A Template Is All You Meme

Luke Bates,[1] **Peter Ebert Christensen,**[2,3] **Preslav Nakov,**[4] **and Iryna Gurevych**[1]
[1]Ubiquitous Knowledge Processing Lab (UKP Lab)
Department of Computer Science and Hessian Center for AI (hessian.AI)
Technical University of Darmstadt
[2]Department of Computer Science, University of Copenhagen, [3]Pioneer Centre for AI
[4]Mohamed bin Zayed University of Artificial Intelligence
https://www.ukp.tu-darmstadt.de/

Abstract

Templatic memes, characterized by a semantic structure adaptable to the creator's intent, represent a significant yet underexplored area within meme processing literature. With the goal of establishing a new direction for computational meme analysis, here we create a knowledge base composed of more than 5,200 meme templates, information about them, and 54,000 examples of template instances (templatic memes). To investigate the semantic signal of meme templates, we show that we can match memes in datasets to base templates contained in our knowledge base with a distance-based lookup. To demonstrate the power of meme templates, we create TSplit, a method to reorganize datasets, where a template or templatic instance can only appear in either the training or test split. Our re-split datasets enhance general meme knowledge and improve sample efficiency, leading to more robust models. Our examination of meme templates results in state-of-the-art performance for every dataset we consider, paving the way for analysis grounded in *templateness*.[1]

WARNING: For demonstration purposes, we show memes that some may find offensive.

1 Introduction

Memes are a form of communication capable of succinctly conveying complicated messages. They have been defined as a unit of cultural transmission or of imitation and replication (Dawkins, 1976), but all memes possess the trait of referencing a cultural moment shared by a group of people. Despite their basis in Internet culture, they exhibit sociolinguistic traits of in-group communication (Styler, 2020; Holm, 2021). A meme's meaning can therefore be opaque to those who do not belong to the in-group.

Meme templates are common patterns or elements, such as text or images, that are used to create novel memes. They can be difficult to parse because they can be combined in different ways and each one has its own unique meaning, the specific semantics of which is customizable by the person posting the meme (the *poster*). The template and its message can be referenced by an image, but may not be directly related to that image. If the viewer is not familiar with the template in question, they may not understand the meme's meaning. For example, in Figure 1, we see three instances of the popular *Is This a Pigeon?*[2] template. This template conveys the idea that the subject of the person is misinterpreting the object of the butterfly due to his own worldview or limited knowledge. The meaning can be tuned by the poster using overlaid text or images. Importantly, such altered images are considered instances of the same template. To interpret memes, one must recognize the entities in the meme and the template the meme uses, if any.

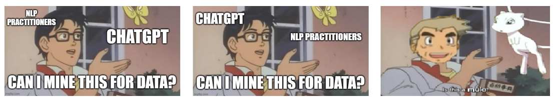
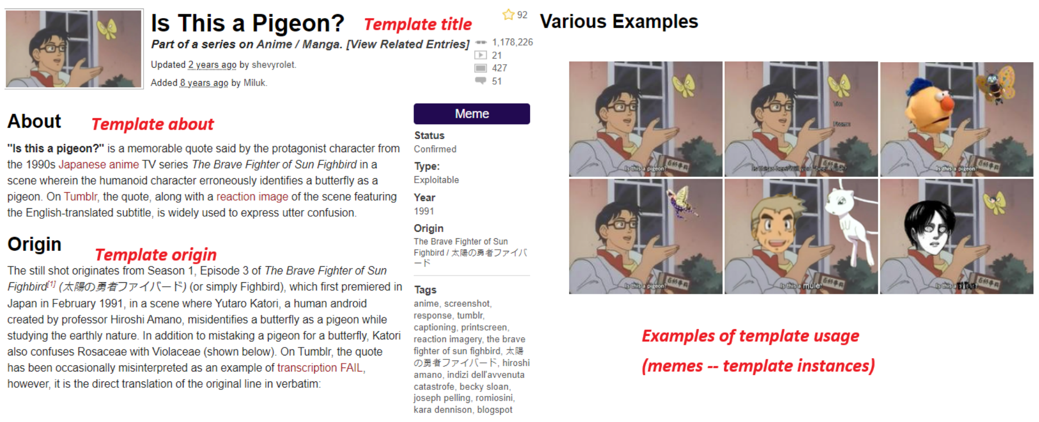


Figure 1: The meaning of templatic memes is customizable via overlaid text or image(s), but remains grounded in the context of the template. The first panel suggests that the NLP community thinks it can use ChatGPT to generate data, while the second one suggests that ChatGPT can exploit the NLP community for data. The third one uses overlaid images to reference Pokemon.

