a Boolean value, indicating the presence or absence of the respective category within the query.

Using this information, **Agent 4** synthesizes the results from the previous agents, extracting all relevant details from the query corresponding to the categories identified as present. The final output is a structured JSON object, which includes the fields `'need_template'` (AI Agent 1), `'has_topics'`(AI Agent 2), `'has_usages'`(AI Agent 3), `'details'`(AI Agent 4) shown in Stage 3 of Figure 6.

This structured output is then passed to the semantic search and ranking stage.

2) Stage 2: Semantic Search and Ranking: In the second stage, the system semantically matches the user's query with precomputed meme or template embeddings.

If either topic or usage information is present, they are concatenated and encoded using `SentenceTransformers("all-mpnet-base-v2")` to form a high-dimensional user input embedding.

The system selects the appropriate embedding index (e.g., `noTemplate_topicGlobal_usage`) and computes cosine similarity between the user input embedding and all items in that index. The top-N most similar memes or templates are then returned as recommendations.

This two-stage approach ensures that meme suggestions are both content-aware and aligned with the user's intent.

3) Fallback Condition: In cases where the user input is vague or lacks explicit topics or usage intents—such as "Give me a funny meme" or "Show me a meme template"—the system activates a fallback mechanism (Figure 8).

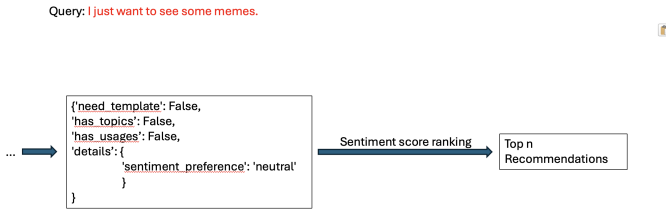

Fig. 8. Fallback Condition for Vague Queries

In this scenario, both `has_topics` and `has_usages` are `False`, meaning there is insufficient semantic information to generate an embedding using the sentence transformer model.

Instead, the system falls back on precomputed sentiment scores for all memes or templates. The user's query is analyzed for affective cues (e.g., "funny" implies high `joy` sentiment). Based on this inferred emotion, the system ranks all candidates by their corresponding sentiment scores and returns the top-N recommendations.

For instance, if the query is "Give me some funny memes," the model selects and returns memes with the highest `joy` scores.

C. Feature 2: Find Similar Memes Based on Local Context (Text-Based) and Global Context (Image-Based)

This feature allows users to upload a meme image, and the system returns the top-N most similar memes based either on the local context (text extracted from the image) or the global context (overall visual semantics of the image). Figure 9 illustrates the pipeline for global context (image-based) similarity, while Figure 10 shows the pipeline for local context (text-based) similarity.

Fig. 9. Finding Similar Memes Based on Global Context (Image-Based)

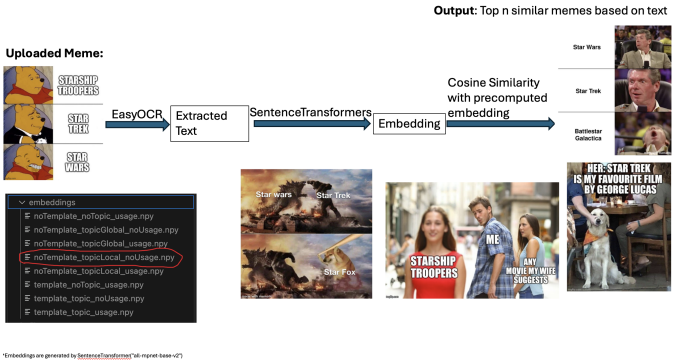

Fig. 10. Finding Similar Memes Based on Local Context (Text-Based)

1) Global Context (Image-Based Similarity): To capture the overall semantic content of the uploaded meme, we first use PaddleOCR to detect and white-mask all text regions, ensuring that the model focuses solely on visual elements. The masked

image is then passed to a pre-trained BLIP (Bootstrapped Language-Image Pretraining) model to generate a descriptive image caption.

This caption is encoded into a high-dimensional embedding using `SentenceTransformers("all-mpnet-base-v2")`. The system uses the **noTemplate_topicGlobal_noUsage** index and computes cosine similarity between the caption embedding and all embeddings in the database. The top-N most semantically similar memes or templates are returned.

