國立臺灣大學管理學院資訊管理學研究所

碩士論文

Department of Information Management

College of Management

National Taiwan University

Master's Thesis

StreaMeme：基於大型語言模型之直播迷因類別推薦

StreaMeme: Meme Category Recommendation of Livestreaming Using LLMs

陳沛竹

Pei-Chu Chen

指導教授: 陳建錦 博士

Advisor: Chien Chin Chen, Ph.D.

中華民國 114 年 6 月

June, 2025

doi:10.6342/NTU202501843

國立臺灣大學碩士學位論文
口試委員會審定書

StreaMeme：基於大型語言模型之直播迷因類別推薦

StreaMeme: Meme Category Recommendation of Livestreaming Using LLMs

本論文係陳沛竹君（學號 R12725022）在國立臺灣大學資訊管理學系、所完成之碩士學位論文，於民國 114 年 6 月 27 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

所　　長：

doi:10.6342/NTU202501843

謝辭

本論文得以順利完成，首先要感謝指導教授陳建錦老師的細心教導，謝謝老師在每週開會時都耐心理解我們的問題，並給予珍貴的建議，使得這項研究能夠臻於完備。也謝謝實驗室夥伴又昕、佳民、亭佑對於本研究的貢獻，能夠在國際會議發表論文，是相當可貴的經驗。特別感謝佳民，於碩士兩年期間，無論是課業、研究、助教等事務，當我遭遇困難或需要意見時，總是熱心相助並一起解決問題。非常慶幸自己能夠成為 WEAL Lab 的一份子，多虧有大家的合作與協助，我才得以順利完成學業。

另外，我想謝謝一路以來陪伴並關心我的朋友們，每當疲倦、焦慮時，你們的溫暖總能讓我感到安定並繼續前進。最後，感謝我的父母、哥哥、姐姐，在我最後的求學生涯，你們無條件的支持讓我可以全力投入研究、順利畢業，實在感激不盡。

陳沛竹 謹識

于國立臺灣大學資訊管理學研究所

民國一一四年六月

doi:10.6342/NTU202501843

摘要

迷因能夠傳達情緒與幽默感，並在直播過程中有效促進觀眾之間的互動及參與。然而，目前鮮少有研究協助直播主在直播環境中適時展示合適的迷因。由於需整合多模態線索、具備對直播情境的細緻理解，在直播過程中即時推薦相關迷因是一項極具挑戰性的任務。為此，本研究提出一套創新的系統 StreaMeme（liveStream Meme category recommender），旨在於直播過程中為直播主推薦合適的迷因類別。StreaMeme 利用視覺語言模型（Visual Language Model, VLM）以分析迷因的幽默感與情緒，並透過大型語言模型（Large Language Model, LLM）理解直播情境，進而推理出推薦合適迷因類別之理由。本系統所訓練之大型語言模型，可有效利用直播主之口述內容與觀眾留言，生成迷因解釋及直播情境推理，並將其輸出用於迷因類別推薦。實驗結果顯示，在真實直播資料中，StreaMeme 之 $F_{0.5}$ 分數優於多種基準模型，包含直接使用 LLM 提示及微調後的語言模型。

關鍵字：直播、迷因推薦、大型語言模型

doi:10.6342/NTU202501843

Abstract

Memes convey sentiments and humor, and are useful to stimulate interactions and engagements of audiences during livestreaming. However, little research aids streamers in showing appropriate memes in livestreaming environments. Suggesting relevant memes timely in a livestream poses significant challenges as it requires multimodal cues and a nuanced understanding of the livestream context. This study proposes StreaMeme (liveStream Meme category recommender), a novel system that recommends meme categories for streamers during livestreaming. StreaMeme employs a visual language model (VLM) to analyze memes' humor and sentiments, and digests livestreaming contexts with a large language model (LLM) to reason for meme recommendations. An LLM is fine-tuned to exploit streamer speeches, audience messages, meme explanations, and recommendation reasons; its output is then used for our meme category recommendation. Experimental results on real-world livestreams show that StreaMeme outperforms several baseline methods, including direct LLM prompting and fine-tuned language models in

doi:10.6342/NTU202501843

terms of the $F_{0.5}$ scores.

Keywords: Livestream, Meme Recommendation, Large Language Models

doi:10.6342/NTU202501843

Contents

doi:10.6342/NTU202501843

doi:10.6342/NTU202501843

List of Figures

doi:10.6342/NTU202501843

List of Tables

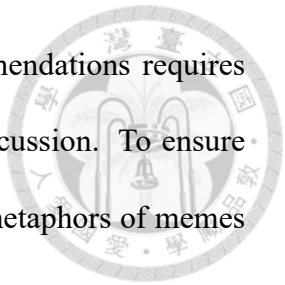
doi:10.6342/NTU202501843

Chapter 1 Introduction

Memes have become an essential communication tool for the younger generation. They usually combine images and text to convey humor, metaphors, sarcasms, or sentiments. In livestreaming, memes can enhance streaming atmosphere and increase the engagement between streamers and audiences [11]. Although many research studies have been conducted on meme analysis, particularly in areas such as multimodal feature extraction and its applications in semantics [22, 28, 34], sentiment analysis [1, 8, 24], and hateful content detection [2, 4, 7, 13], there has been limited focus of helping streamers show appropriate memes during livestreaming. To address this research gap, we explore the problem of meme category recommendation in livestreaming.

The meme category recommendation of livestreaming analyzes contexts of livestreaming to recommend appropriate meme categories for streamers. Here, the context refers to streamer speeches and audience feedback messages posted in the chatroom. Instead of individual memes, this study recommends meme categories. This is because individual memes are too specific to recommend. Also, by showing a set of memes of the recommended category, streamers can evaluate which memes entertain the audience the most. The meme category recommendation is challenging due to several factors. One key difficulty lies in the metaphorical meaning of memes [34] that involve rich interplay between a meme’ s text and image modalities. Comprehending a meme thus is not easy.

doi:10.6342/NTU202501843



Additionally, identifying right moments for meme category recommendations requires understanding both streamer speeches and the ongoing audience discussion. To ensure that the recommended meme categories are appropriate and timely, metaphors of memes as well as livestream contexts must be analyzed.

In this study, we develop StreaMeme (liveStream Meme category recommender), an effective meme category recommendation system for streamers during livestreaming. To address the above difficulties, the proposed method leverages a visual language model (VLM) that explains memes by extracting their metaphorical humor and sentiments. Next, we utilize a pre-trained large language model (LLM) to derive meme recommendation reasons of livestreaming segments. Finally, an LLM is fine-tuned to generate meme recommendation reasons and meme explanations according to streaming contexts. The outputs are then examined to recommend an appropriate meme category for a streamer.

Our main contributions are as follows:

- To the best of our knowledge, this is the first work that investigates the meme category recommendation of livestreaming. Also, LLMs and prompts are designed to comprehend metaphorical expressions of memes and livestreaming, and to recommend streamers meme categories timely and correctly.
- We compare StreaMeme with various baseline models, including direct prompting of pre-trained LLMs and fine-tuned language models. Experimental results demonstrate that StreaMeme surpasses these baselines, particularly excelling in meme category recommendation precision.
- Through our ablation study, we prove that the extracted meme explanations are helpful to guide our LLM in meme metaphor comprehension during fine-tuning.

doi:10.6342/NTU202501843

The fine-tuned LLM with meme explanations thus recommends meme categories successfully.