It is important to distinguish between templatic memes and other meme types. Templatic memes reference a meme template, which is a commonly reused material (e.g., text, images, audio), to create a novel instance that is still grounded in the meme template's semantics. Non-templatic memes can be (visual) puns, jokes, or emphasis that might not directly reference a meme template and can be understood without knowledge about meme templates

[1]Our code is available at https://github.com/UKPLab/naacl2025-a-template-is-all-you-meme.

[2]https://knowyourmeme.com/memes/is-this-a-pigeon



Is This a Pigeon?

Part of a series on Anime / Manga. [View Related Entries]

Updated 2 years ago by shevyrolet.

Added 8 years ago by Miluk.

92
1,178,226
21
427
51

Various Examples

Meme

Status
Confirmed

Type:
Exploitable

Year
1991

Origin
The Brave Fighter of Sun Fighbird / 太陽の勇者ファイバード

Tags
anime, screenshot, response, tumblr, captioning, printscreen, reaction imagery, the brave fighter of sun fighbird, 太陽の勇者ファイバード, hiroshi amano, indizi dell'avvenuta catastrofe, becky sloan, joseph pelling, romiosini, kara dennison, blogspot

About

"Is this a pigeon?" is a memorable quote said by the protagonist character from the 1990s Japanese anime TV series *The Brave Fighter of Sun Fighbird* in a scene wherein the humanoid character erroneously identifies a butterfly as a pigeon. On Tumblr, the quote, along with a reaction image of the scene featuring the English-translated subtitle, is widely used to express utter confusion.

Origin

The still shot originates from Season 1, Episode 3 of *The Brave Fighter of Sun Fighbird*[1] *(太陽の勇者ファイバード)* (or simply Fighbird), which first premiered in Japan in February 1991, in a scene where Yutaro Katori, a human android created by professor Hiroshi Amano, misidentifies a butterfly as a pigeon while studying the earthly nature. In addition to mistaking a pigeon for a butterfly, Katori also confuses Rosaceae with Violaceae (shown below). On Tumblr, the quote has been occasionally misinterpreted as an example of transcription FAIL, however, it is the direct translation of the original line in verbatim:

Figure 2: Example entry from KYM where we have labeled relevant data fields in red.

or even memes (see Section A.9).

Know Your Meme (KYM),[3] the Internet Meme Database, is a valuable resource for information related to memes, and especially, to templatic memes. Even people familiar with memes may not be aware of a specific template and can look it up on KYM. The meme entries provide the base template and information about it, such as its meaning, origin, examples, etc. By reviewing entries of unfamiliar templates, users learn how to interpret and use the template themselves to create novel instances for their specific communication needs.

Memes are of interest to the machine learning community (Aggarwal et al., 2023) because they are difficult, but can still be formulated as classification or generation tasks (Peirson and Tolunay, 2018). They are also used to spread misinformation and hate speech (Pramanick et al., 2021a). Memes usually express concepts humorously, and humor has been shown to increase the persuasiveness of an idea (Walter et al., 2018). Thus, developing systems that can understand memes is important to prevent the spread of harmful content. However, AI researchers and datasets treat memes as static images with text (Du et al., 2020; Qu et al., 2022) despite memes having their own unique characteristics, such as being templatic or not.

Here, we analyze memes as memes, rather than as images, text, or a combination of both. To support this, we introduce the Know Your Meme Knowledge Base (KYMKB), a database of meme templates and associated information sourced from KYM. We hypothesize that knowledge of meme templates provides valuable context for understanding memes. Using a distance-based lookup, we match templatic instances to base templates in the KYMKB, enabling retrieval of meme-specific context. This context is useful for prompting large language models (LLMs) or multimodal Vision-LMs in a retrieval-augmented generation (RAG) framework, as demonstrated in Section A.14.

To investigate the effect meme templates have on modelling, we create a method for (re)splitting meme datasets such that a template and its instances can only appear in either the training or test split, which we call the Template-Aware Splitter (TSplit). TSplit (re)organizes dataset entries based on their distance in embedding space from the KYMKB. We find that controlling for *template awareness* makes TSplit more sample-efficient than standard fine-tuning or random sampling. We also show that TSplit has strong regularization effects in scenarios where models are likely to overfit.

Our analysis of meme templates results in state-of-the-art (SOTA) performance for all datasets we consider, establishing a robust foundation for future research grounded in *templateness*. Our contributions are as follows:

1. We suggest a new direction for future meme analysis grounded in templates.

2. We construct a knowledge base of meme templates and information about them.

3. We show meme templates are useful for a number of tasks and can have marked effects in fine-tuning experiments.

2 Related Work

There has been a lot of work on analyzing memes in various task formulations.

Memes as harmful content This includes MultiOFF (Suryawanshi et al., 2020), a dataset

[3]https://knowyourmeme.com/

of offensive memes related to the 2016 US presidential election. The MAMI dataset (Fersini et al., 2022) is composed of two datasets from SemEval-2022: in subtask A, the goal is to identify misogyny in memes, while in subtask B, it is to determine different types of misogyny expressed by a meme. Lin et al. (2023) recognized that the surface-level text and the image of memes are insufficient and employed large language model (LLM) knowledge distillation to classify dangerous memes.