2) Local Context (Text-Based Similarity): To analyze textual content directly from the meme, we apply EasyOCR to extract text from the uploaded image. The extracted text is then encoded using `SentenceTransformers("all-mpnet-base-v2")` to form a high-dimensional embedding.

The system then uses the **noTemplate_topicLocal_noUsage** index and calculates cosine similarity between the text embedding and all precomputed embeddings. The top-N closest matches are returned, allowing users to find memes with similar textual context.

V. Evaluation

This section outlines our evaluation methodology for assessing the performance of the text-to-image retrieval system across two tasks: meme retrieval and meme template retrieval. We begin by detailing our semi-automated approach for generating a relevance-labeled test dataset, including query sampling and annotation guidelines. We also describe the evaluation setup, including how relevance thresholds are defined and used. Finally, we introduce the set of evaluation metrics used to quantify retrieval performance under varying definitions of relevance.

A. Test Dataset Generation and Setup

To evaluate our retrieval system, we employed a semi-automated pipeline for generating ground-truth relevance labels, as illustrated in Figure 11.

We sampled approximately 200 diverse text queries for each task—meme retrieval and meme template retrieval—using Gemini Generative AI. These queries varied in length, phrasing, and topic to reflect a broad range of user intents. Each query was then used as input to the retrieval system.

For each meme query, the top 20 retrieved meme images were collected. For each meme template query, the top 10 retrieved templates were recorded. This produced a set of query–candidate pairs: 20 pairs per meme query and 10 pairs per template query.

Each pair was evaluated by Gemini Generative AI, which assigned a **relevance score** based on semantic and contextual alignment between the query and the retrieved item:

- **2**: Highly relevant — a direct semantic and contextual match.
- **1**: Relevant — conceptually related but less precise.
- **0**: Irrelevant — no meaningful connection to the query.

All labeled pairs were stored in the format: `<query>, <relevance_score>, <image_path>`, creating a standardized dataset for quantitative evaluation.

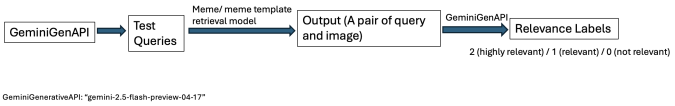

Fig. 11. Semi-automated pipeline for generating relevance-labeled test data for meme and template retrieval.

By standardizing labeling and query diversity, this evaluation setup ensures fair, consistent, and scalable benchmarking across different query types and relevance criteria.

B. Evaluation Metrics

We employed several standard information retrieval metrics to evaluate the performance of our system at different cutoff points (k=5, 10, 20):

- **Normalized Discounted Cumulative Gain (nDCG@k):** Measures the ranking quality of the retrieved items, giving higher scores to more relevant items ranked higher. It is normalized by the ideal DCG at that cutoff.
- **Precision@k (P@k):** The proportion of retrieved items in the top k that are relevant.
- **Recall@k (R@k):** The proportion of all relevant items in the collection that are retrieved in the top k.
- **Mean Average Precision (MAP):** The mean of the average precision scores for each query. Average precision rewards ranking relevant items higher.
- **Mean Reciprocal Rank (MRR):** The average of the reciprocal ranks of the first relevant item for each query. This metric is particularly useful for tasks where finding the first correct answer quickly is important.

Additionally, we present Precision-Recall curves to visualize the trade-off between precision and recall across different thresholds, and analyze the Mean Reciprocal Rank (MRR) as a function of query length.

VI. Results

This section presents the quantitative evaluation of our meme and meme template retrieval system using standard information retrieval metrics: nDCG, Precision, Recall, Mean Average Precision (MAP), and Mean Reciprocal Rank (MRR), computed at various cut-off ranks (e.g., @5, @10, @20). We evaluate performance under two relevance thresholds:

- **Threshold = 1 (Relevant):** A retrieved item is considered relevant if it shares a general conceptual relation to the query, allowing for broader matches with less strict precision.
- **Threshold = 2 (Highly Relevant):** A stricter criterion where the retrieved item must exhibit a direct semantic and contextual match to the query, reflecting high specificity.

We first present meme retrieval results, followed by meme template retrieval, and conclude with analyses based on

precision-recall curves and the impact of query length on performance.

A. Meme Retrieval Evaluation Results

Table I summarizes the meme retrieval performance under two relevance thresholds.