The remainder of this paper is organized as follows. First, we review related work of meme analysis and LLM reasoning. Next, the Methodology section details the proposed system. We evaluate StreaMeme's performance in the Experiments and Analysis section. Finally, we summarize our conclusions.

doi:10.6342/NTU202501843

Chapter 2 Related Work

2.1 Meme Analysis

As memes have become an influential form of communication on the Internet, researchers have proposed various methods to analyze memes in different perspectives. For instance, the Hateful Memes Challenge, initiated by Facebook [13], invites researchers to develop methods for identifying offensive content in memes. To address this task, numerous studies adopt multimodal strategies that integrate text and image features for classification [2, 4, 15]. Deshpande & Mani [4] developed classifiers based on gradient-boosted decision trees and LSTMs, utilizing a variety of input features. The textual features included embeddings of meme text, named entities, profanity, sentiment, emotion, and semantic similarity between meme text and image captions. The image-based features encompassed image captions and outputs from object detection and web entity detection. These high-level features align with the criteria humans use to evaluate whether a meme contains hateful content. By incorporating these features, their model achieved performance comparable to human judgment and transformer-based models. Lee et al. [15] introduced *DisMultiHate*, a framework leveraging self-supervised training to learn disentangled latent representations of text and images for hateful meme detection. The text representation module employs a BERT encoder, which processes a concatenation of web

doi:10.6342/NTU202501843

entities, demographic information detected in the meme, and the meme text. The visual representation module uses an attention-based image encoder, taking regions extracted by Faster R-CNN as input. The textual and visual representations are then combined and passed through a regression layer to estimate the likelihood of hatefulness. This approach outperformed several multimodal baselines in the hateful meme detection task. Similarly, Cao et al. [2] utilized the implicit knowledge of a pre-trained language model by inputting the meme text, image caption, a prompt template ("It was `[MASK]`"), and examples of positive and negative cases into RoBERTa. The meme is classified as hateful if the `[MASK]` token's probability of being the positive label exceeds that of the negative label. After fine-tuning this framework using cross-entropy loss, it achieved an impressive AUC of 90.96, demonstrating its effectiveness in hateful meme detection.

Beyond harmful content detection, researchers also explore meme semantics and sentiment analysis. For instance, Prakash et al. [22] introduced *PromptMTopic*, leveraging the language modeling capabilities of LLMs for topic modeling on memes. Similarly, studies by Alluri & Krishna [1] and Pranesh & Shekhar [24] adopt dual-stream methods that utilize visual and textual encoders to extract features, then classify memes into sentiment categories.

Moreover, many scholars have released public dataset to advance meme research in recent years [13, 17, 23, 27, 31, 34]. For example, Sharma et al. [27] launched a sentiment analysis challenge based on their dataset *MEMOTION*. Xu et al. [34] introduced *MET-Meme*, a multimodal meme dataset enriched with metaphorical features and annotated for metaphor occurrence, sentiment categories, intentions, and offensiveness degree. Liu et al. [17] presented *FigMemes*, a multi-label dataset designed for figurative language classification in politically opinionated memes. These contributions have significantly

doi:10.6342/NTU202501843

advanced the field by providing diverse resources for meme analysis.

2.2 Large Language Model Prompting and Reasoning

Large Language Models (LLMs) have demonstrated increasingly sophisticated reasoning capabilities through different prompting techniques, as demonstrated by recent research [10, 14, 33]. Wei et al. [33] introduced the concept of chain-of-thought (CoT) reasoning approach, which involves a series of intermediate reasoning steps, and compared model performance using CoT prompting against standard prompting. Their research demonstrated that LLMs significantly benefit from CoT prompting, achieving substantial performance improvements across complex arithmetic, commonsense reasoning, and symbolic reasoning tasks. Kojima et al. [14] demonstrated that LLMs perform effectively as zero-shot reasoners when preceded by the prompt "Let's think step by step" before each response. This approach highlights how simple prompting can unlock high-level, multi-task cognitive abilities in these models. With powerful reasoning capabilities, LLMs can identify and extract relevant evidence from given contexts and generate detailed explanations of their problem-solving processes. For example, Sun et al. [30] introduced *CARP*, which prompts LLMs to perform text classification tasks through progressive reasoning, achieving significant performance gains on widely-used benchmarks.

Additionally, many researchers have fine-tuned LLMs on datasets containing reasoning information, showing strong performance on downstream tasks [3, 16, 21, 32, 35]. For instance, Google's Flan models were instruction-finetuned on 1.8k tasks, incorporating data with chain-of-thought (CoT) reasoning. The experiments demonstrated that the CoT data is crucial to keep reasoning abilities and performances of these models [3].

doi:10.6342/NTU202501843

Similarly, Yu et al. [35] fine-tuned OPT [36] using data with rationales, enhancing model performance on logical and causal reasoning tasks. Meta introduced Galactica, an LLM designed to store, integrate, and reason over scientific knowledge. By incorporating a working memory token to structure step-by-step reasoning, Galactica achieved significant performance improvements over chain-of-thought prompting, even with reduced model capacity [32]. In the research of Lewkowycz et al. [16], they built a dataset of over 200 undergraduate-level questions in science and mathematics from MIT's OpenCourseWare (OCW). This dataset is used to train a LLM to solve mathematics, science, and engineering problems that require quantitative reasoning. The study found that the model can achieve better performance in a chain-of-thought context instead of a pure mathematical setting.

Building on these insights, we augmented our dataset with two types of reasoning data, e.g., meme explanations and recommendation reasons. We then format the instruction-response pairs with this augmented data to fine-tune our livestream meme recommendation model. Further details are provided in the next section.

doi:10.6342/NTU202501843

Chapter 3 Methodology

3.1 System Overview

StreaMeme suggests streamers meme categories (e.g., happiness or love), indicating that displaying memes of the recommended categories during live streaming would increase audience engagement. Rather than individual memes, we recommend meme categories. This is because individual memes are too specific to recommend. Also, by showing a set of memes of the recommended category, streamers can evaluate which memes entertain the audience the most. Figure 1 illustrates our system architecture, which consists of a *model construction phase* and a *meme category recommendation phase*. In the model construction phase, a set of livestream videos is collected and each is partitioned into a series of segments. Domain experts are invited to assess the streamer speeches and the feedback messages of the audiences in the chatroom to determine if it is appropriate for the streamer to show a meme at a segment. Moreover, if a meme is deemed appropriate, the system determines which meme category should be recommended. StreaMeme leverages LLMs to comprehend metaphors of memes and reasons of meme recommendations. To this end, the segments are paired with a sample meme of the annotated categories, and are fed into two auxiliary pre-trained models (i.e., Llama 3 8B Instruct [6] and LLaVA-1.6-34B [18]) to derive reasons of meme recommendations and meme explanations. These

doi:10.6342/NTU202501843

textual responses, along with streamer speeches and audience messages, are used to fine-tune our LLM. In the meme category recommendation phase, segments of a livestream are examined sequentially. The fine-tuned LLM processes the speech of the streamer and the feedback messages of the audiences to explain which meme category, including *none*, is suitable for recommendations. Below, we detail each system component.

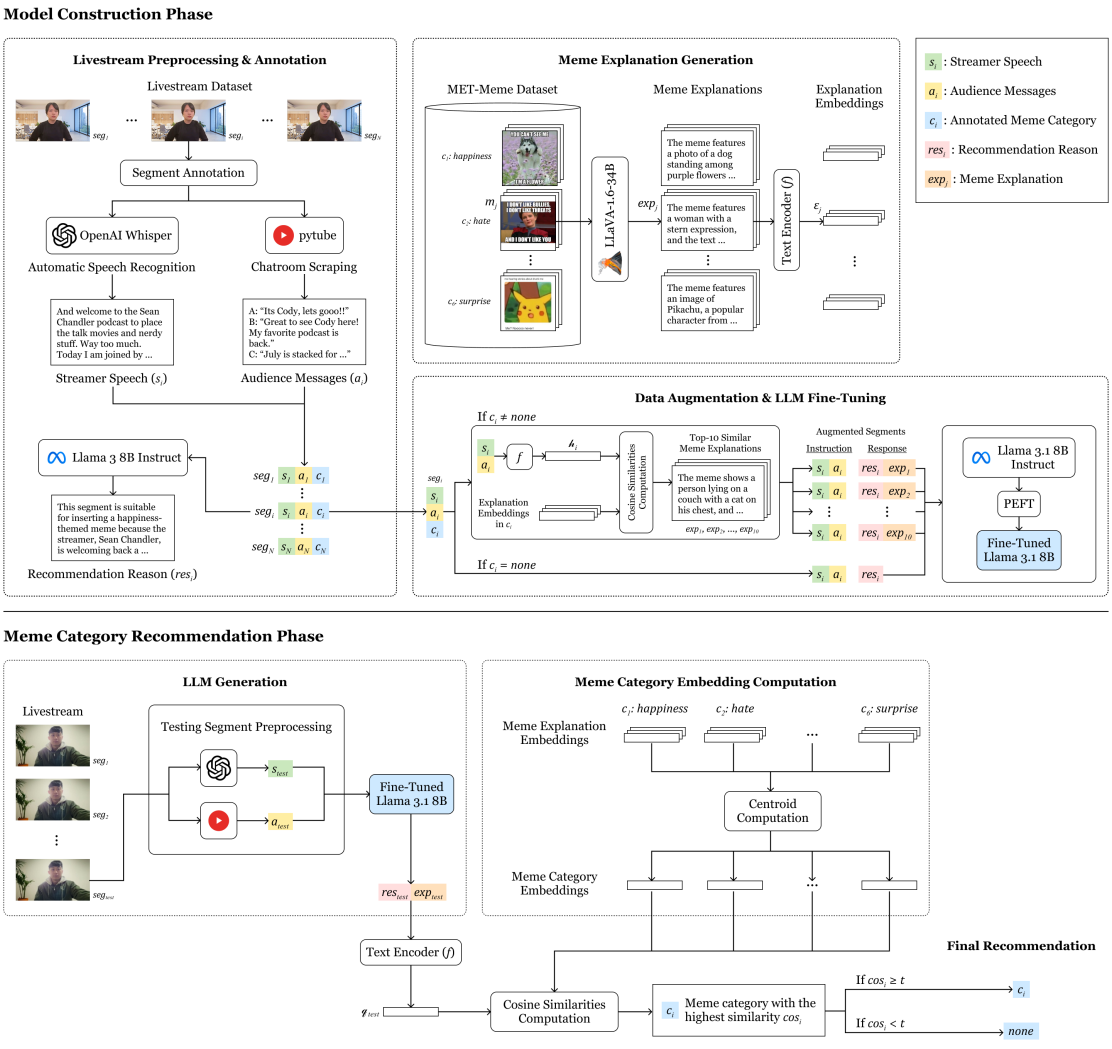


Figure 1: Overview of StreaMeme

doi:10.6342/NTU202501843

3.2 Model Construction Phase

3.2.1 Livestream Preprocessing and Meme Category Annotation

In the model construction phase, we divide the collected livestream videos into series of one-minute segments. For each segment seg_i, domain experts evaluate both the original livestream videos and the accompanying chatroom messages to determine whether seg_i is suitable for meme recommendation, and a meme category $c_i \in C$ is annotated. In this research, the set of meme categories, C, is based on the *MET-Meme* dataset [34] that six categories of memes covering sentiments of *happiness*, *love*, *anger*, *sorrow*, *hate*, and *surprise* are selected. In addition to the six sentiment categories, C includes a dummy label *none* indicating that a segment does not deserve a meme recommendation and $C = \{happiness, love, anger, sorrow, hate, surprise, none\}$. Since the decision of meme category is relatively subjective, each segment's ground truth label is determined by two domain experts. After the annotation, the streamer speech s_i is transcribed from the audio of streaming using OpenAI Whisper[1]. Additionally, the Python library *pytube*[2] is employed to extract the audience messages a_i from the streaming chatroom.

3.2.2 Recommendation Reason Generation

Meme category recommendation can intuitively be formulated as a text classification problem. In other words, the task assigns a meme category to a segment given the stream speech s_i and audience messages a_i. However, topics of livestreaming are very diverse and memes generally involve complex metaphors. It is therefore difficult to rec-

[1]https://github.com/openai/whisper
[2]https://github.com/pytube/pytube

doi:10.6342/NTU202501843

ommend appropriate meme categories using traditional text mining techniques which normally based on token matching. To recommend meme categories accurately, StreaMeme incorporates LLMs. By leveraging the reasoning ability of large language models, we expect StreaMeme to not only recommend appropriate meme categories but also provide explanations for its recommendations.

For each segment seg_i, the Meta Llama 3 8B Instruct model is prompted to consider both the streamer speech s_i, the audience messages a_i, and their interactions. If c_i of the segment is not *none*, we ask the pre-trained LLM to explain the reason of recommending meme category c_i, otherwise, the model explains the reason of not recommending. The recommendation reason res_i responded by the pre-trained model is recorded to fine-tune our LLM of the meme category recommendation. Each segment is then represented as $seg_i = \langle s_i, a_i, c_i, res_i \rangle$ and $S = \{seg_1, seg_2, seg_3, \ldots, seg_N\}$ is the set of the annotated segments used to construct our LLM of meme category recommendation.

3.2.3 Meme Explanation Generation

In addition to the recommendation reasons above, we extract meme explanations to help our LLM comprehend meme metaphors. The *MET-Meme* dataset mentioned earlier collects a lot of memes. We sample 30 memes for each sentiment category $c_i \in C$. The resulting 180 memes then are fed into the pre-trained multimodal model LLaVA-1.6-34B to obtain metaphorical explanations of the memes.

Let img_j be the raw image of a sampled meme, m_j. We prompt LLaVA-1.6-34B with m_j and a text instruction asking the model to explain why the meme belongs to the specified sentiment category c_j and the humor of the meme. LLaVA-1.6-34B has

doi:10.6342/NTU202501843

demonstrated extraordinary performance in many multimodal understanding and reasoning benchmarks[3]. The generated explanation, denoted as exp_j, is meaningful and typically describes the interplay between the superimposed text and visual elements in img_j. Based on the replies, the sampled meme is represented as $m_j = \langle img_j, exp_j, c_j \rangle$ and a meme explanation dataset M is formed and $M = \{m_1, m_2, \ldots, m_{180}\}$.

3.2.4 Large Language Model Fine-Tuning

In StreaMeme, we fine-tune an LLM to help it reason whether a segment deserves a meme category recommendation. Figure 2 outlines the function of the fine-tuned LLM. Specifically, given the context of a segment, that is, the streamer speech and the audience messages, the fine-tuned LLM outputs the reason of recommending a meme category (or the reason of not recommending one) and the corresponding meme explanation. The output will be examined in the meme category recommendation phase to generate the final meme category recommendation.

We select the Meta Llama 3.1 8B Instruct model [6] as our base model. To fine-tune it, we construct a training dataset consisting of a set of instruction-response pairs. For each segment $seg_i = \langle s_i, a_i, c_i, res_i \rangle$ in S, an instruction-response pair is formatted by chaining up the following information:

- The input instruction generated by simply applying a template "*Streamer Speech:* $\{s_i\}$ *Audience Messages:* $\{a_i\}$".

[3] https://llava-vl.github.io/blog/2024-01-30-llava-next/

doi:10.6342/NTU202501843

- If c_i is *none*, the response is just res_i; otherwise, we sample a meme m_j of category c_i in M and format the response by the template "$\{res_i\}$ *This segment is suitable for inserting a meme in the* c_i *category. The description and explanation of a sample meme may be like the below paragraphs:* $\{exp_j\}$".