TABLE I
EVALUATION RESULTS FOR MEME RETRIEVAL

Metric	Threshold = 1 (Relevant)			Threshold = 2 (Highly Relevant)		
	@5	@10	@20	@5	@10	@20
nDCG	0.6383	0.6837	0.8411	0.6383	0.6837	0.8411
Precision	0.8465	0.8446	0.8255	0.4901	0.4871	0.4869
Recall	0.2590	0.5151	1.0000	0.2503	0.4893	0.9802
MAP		0.8635			0.5623	
MRR		0.9163			0.6578	

Under the broader relevance threshold (1), the system achieves high precision (P@5 = 0.8465) and perfect recall at @20, indicating effective retrieval coverage and ranking quality (MAP = 0.8635, MRR = 0.9163). With the stricter threshold (2), precision decreases substantially (P@5 = 0.4901), reflecting the challenge of satisfying both usage and sentiment matches simultaneously. Nonetheless, recall remains strong (Recall@20 = 0.9802), showing that the system retrieves a broad set of relevant items, though ranking quality diminishes (MAP = 0.5623, MRR = 0.6578). Notably, nDCG remains stable across thresholds, suggesting consistent ranking order despite changes in relevance criteria.

B. Meme Template Retrieval Evaluation Results

Table II shows the results for meme template retrieval.

TABLE II
EVALUATION RESULTS FOR MEME TEMPLATE RETRIEVAL

Metric	Threshold = 1 (Relevant)		Threshold = 2 (Highly Relevant)	
	@5	@10	@5	@10
nDCG	0.5008	0.7029	0.5008	0.7029
Precision	0.4960	0.4697	0.3714	0.3417
Recall	0.5233	1.0000	0.5021	0.9657
MAP		0.5982		0.4944
MRR		0.6558		0.5509

Template retrieval exhibits lower precision and ranking metrics compared to meme retrieval, with moderate precision at top ranks (P@5 = 0.4960 at Threshold 1) and recall reaching 1.0 at @10. While stricter relevance criteria reduce precision further, recall remains above 0.96. The reasons for this performance gap are discussed in detail in the Discussion section.

C. Precision-Recall Analysis for Memes and Meme Templates

Figures 12 and 13 display precision-recall curves for both tasks. For meme retrieval, PR-AUC is 0.87 under Threshold 1 and decreases to 0.57 under Threshold 2, indicating reduced precision at high recall with stricter relevance. Template retrieval curves show lower PR-AUC values (0.68 and 0.57), consistent with the observed performance drop relative to memes.

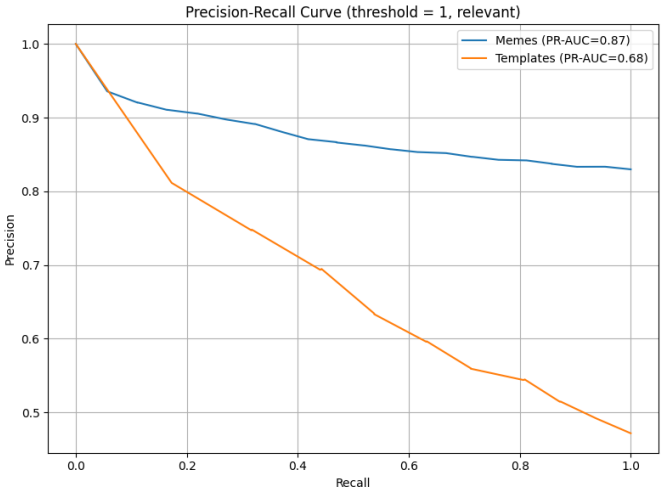

Fig. 12. Precision-Recall Curve (Threshold = 1)

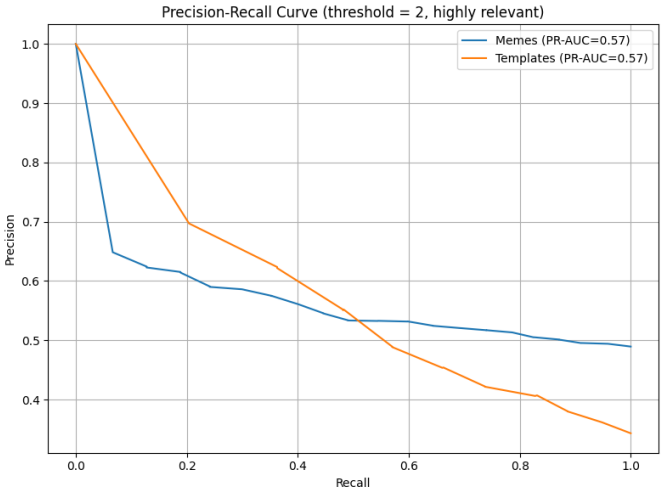

Fig. 13. Precision-Recall Curve (Threshold = 2)