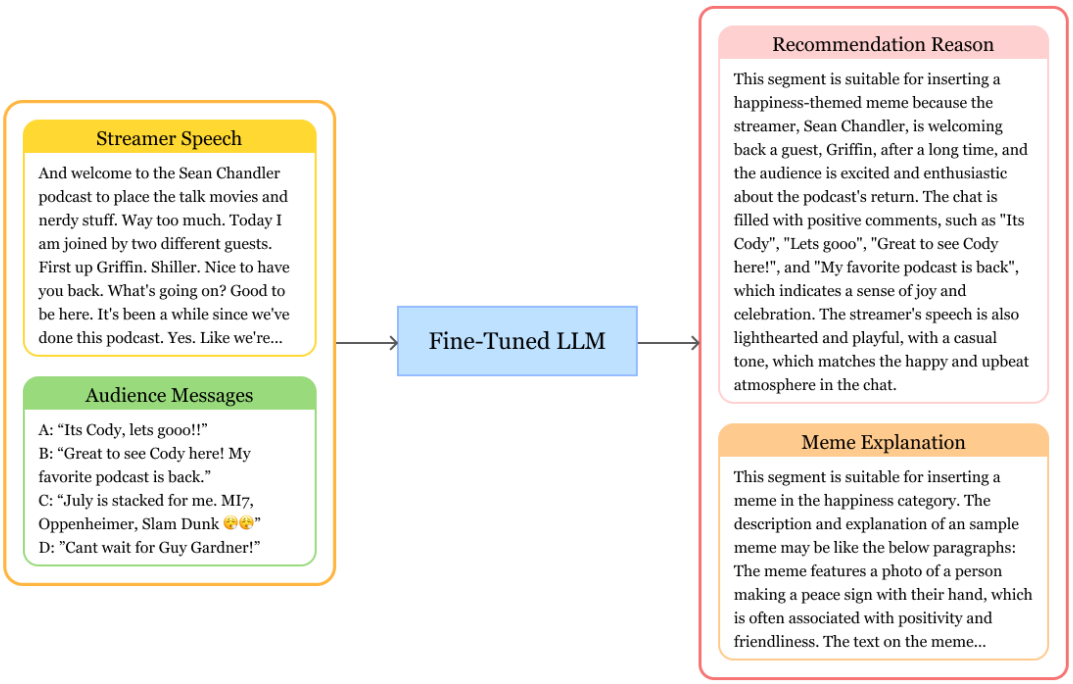


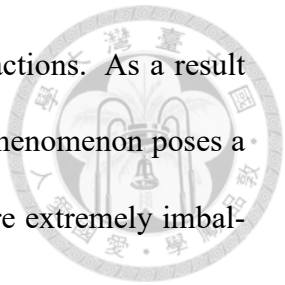
Figure 2: Sample Input and Output of the Fine-Tuned LLM

The base model, Meta Llama 3.1 8B Instruct, is fine-tuned with the training data using the Parameter-Efficient Fine-Tuning (PEFT) method, specifically Low-Rank Adaptation (LoRA) [9]. The approach is adopted because it has been proven to be an effective fine-tuning method that significantly reduces the computational cost [37].

3.2.5 Training Data Augmentation

When preparing the above training dataset, we noticed the meme category distribution was highly skewed. This is because livestreams are normally lengthy, and most

doi:10.6342/NTU202501843



segments have a neutral tone of speech and peaceful audience interactions. As a result of this, few segments are assigned a sentiment category label. This phenomenon poses a challenge for our fine-tuning, as the categories in the training data are extremely imbalanced.

We design a data augmentation method to generate new instruction-response pairs of sentiment categories. For each meme $m_j = \{img_j, exp_j, c_j\}$ in M, we employ a text encoder f to encode exp_j into an explanation embedding vector ε_j. Here, the all-mpnet-base-v2 model based on the SBERT framework [26] is used as f. Next, we examine the collected segments in S individually. For a segment seg_i whose c_i is not *none*, the same text encoder f is used to encode $s_i \circ a_i$ into an embedding vector $\hbar_i$, where $\circ$ denotes text concatenation. Then, we compute the cosine similarities between $\hbar_i$ and the explanation embeddings in M to identify the top-10 similar memes of sentiment category c_i. Their meme explanation exp_j's are individually combined with the segment's streamer speech s_i, audience message a_i, sentiment category c_i, and recommendation reason res_i to form 10 instruction-response pairs. This approach increases the number of instruction-response pairs regarding each sentiment category tenfold, effectively mitigating the data imbalance issue in fine-tuning.

3.3 Meme Category Recommendation Phase

In the meme recommendation phase, the fine-tuned LLM is used to evaluate a livestream. The bottom half of Figure 1 illustrates the process of our meme category recommendation. As in the model construction phase, we sequentially examine the one-minute segments of a livestream. Let seg_{test} be the segment that currently under examination.

doi:10.6342/NTU202501843

We first extract the streamer speech s_{test} and audience messages a_{test} from the segment. The textual context is formatted using the template "*Streamer Speech:* $\{s_{test}\}$ *Audience Messages:* $\{a_{test}\}$" to create an instruction prompt for the fine-tuned LLM. As shown in Figure 2, the response of the prompting would explain why the segment deserves (or does not deserve) a meme category recommendation, together with the explanation of a suitable meme. Since LLMs generate responses in a stochastic manner, we do not match the response of our LLM with specific keywords (e.g., *happy*) to recommend meme categories. Instead, we use the SBERT-based sentence encoder f to convert the response into a query embedding q_{test}.

For each sentiment category of M, we compute the average of the meme explanation embedding ε_j's in that category. The centroid embeddings of the sentiment categories are then measured to compute their cosine similarities to q_{test}, and the category with the highest similarity is recommended. Note that if the highest similarity is below a predefined threshold t, we assign a *none* label to the segment, indicating that the segment is not suitable for a meme category recommendation. In the Experiments and Analysis section, we evaluate the effect of t on the system performance.

doi:10.6342/NTU202501843

Chapter 4 Experiments and Analysis

In this section, we first introduce the experiment dataset, evaluation procedure, and performance metrics. Then, we compare StreaMeme with baseline methods. Last, system parameters are evaluated to verify the effect of the designed components.

4.1 Dataset, Evaluation, and Metrics

To the best of our knowledge, StreaMeme is the first study to investigate the meme category recommendation problem. Since no existing dataset aligns with our research, we evaluate system performance using our own dataset. We constructed an experiment dataset by collecting 12 livestream videos from four YouTube streamers, primarily focusing on stream category “just chatting”. While there are many video game livestreams online, we do not use them for evaluations. This is because the audiences of game streaming love watching game playing. Thus, insertions of memes during game livestreams can be disruptive. The collected livestream videos covered various topics, and the streamers engaged actively with their audiences through chatrooms and messages. We partitioned the videos into series of one-minute segments. Then, domain experts were invited to annotate the meme categories. We adopted the sentiment categories defined in the *MET-Meme* dataset [34] where six meme categories including *happiness*, *love*, *anger*, *sorrow*, *hate*,

doi:10.6342/NTU202501843

and *surprise* were selected for evaluations. For each segment, the experts evaluated the content presented by the streamer (i.e., the streamer's speech) and the audience interactions (i.e., the audience messages) to determine its appropriate meme category. A "*none*" category label was assigned if the experts determined the segment did not warrant a meme recommendation. In order to test the generality of StreaMeme, we further divided the sentiment categories into two polarity groups. The positive includes sentiment categories of *happiness* and *love*, and the negative contains *anger*, *sorrow*, and *hate*. Note that category *surprise* is neutral, we thus exclude the segments with a *surprise* category label when testing the polarity-based meme category recommendations. Table 1 shows the statistics of the experiment dataset and the distribution of meme categories. In average, each livestream video has 155 segments. The average token lengths of streamer speeches and audience messages in a segment are 220.03 and 257.94, respectively.

Table 1: The Statistics of the Experiment Dataset

Statistic	**Value**
Number of Livestream Videos	12
Length of Videos (min.)	1869.30
Total Number of Segments	1863
Average Number of Audience Messages in a Segment	21.16
Average Number of Message Tokens in a Segment	257.94
Average Number of Speech Tokens in a Segment	220.03

Number of Segments in Each Category

None	Happiness	Hate	Love	Sorrow	Anger	Surprise
1764	33	16	13	13	12	12

We adopted the 10-fold cross-validation to evaluate the performance of StreaMeme. Specifically, we randomly partitioned the segments in 10 subsets and evaluated system performance in 10 runs. Each run evaluated one subset of the segments, and the remaining segments were used for our model construction. The results of the 10 runs were averaged to report the global system performance. The performance metrics include precision and

doi:10.6342/NTU202501843

recall. Note that the precision and recall scores are averaged by means of the $F_{0.5}$ measure. In practice, meme category recommendations need to be accurate because displaying an irrelevant meme would ruin livestreaming atmosphere and further diminishes user experience. The $F_{0.5}$ measure emphasizes precision over recall, and is well-accepted for evaluating the overall performance of a task that penalizes false positives (i.e., irrelevant meme category recommendations) more than false negatives (i.e., missed meme category recommendations) [20, 25]. For each meme category c, the precision, recall, and $F_{0.5}$ scores are calculated as follows:

$$\text{Precision} = \frac{|\hat{S}_c \cap S_c|}{|\hat{S}_c|}$$

$$\text{Recall} = \frac{|\hat{S}_c \cap S_c|}{|S_c|}$$

$$F_{0.5} = (1 + 0.5^2) \times \frac{\text{Precision} \times \text{Recall}}{(0.5^2 \times \text{Precision}) + \text{Recall}},$$

where $\hat{S}_c$ stands for the set of segments that are recommended to category c. Symbol S_c is the set of segments that are annotated to category c. The macro average is adopted to report the performance scores across all meme categories.

4.2 Comparison with Baseline Methods

Since meme category recommendation involves predicting the category label of a segment, it can be formulated as a classification problem. We therefore compare two baselines that directly prompt an LLM to perform the classification task. Additionally, two supervised classification baselines that utilize the training segments to fine-tune a language model are evaluated. Note that in our meme category recommendation phase,

doi:10.6342/NTU202501843

the input of StreaMeme consists of the streamer speech of a testing segment and the corresponding audience messages. To obtain fair comparison results, each of the following baselines simply takes a segment's streamer speech and audience messages to predict meme categories.

- Llama3.1$_{\text{zero-shot}}$:

  For each testing segment, a clear task description followed by the streamer speech and the audience messages is compiled to form a prompt. The prompt is inputted to the Meta Llama 3.1 8B Instruct model to predict the meme category without demonstrations.

- Llama3.1$_{\text{few-shot}}$:

  This baseline enhances Llama3.1$_{\text{zero-shot}}$ by including a few of demonstrations in the formatted prompt.

- BERT$_{\text{base}}$:

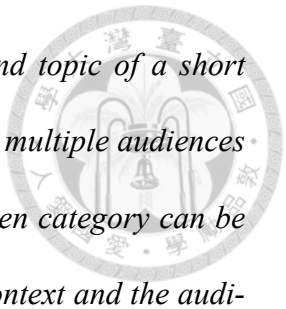
  The baseline utilizes the streamer speeches, the audience messages, and the meme category labels of the training segments to fine-tune BERT (Bidirectional Encoder Representations from Transformers) [5] with a downstream classification task. The fine-tuned BERT model then takes the streamer speech and the audience messages of a testing segment as input and predicts the meme category as the output.

- RoBERTa$_{\text{base}}$:

  Similar to BERT$_{\text{base}}$, this supervised classification baseline fine-tunes RoBERTa (Robustly Optimized BERT Approach) [19] to perform the meme category recommendation.

The prompts of Llama3.1$_{\text{zero-shot}}$ and Llama3.1$_{\text{few-shot}}$ are based on the following template:

doi:10.6342/NTU202501843



"*You are an AI assistant tasked with identifying the context and topic of a short livestream segment based on the streamer's speech and the chat from multiple audiences in the chatroom. Your goal is to determine whether a meme in a given category can be inserted into the livestream segment. Consider both the streamer's context and the audience's interactions. Your evaluation should fall into one of the following categories:* $\{L\}$*. Use the "none" category if the segment is unsuitable for a meme. Provide only one of the categories without any additional content.*"

Symbol L denotes the set of meme categories. For the sentiment-based meme category recommendation, it is $\{happiness, love, anger, sorrow, hate, surprise\}$. For the polarity-based recommendation, $L = \{positive, negative\}$. In the few-shot baseline, we randomly sampled one demonstration for each meme category, including *none*. As a result, the prompt for the sentiment-based recommendation includes 7 shots, while the prompt for the polarity-based recommendation task contains 3 shots. For our method, the similarity threshold t used for meme category recommendations is set at 0.7. Later, we examine the effect of t under different parameter settings.

Table 2 shows the performances of the sentiment-based and polarity-based meme category recommendations. The best and the second-best results of each evaluation metric are boldfaced and underlined, respectively. Compared to the performance scores of the polarity-based recommendation, the scores of the sentiment-based recommendation are low. This is because the categories of the sentiment-based recommendation involve deep emotions that the differences between some of them are subtle (e.g., happiness vs. love). It is therefore not easy to predict sentiment categories. Nevertheless, for the sentiment-based meme category recommendation, the $F_{0.5}$ score of our method is significantly better than those of the baselines. Without fine-tuning, the pre-trained LLMs of Llama3.1$_{\text{zero-shot}}$

doi:10.6342/NTU202501843

and Llama3.1$_{\text{few-shot}}$ cannot comprehend complex metaphor of meme categories, speeches of streamers, and audience messages mentioned in the given prompts. Though few-shot examples are provided, the LLMs can merely answer some plain and simple cases. As a result of this, the precision scores of Llama3.1$_{\text{zero-shot}}$ and Llama3.1$_{\text{few-shot}}$ are suboptimal.

Table 2: The Comparisons between StreaMeme and the Baselines

Sentiment-Based Meme Category Recommendation			
	Recall	**Precision**	$\boldsymbol{F_{0.5}}$
StreaMeme	<u>0.225</u>	**0.224**	**0.215**
Llama3.1$_{\text{zero-shot}}$	0.207	0.203	<u>0.182</u>
Llama3.1$_{\text{few-shot}}$	**0.251**	<u>0.204</u>	0.170
BERT$_{\text{base}}$	0.171	0.154	0.149
RoBERTa$_{\text{base}}$	0.202	0.165	0.165
Polarity-Based Meme Category Recommendation			
	Recall	**Precision**	$\boldsymbol{F_{0.5}}$
StreaMeme	<u>0.456</u>	<u>0.498</u>	**0.478**
Llama3.1$_{\text{zero-shot}}$	0.431	0.384	0.290
Llama3.1$_{\text{few-shot}}$	**0.470**	0.421	0.379
BERT$_{\text{base}}$	0.426	0.416	0.410
RoBERTa$_{\text{base}}$	0.442	**0.501**	<u>0.467</u>

Under the sentiment-based recommendation, BERT$_{\text{base}}$ and RoBERTa$_{\text{base}}$ are inferior to Llama3.1$_{\text{zero-shot}}$ and Llama3.1$_{\text{few-shot}}$. However, the former two baselines outperform the latter two under the polarity-based recommendation. We speculate the size of training data affected the fine-tuning of BERT$_{\text{base}}$ and RoBERTa$_{\text{base}}$. In the sentiment-based recommendation, the training segments of each sentiment category are too few to fine-tune the language models. By aggregating the training segments of sentiments, BERT$_{\text{base}}$ and RoBERTa$_{\text{base}}$ successfully fine-tuned their language models and therefore achieved good performance results in the polarity-based recommendation.

StreaMeme outperforms BERT$_{\text{base}}$ and RoBERTa$_{\text{base}}$ significantly. This is because our fine-tuning exploits meme explanations and recommendation reasons. We found that

doi:10.6342/NTU202501843

the topics of the evaluated livestreaming are very different. The contexts of the training segments (i.e., the streamer speeches and the audience messages) are diverse that make LLMs hard to infer the association between meme categories and streaming context. The auxiliary explanations of meme and recommendation reasons help StreaMeme capture the humor and metaphor embedded in livestreaming. Consequently, StreaMeme achieves the best performance score. This comparison result also validates the value of fine-tuning when dealing with sophisticated meme category recommendations.

Compared to the scores of the sentiment-based meme category recommendation, nearly all scores of the methods in the polarity-based meme category recommendation improved. As mentioned above, some sentiment categories are very close. Hence, in the sentiment-based recommendation segments of these categories often mis-classified. By grouping these sentiments into positive or negative polarities, the meme category recommendation problem becomes easier to solve. Therefore, the results of the methods improved. Again, under the polarity-based setting, our method is still the best in terms of $F_{0.5}$.

To further test the robustness of our method, we conducted a binary meme recommendation experiment. In this experiment, segments with a *none* category label are treated as "*not recommended*" and the others are regarded as "*recommended*". The recommended binary labels are compared with the ground truth to report system performances. We designed this experiment to see if StreaMeme is capable of reminding streamers moments suitable for meme recommendations.