D. MRR vs. Query Length for Memes and Meme Templates

Figures 14 and 15 analyze MRR across query lengths. Meme retrieval maintains high MRR under Threshold 1 but decreases with Threshold 2, suggesting that highly specific queries pose greater challenges in early relevant retrieval. Template retrieval exhibits significantly more fluctuation, reflecting the inherent difficulty in meme template retrieval; further discussion is provided in the Discussion section.

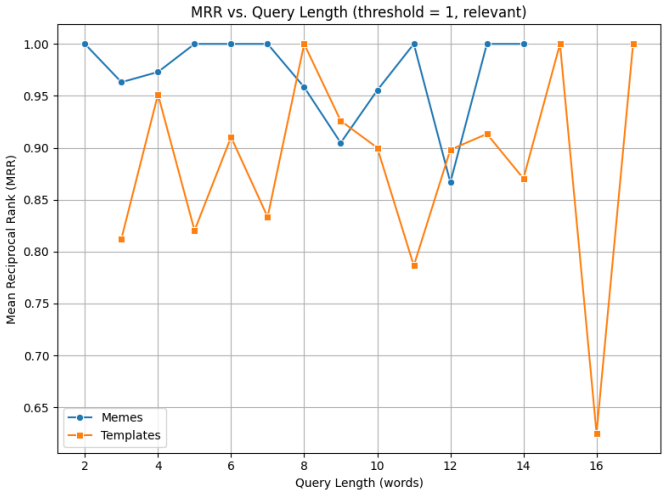

Fig. 14. MRR vs. Query Length (Threshold = 1)

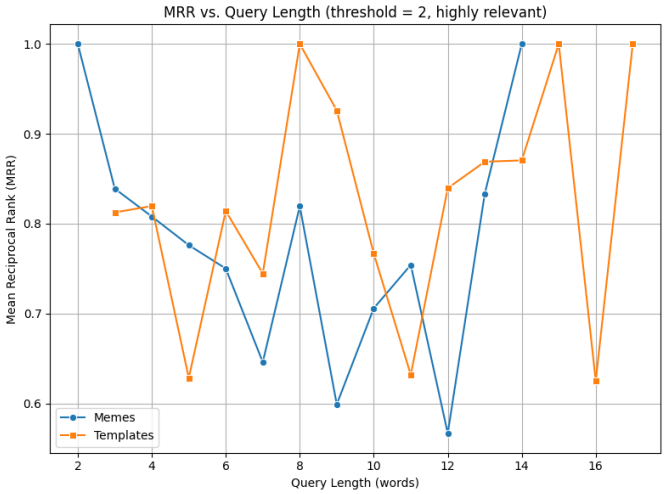

Fig. 15. MRR vs. Query Length (Threshold = 2)

Overall, our system demonstrates strong effectiveness in retrieving relevant memes with robust ranking quality under broad relevance criteria, while stricter definitions and template retrieval remain more challenging but still effective.

VII. Discussion

We did not implement a baseline model for this task because standard baselines (e.g., BM25, TF-IDF, or sentence embedding cosine similarity) are unlikely to provide meaningful insight in our context. Our system is designed around structured intent extraction—decomposing queries into usage, sentiment, and template preference—and retrieving memes accordingly. A naïve baseline ignoring this structured decomposition would neither capture the intent effectively nor offer a fair comparison, since the core of our contribution lies in this multi-stage, semantically-aware retrieval approach.

The evaluation results demonstrate that the system performs effectively for both meme and meme template retrieval tasks. As expected, performance is higher under the more lenient relevance threshold (Threshold = 1). Meme retrieval consistently outperforms meme template retrieval across all major metrics. Notably, the system achieves high MAP (0.8635) and MRR (0.9163) for meme retrieval under Threshold = 1, compared to lower but still reasonable scores for template retrieval (MAP 0.5982, MRR 0.6558).