Table 3 shows that our method surpasses the baseline methods again. Note that our precision score (0.528) has improved against the polarity-based experiment. We found

doi:10.6342/NTU202501843

that metaphors of streamers speeches and audience messages sometimes are difficult to reason, especially those with a sarcastic sense. As a result, our LLM predicted a wrong polarity label. While the predicted polarity label was opposing, our LLM still identified appropriate moments for meme recommendations. Hence, the precision score improved.

Table 3: The Comparison under the Binary Meme Recommendation

	Recall	**Precision**	$\boldsymbol{F_{0.5}}$
StreaMeme	0.356	**0.528**	**0.479**
$\text{Llama3.1}_{\text{zero-shot}}$	<u>0.708</u>	0.376	0.415
$\text{Llama3.1}_{\text{few-shot}}$	**0.859**	0.390	<u>0.437</u>
$\text{BERT}_{\text{base}}$	0.598	<u>0.414</u>	<u>0.437</u>
$\text{RoBERTa}_{\text{base}}$	0.502	0.386	0.402

4.3 Effect of the Similarity Threshold

Figure 3 shows the performance of StreaMeme under different settings of similarity thresholds t. In general, a large threshold leads to a strict meme category recommendation. In other words, when t is large, a testing segment will not be classified to any category unless the output embedding of our fine-tuned LLM is very similar to one of the meme category embeddings. As shown in the figure, the recommendation performance improves as t increases. However, setting t too high degrades our recommendation performance. Since a large t would misclassify a lot of testing segments into the *none* category, the precision and recall scores decline accordingly. We suggest setting t at 0.7 because of its superior $F_{0.5}$ performance.

doi:10.6342/NTU202501843

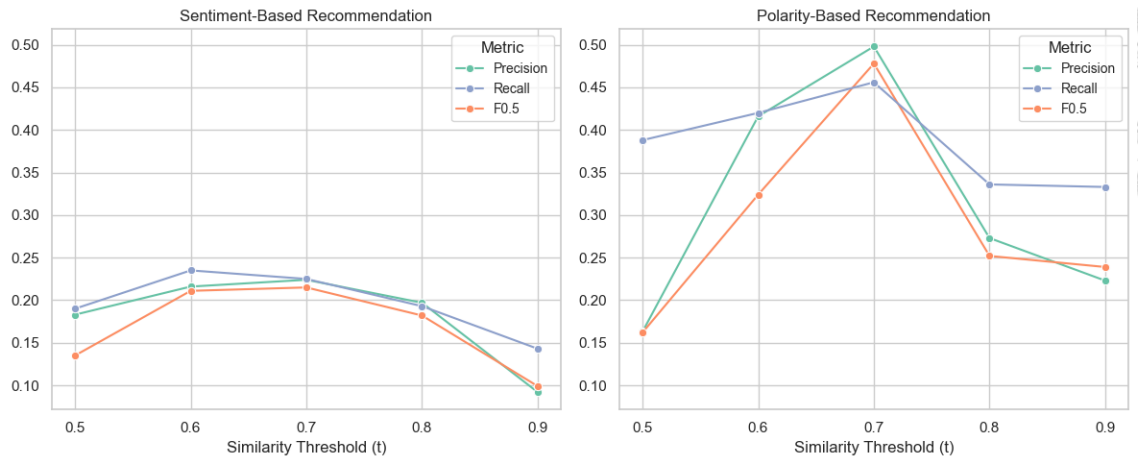


Figure 3: Performance under Different Similarity Threshold Settings

4.4 Effect of Meme Explanations

Finally, we conduct an ablation study to assess the importance of meme explanations. Table 4 shows the performances of StreaMeme with and without meme explanations. When without meme explanations, only the recommendation reasons of the training segments are used in the instruction fine-tuning stage. Accordingly, the fine-tuned LLM simply outputs the recommendation reasons of the testing segments for meme category recommendations. As shown in the table, the performance scores of StreaMeme decline if meme explanations are excluded. As the extracted explanations reveal metaphorical information of memes, incorporating them in the fine-tuning process provides valuable guidance in associating streaming contexts with sentiment categories. Our system performance thus deteriorates if this valuable information is missing.

doi:10.6342/NTU202501843

Table 4: Experiments Results with and without Meme Explanations

Sentiment-Based Meme Category Recommendation			
	Recall	**Precision**	$F_{0.5}$
StreaMeme	0.225	0.224	0.215
StreaMeme$_{\text{-explanation}}$	0.200	0.200	0.189
Polarity-Based Meme Category Recommendation			
	Recall	**Precision**	$F_{0.5}$
StreaMeme	0.456	0.498	0.478
StreaMeme$_{\text{-explanation}}$	0.420	0.518	0.459

4.5 Recommendation Time Analysis

To evaluate the applicability of our method in real-time streaming scenarios, we measured the processing time of each component involved in the meme category recommendation phase. All tests were conducted on an NVIDIA GeForce RTX 4090 GPU. For a one-minute livestream segment, transcribing audio into streamer speech using OpenAI Whisper takes an average of 1.61 seconds. The fine-tuned Llama 3.1 8B model requires an average of 2.23 seconds to generate reasoning output. Finally, encoding the output into a query embedding and computing the similarity scores against all meme category embeddings takes approximately 0.005 seconds on average. These results indicate that our method operates with minimal latency and is well-suited for real-time recommendation in livestreaming environments.

doi:10.6342/NTU202501843

Chapter 5 Conclusion

In this paper, we have developed StreaMeme, an innovative system that leverage LLMs for livestream meme category recommendations. Our system involves fine-tuning a large language model (LLM) using streamer speeches, audience messages, recommendation reasons, and meme explanations. The final recommendation label is determined through computing the cosine similarities between the LLM output and the centroid embeddings of each meme category. Experimental results demonstrate that StreaMeme surpasses four baseline models including direct prompting of LLMs and supervised fine-tuning of language models, which highlights its effectiveness for real-world livestream applications.

The research is subjected to the following limitations. First, the recommendation reasons and meme explanations are generated by prompting the pre-trained LLM and VLM. Although our instruction templates work well, there is still a room for prompt engineering. As responses of LLMs can be sensitive to the prompt templates [12, 29], we will explore optimal prompts in an automated way. Now, the memes we used are based on an existing dataset. Memes quickly become outdated. To stay current, we will create and update our meme dataset periodically. Additionally, while we have collected the first dataset for evaluating meme category recommendations, its size remains limited. Data annotation is time consuming. In the future, the experiment dataset will be extended and

doi:10.6342/NTU202501843

will also be released to stimulate future research. Finally, all evaluations have been conducted offline using annotated data. As a key direction for future work, we plan to build a complete system implementation of our method, enabling real-time deployment and user assessment. This will help us better evaluate the practical effectiveness, appropriateness, and user engagement potential of our meme recommendation framework in live streaming environments.

doi:10.6342/NTU202501843

References

[1] N. V. Alluri and N. Dheeraj Krishna. Multi Modal Analysis of memes for Sentiment extraction. In *2021 Sixth International Conference on Image Information Processing (ICIIP)*, pages 213–217, Shimla, India, Nov. 2021. IEEE.

[2] R. Cao, R. K.-W. Lee, W.-H. Chong, and J. Jiang. Prompting for Multimodal Hateful Meme Classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 321–332, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics.