The performance difference between meme and meme template retrieval can be attributed to two key factors. Firstly, the inherent nature of meme templates means a single template can spawn countless meme variations with diverse textual overlays. This abstractness can make it more challenging to directly match a specific query to a template compared to matching to a fully instantiated meme which often contains more explicit textual cues relevant to the query. Secondly, the disparity in dataset sizes significantly impacts performance; our meme template collection consists of approximately 2,100 items, whereas the meme dataset is substantially larger with over 301,000 instances. Such a smaller dataset for templates likely offers sparser data for learning effective retrieval models, potentially leading to the observed weaker performance in template retrieval tasks.

An interesting observation is that nDCG scores remain relatively stable across both thresholds for both tasks. This suggests that while the number of items considered relevant changes with stricter relevance criteria, the relative ranking order assigned by the system is consistent. This stability indicates that the system's scoring function captures meaningful ranking signals regardless of the strictness of the evaluation.

The Precision-Recall curves (Figures 12, 13) confirm the expected trade-off: high recall can be achieved, but precision drops more sharply when stricter relevance definitions are used. The analysis of MRR versus query length (Figures 14, 15) reveals variability in how quickly the system returns a relevant item depending on query length. This suggests that some queries are harder to interpret correctly, and that improvements in query understanding or query reformulation (e.g., through synonym expansion or paraphrase recognition) could help stabilize performance across diverse queries.

Finally, the system achieves perfect recall (Recall@20 = 1.0000) for meme retrieval under Threshold = 1 and Recall@10 = 1.0000 for template retrieval under the same threshold. This is a promising sign that users are highly likely to find at least one relevant result with a modest number of retrieved items.

VIII. Summary

We developed a multimodal meme retrieval and recommendation system that supports both text-to-image and image-to-image queries. For text queries, our system performs structured intent extraction—parsing the input into usage, sentiment, and template preference—to retrieve relevant memes or meme templates using a two-stage semantic matching and vector search pipeline. For image queries, we extract both local (textual) and global (visual) features to identify and recommend visually and contextually similar memes. We constructed a

semi-automated relevance-labeled test dataset and evaluated the system using MAP, MRR, nDCG, and Recall@k under different relevance thresholds. The system achieves strong performance in meme retrieval and reasonable performance in template retrieval, with the latter influenced by dataset size and task nature. Our results demonstrate the effectiveness of structured query understanding and multimodal similarity in enhancing meme recommendation accuracy.

REFERENCES

[1] Jaided AI. Easyocr: Ready-to-use ocr with 80+ supported languages and all popular writing scripts. https://github.com/JaidedAI/EasyOCR, 2020. Accessed: 2025-05-05.

[2] Jose Camacho-Collados, Kiamehr Rezaee, Talayeh Riahi, Asahi Ushio, Daniel Loureiro, Dimosthenis Antypas, Joanne Boisson, Luis Espinosa-Anke, Fangyu Liu, Eugenio Martínez-Cámara, et al. TweetNLP: Cutting-Edge Natural Language Processing for Social Media. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Abu Dhabi, U.A.E., November 2022. Association for Computational Linguistics.

[3] Arjun Chandrasekaran, Mark Yatskar, Yonatan Bisk, Ali Farhadi, and Luke Zettlemoyer. Do you have a moment to talk about my model's sentiment? understanding the role of sentiment in meme generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3248–3264, 2021.

[4] Yuning Du, Jinhui Li, Ruoyu Yan, et al. Paddleocr: An ultra lightweight ocr system. https://github.com/PaddlePaddle/PaddleOCR, 2020. Accessed: 2025-05-05.

[5] Douwe Kiela, Hamed Firooz, Anjali Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[6] Junnan Li, Dongxu Li, Caiming Xiong, and Steven CH Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022.

[7] Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-collados. TimeLMs: Diachronic language models from Twitter. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 251–260, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[8] Alec Radford, Jong Wook Kim, M Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[9] Ray Smith. An overview of the tesseract ocr engine. *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR)*, pages 629–633, 2007.

[10] Shubham Suryawanshi, Bharathi Raja Chakravarthi, Anand Kumar Madasamy, and John P. McCrae. Multimodal sentiment analysis of memes. *arXiv preprint arXiv:2004.14355*, 2020.