[3] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei. Scaling Instruction-Finetuned Language Models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[4] T. Deshpande and N. Mani. An Interpretable Approach to Hateful Meme Detection. In *Proceedings of the 2021 International Conference on Multimodal Interaction*, pages 723–727, Montréal QC Canada, Oct. 2021. ACM.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep

doi:10.6342/NTU202501843

Bidirectional Transformers for Language Understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[6] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Roziere, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, C. Marra, C. McConnell, C. Keller, C. Touret, C. Wu, C. Wong, C. C. Ferrer, C. Nikolaidis, D. Allonsius, D. Song, D. Pintz, D. Livshits, D. Wyatt, D. Esiobu, D. Choudhary, D. Mahajan, D. Garcia-Olano, D. Perino, D. Hupkes, E. Lakomkin, E. AlBadawy, E. Lobanova, E. Dinan, E. M. Smith, F. Radenovic, F. Guzmán, F. Zhang, G. Synnaeve, G. Lee, G. L. Anderson, G. Thattai, G. Nail, G. Mialon, G. Pang, G. Cucurell, H. Nguyen, H. Korevaar, H. Xu, H. Touvron, I. Zarov, I. A. Ibarra, I. Kloumann, I. Misra, I. Evtimov, J. Zhang, J. Copet, J. Lee, J. Geffert, J. Vranes, J. Park, J. Mahadeokar, J. Shah, J. van der Linde, J. Billock, J. Hong, J. Lee, J. Fu, J. Chi, J. Huang, J. Liu, J. Wang, J. Yu, J. Bitton, J. Spisak, J. Park, J. Rocca, J. Johnstun, J. Saxe, J. Jia, K. V. Alwala, K. Prasad, K. Upasani, K. Plawiak, K. Li, K. Heafield, K. Stone, K. El-Arini, K. Iyer, K. Malik, K. Chiu, K. Bhalla, K. Lakhotia, L. Rantala-Yeary, L. van der Maaten, L. Chen, L. Tan, L. Jenkins, L. Martin, L. Madaan, L. Malo, L. Blecher, L. Landzaat, L. de Oliveira, M. Muzzi, M. Pasupuleti, M. Singh, M. Paluri, M. Kardas, M. Tsimpoukelli, M. Oldham, M. Rita, M. Pavlova, M. Kambadur, M. Lewis,

doi:10.6342/NTU202501843

M. Si, M. K. Singh, M. Hassan, N. Goyal, N. Torabi, N. Bashlykov, N. Bogoychev, N. Chatterji, N. Zhang, O. Duchenne, O. Çelebi, P. Alrassy, P. Zhang, P. Li, P. Vasic, P. Weng, P. Bhargava, P. Dubal, P. Krishnan, P. S. Koura, P. Xu, Q. He, Q. Dong, R. Srinivasan, R. Ganapathy, R. Calderer, R. S. Cabral, R. Stojnic, R. Raileanu, R. Maheswari, R. Girdhar, R. Patel, R. Sauvestre, R. Polidoro, R. Sumbaly, R. Taylor, R. Silva, R. Hou, R. Wang, S. Hosseini, S. Chennabasappa, S. Singh, S. Bell, S. S. Kim, S. Edunov, S. Nie, S. Narang, S. Raparthy, S. Shen, S. Wan, S. Bhosale, S. Zhang, S. Vandenhende, S. Batra, S. Whitman, S. Sootla, S. Collot, S. Gururangan, S. Borodinsky, T. Herman, T. Fowler, T. Sheasha, T. Georgiou, T. Scialom, T. Speckbacher, T. Mihaylov, T. Xiao, U. Karn, V. Goswami, V. Gupta, V. Ramanathan, V. Kerkez, V. Gonguet, V. Do, V. Vogeti, V. Albiero, V. Petrovic, W. Chu, W. Xiong, W. Fu, W. Meers, X. Martinet, X. Wang, X. Wang, X. E. Tan, X. Xia, X. Xie, X. Jia, X. Wang, Y. Goldschlag, Y. Gaur, Y. Babaei, Y. Wen, Y. Song, Y. Zhang, Y. Li, Y. Mao, Z. D. Coudert, Z. Yan, Z. Chen, Z. Papakipos, A. Singh, A. Srivastava, A. Jain, A. Kelsey, A. Shajnfeld, A. Gangidi, A. Victoria, A. Goldstand, A. Menon, A. Sharma, A. Boesenberg, A. Baevski, A. Feinstein, A. Kallet, A. Sangani, A. Teo, A. Yunus, A. Lupu, A. Alvarado, A. Caples, A. Gu, A. Ho, A. Poulton, A. Ryan, A. Ramchandani, A. Dong, A. Franco, A. Goyal, A. Saraf, A. Chowdhury, A. Gabriel, A. Bharambe, A. Eisenman, A. Yazdan, B. James, B. Maurer, B. Leonhardi, B. Huang, B. Loyd, B. De Paola, B. Paranjape, B. Liu, B. Wu, B. Ni, B. Hancock, B. Wasti, B. Spence, B. Stojkovic, B. Gamido, B. Montalvo, C. Parker, C. Burton, C. Mejia, C. Liu, C. Wang, C. Kim, C. Zhou, C. Hu, C.-H. Chu, C. Cai, C. Tindal, C. Feichtenhofer, C. Gao, D. Civin, D. Beaty, D. Kreymer, D. Li, D. Adkins, D. Xu, D. Testuggine,

doi:10.6342/NTU202501843

D. David, D. Parikh, D. Liskovich, D. Foss, D. Wang, D. Le, D. Holland, E. Dowling, E. Jamil, E. Montgomery, E. Presani, E. Hahn, E. Wood, E.-T. Le, E. Brinkman, E. Arcaute, E. Dunbar, E. Smothers, F. Sun, F. Kreuk, F. Tian, F. Kokkinos, F. Ozgenel, F. Caggioni, F. Kanayet, F. Seide, G. M. Florez, G. Schwarz, G. Badeer, G. Swee, G. Halpern, G. Herman, G. Sizov, Guangyi, Zhang, G. Lakshminarayanan, H. Inan, H. Shojanazeri, H. Zou, H. Wang, H. Zha, H. Habeeb, H. Rudolph, H. Suk, H. Aspegren, H. Goldman, H. Zhan, I. Damlaj, I. Molybog, I. Tufanov, I. Leontiadis, I.-E. Veliche, I. Gat, J. Weissman, J. Geboski, J. Kohli, J. Lam, J. Asher, J.-B. Gaya, J. Marcus, J. Tang, J. Chan, J. Zhen, J. Reizenstein, J. Teboul, J. Zhong, J. Jin, J. Yang, J. Cummings, J. Carvill, J. Shepard, J. McPhie, J. Torres, J. Ginsburg, J. Wang, K. Wu, K. H. U, K. Saxena, K. Khandelwal, K. Zand, K. Matosich, K. Veeraraghavan, K. Michelena, K. Li, K. Jagadeesh, K. Huang, K. Chawla, K. Huang, L. Chen, L. Garg, L. A, L. Silva, L. Bell, L. Zhang, L. Guo, L. Yu, L. Moshkovich, L. Wehrstedt, M. Khabsa, M. Avalani, M. Bhatt, M. Mankus, M. Hasson, M. Lennie, M. Reso, M. Groshev, M. Naumov, M. Lathi, M. Keneally, M. Liu, M. L. Seltzer, M. Valko, M. Restrepo, M. Patel, M. Vyatskov, M. Samvelyan, M. Clark, M. Macey, M. Wang, M. J. Hermoso, M. Metanat, M. Rastegari, M. Bansal, N. Santhanam, N. Parks, N. White, N. Bawa, N. Singhal, N. Egebo, N. Usunier, N. Mehta, N. P. Laptev, N. Dong, N. Cheng, O. Chernoguz, O. Hart, O. Salpekar, O. Kalinli, P. Kent, P. Parekh, P. Saab, P. Balaji, P. Rittner, P. Bontrager, P. Roux, P. Dollar, P. Zvyagina, P. Ratanchandani, P. Yuvraj, Q. Liang, R. Alao, R. Rodriguez, R. Ayub, R. Murthy, R. Nayani, R. Mitra, R. Parthasarathy, R. Li, R. Hogan, R. Battey, R. Wang, R. Howes, R. Rinott, S. Mehta, S. Siby, S. J. Bondu, S. Datta, S. Chugh, S. Hunt, S. Dhillon, S. Sidorov, S. Pan, S. Mahajan, S. Verma, S. Yamamoto, S. Ra-

doi:10.6342/NTU202501843

maswamy, S. Lindsay, S. Feng, S. Lin, S. C. Zha, S. Patil, S. Shankar, S. Zhang, S. Wang, S. Agarwal, S. Sajuyigbe, S. Chintala, S. Max, S. Chen, S. Kehoe, S. Satterfield, S. Govindaprasad, S. Gupta, S. Deng, S. Cho, S. Virk, S. Subramanian, S. Choudhury, S. Goldman, T. Remez, T. Glaser, T. Best, T. Koehler, T. Robinson, T. Li, T. Zhang, T. Matthews, T. Chou, T. Shaked, V. Vontimitta, V. Ajayi, V. Montanez, V. Mohan, V. S. Kumar, V. Mangla, V. Ionescu, V. Poenaru, V. T. Mihailescu, V. Ivanov, W. Li, W. Wang, W. Jiang, W. Bouaziz, W. Constable, X. Tang, X. Wu, X. Wang, X. Wu, X. Gao, Y. Kleinman, Y. Chen, Y. Hu, Y. Jia, Y. Qi, Y. Li, Y. Zhang, Y. Zhang, Y. Adi, Y. Nam, Yu, Wang, Y. Zhao, Y. Hao, Y. Qian, Y. Li, Y. He, Z. Rait, Z. DeVito, Z. Rosnbrick, Z. Wen, Z. Yang, Z. Zhao, and Z. Ma. The Llama 3 Herd of Models, 2024. Version Number: 3.

[7] M. S. Hee, R. K.-W. Lee, and W.-H. Chong. On Explaining Multimodal Hateful Meme Detection Models. In *Proceedings of the ACM Web Conference 2022*, pages 3651–3655, Virtual Event, Lyon France, Apr. 2022. ACM.

[8] A. Hu and S. Flaxman. Multimodal Sentiment Analysis To Explore the Structure of Emotions. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 350–358, London United Kingdom, July 2018. ACM.

[9] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[10] J. Huang and K. C.-C. Chang. Towards Reasoning in Large Language Models: A Survey. In *Findings of the Association for Computational Linguistics: ACL 2023*,

doi:10.6342/NTU202501843

pages 1049–1065, Toronto, Canada, 2023. Association for Computational Linguistics.

[11] N. J. Jackson. Understanding memetic media and collective identity through streamer persona on Twitch.TV. *Persona Studies*, 6(2):69–87, Oct. 2020. Place: Burwood, VIC, Australia Publisher: Deakin University - School of Communication and Creative Arts.

[12] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438, Dec. 2020.

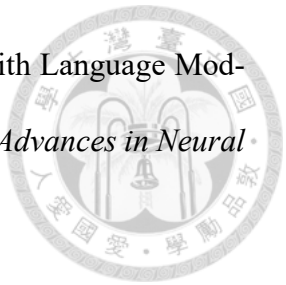
[13] D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, and D. Testuggine. The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2611–2624. Curran Associates, Inc., 2020.

[14] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa. Large Language Models are Zero-Shot Reasoners. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc., 2022.

[15] R. K.-W. Lee, R. Cao, Z. Fan, J. Jiang, and W.-H. Chong. Disentangling Hate in Online Memes. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5138–5147, Virtual Event China, Oct. 2021. ACM.

[16] A. Lewkowycz, A. J. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo, Y. Wu, B. Neyshabur, G. Gur-

doi:10.6342/NTU202501843



Ari, and V. Misra. Solving Quantitative Reasoning Problems with Language Models. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[17] C. Liu, G. Geigle, R. Krebs, and I. Gurevych. FigMemes: A Dataset for Figurative Language Identification in Politically-Opinionated Memes. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7069–7086, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics.

[18] H. Liu, C. Li, Y. Li, B. Li, Y. Zhang, S. Shen, and Y. J. Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge, Jan. 2024.

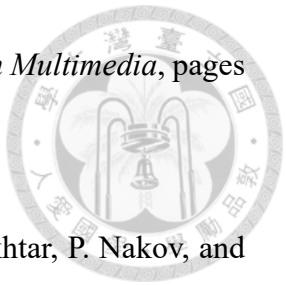
[19] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019. Version Number: 1.

[20] H. T. Ng, S. M. Wu, T. Briscoe, C. Hadiwinoto, R. H. Susanto, and C. Bryant. The CoNLL-2014 Shared Task on Grammatical Error Correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland, 2014. Association for Computational Linguistics.

[21] X. Pi, Q. Liu, B. Chen, M. Ziyadi, Z. Lin, Q. Fu, Y. Gao, J.-G. Lou, and W. Chen. Reasoning Like Program Executors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 761–779, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics.

[22] N. Prakash, H. Wang, N. K. Hoang, M. S. Hee, and R. K.-W. Lee. PromptMTopic: Unsupervised Multimodal Topic Modeling of Memes using Large Language Mod-

doi:10.6342/NTU202501843



els. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 621–631, Ottawa ON Canada, Oct. 2023. ACM.

[23] S. Pramanick, D. Dimitrov, R. Mukherjee, S. Sharma, M. S. Akhtar, P. Nakov, and T. Chakraborty. Detecting Harmful Memes and Their Targets. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2783–2796, Online, 2021. Association for Computational Linguistics.

[24] R. R. Pranesh and A. Shekhar. MemeSem:A Multi-modal Framework for Sentimental Analysis of Meme via Transfer Learning. In *4th Lifelong Machine Learning Workshop at ICML 2020*, 2020.

[25] M. R. Qorib and H. T. Ng. System Combination via Quality Estimation for Grammatical Error Correction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12746–12759, Singapore, 2023. Association for Computational Linguistics.

[26] N. Reimers and I. Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3980–3990, Hong Kong, China, 2019. Association for Computational Linguistics.

[27] C. Sharma, D. Bhageria, W. Scott, S. Pykl, A. Das, T. Chakraborty, V. Pulabaigari, and B. Gambäck. SemEval-2020 Task 8: Memotion Analysis- the Visuo-Lingual Metaphor! In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 759–773, Barcelona (online), 2020. International Committee for Computational Linguistics.

doi:10.6342/NTU202501843

[28] S. Sharma, S. Agarwal, T. Suresh, P. Nakov, M. S. Akhtar, and T. Chakraborty. What Do You MEME? Generating Explanations for Visual Semantic Role Labelling in Memes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8):9763–9771, June 2023.

[29] T. Shin, Y. Razeghi, R. L. Logan Iv, E. Wallace, and S. Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online, 2020. Association for Computational Linguistics.

[30] X. Sun, X. Li, J. Li, F. Wu, S. Guo, T. Zhang, and G. Wang. Text Classification via Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8990–9005, Singapore, 2023. Association for Computational Linguistics.

[31] S. Suryawanshi, B. R. Chakravarthi, M. Arcan, and P. Buitelaar. Multimodal Meme Dataset (MultiOFF) for Identifying Offensive Content in Image and Text. In R. Kumar, A. K. Ojha, B. Lahiri, M. Zampieri, S. Malmasi, V. Murdock, and D. Kadar, editors, *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 32–41, Marseille, France, May 2020. European Language Resources Association (ELRA).

[32] R. Taylor, M. Kardas, G. Cucurull, T. Scialom, A. Hartshorn, E. Saravia, A. Poulton, V. Kerkez, and R. Stojnic. Galactica: A Large Language Model for Science, 2022. Version Number: 1.

[33] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, and

doi:10.6342/NTU202501843

D. Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc., 2022.

[34] B. Xu, T. Li, J. Zheng, M. Naseriparsa, Z. Zhao, H. Lin, and F. Xia. MET-Meme: A Multimodal Meme Dataset Rich in Metaphors. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2887–2899, Madrid Spain, July 2022. ACM.

[35] P. Yu, T. Wang, O. Golovneva, B. AlKhamissi, S. Verma, Z. Jin, G. Ghosh, M. Diab, and A. Celikyilmaz. ALERT: Adapt Language Models to Reasoning Tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1055–1081, Toronto, Canada, 2023. Association for Computational Linguistics.

[36] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer. OPT: Open Pre-trained Transformer Language Models, 2022. Version Number: 4.

[37] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen. A Survey of Large Language Models, 2023. Version Number: 16.

doi:10.6342/NTU202501843