Memes as a form of language Dimitrov et al. (2021) pointed out that memes can be persuasive by exploiting more than 20 different propaganda techniques. FigMemes (Liu et al., 2022) scraped images from a politically incorrect and infamously toxic board on 4chan, /pol/,[4] and labeled over five thousand memes with six different types of figurative language used in the meme, recognizing that memes are capable of expressing abstract and complicated messages. Mishra et al. (2023) released Memotion 3, which is composed of memes in Hindi and English, labeled for sentiment, emotion detection, and emotion intensity. Hwang and Shwartz (2023) released a dataset of meme explanations to aid in resolving metaphors in memes. Zhou et al. (2024) showed that memes and meme templates can be clustered by their semantic function.

Context for memes All the above work fine-tuned multimodal PLMs or prompted LLMs on their respective datasets, but did not use additional context in order to increase meme understanding. This is a trend in meme-related ML research. One exception is MEMEX (Sharma et al., 2023). They use Wikipedia and Quora to assemble explanations to ask if an explanation document is relevant for a meme, formulating a novel task and multimodal model. Notably, this work uses meme-external information (Wikipedia/Quora), but not meme knowledge, e.g., information about the template used by the meme. We emphasize that the context they inject is common knowledge or knowledge about named entities, not knowledge about memes.

General meme resources Tommasini et al. (2023) developed a knowledge graph of memes by scraping and querying different sources of information, such as KYM, to connect memes to the information they reference. However, they do not leverage the graph in a downstream task, nor is it clear how their graph could be applied to meme analysis due to a lack of demonstrations.

[4]https://boards.4chan.org/pol/

The current work Our work focuses on the meme-specific phenomenon of meme templates. To this end, we construct a knowledge base from KYM to ground our analysis in a meme-specific context. The KYMKB is not a dataset and is not labeled for a specific task, but contains information about templatic memes, such as their title, meaning, and origin. We show that we can use CLIP (Radford et al., 2021) and distanced-based lookup algorithms "off the shelf" and match templates to memes in existing datasets. To investigate the effect meme templates have on meme processing, we develop TSplit for (re)organizing datasets based on *templateness*, the Euclidean distance of meme-vector representations from the KYMKB. This has significant effects in fine-tuning experiments, establishing SOTA performance in multiple setups.

3 The Know Your Meme Knowledge Base

Know Your Meme, the "Internet Meme Database", can be thought of as the Wikipedia for memes. Users create web pages of a meme template and document information about it, e.g., its meaning, and add examples of its usage (see Figure 2). The community reviews and approves entries, updating them as the template's usage evolves.

Template instances are important for meme understanding. In Figure 2, we see that the template can be altered via overlaid text and images to tune it for a specific communication goal. Existing approaches rely on OCR to extract the text and/or the named entities (Kougia et al., 2023), but this would not work in many cases, e.g., if the entities are images referencing a popular YouTube video.[5]

KYM is a valuable resource that has been underutilized by the AI community. To address this, we create the KYMKB, a collection of meme templates, examples, and information about the meme's usage. To ensure the quality of the entries, we crawl templates from KYM that are approved by the community, scraping 5,220 base templates and 49,531 examples (see Section A.9).

Memes may deviate from their template-based origin, and the KYMKB accounts for this. Consider our running example of *Is This a Pigeon?*. This template was popularized in 2011, but it then had a resurgence in April of 2018. By June 2018, a female version of the template had emerged, which

[5]https://www.youtube.com/watch?v=sXOdn6vLCuU&t=8s

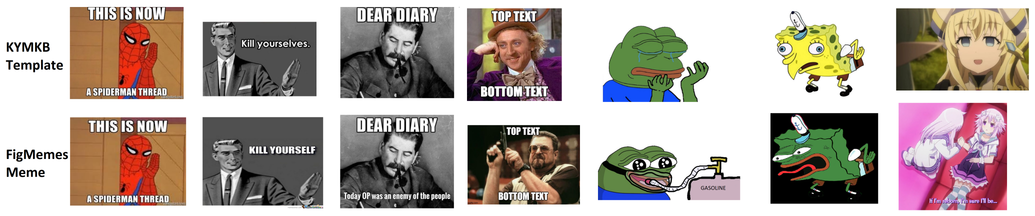


Figure 3: KYMKB templates (first row) vs. their nearest neighbor in the FigMemes dataset (second row).

was interpreted by some users as an example of gender transitioning. Such usage and evolution is documented by KYM users in the form of both text and images. Popular template instances that differ from their origin often become their own template, such as *Pepe the Frog*[6] vs. *Feels Bad Man/Sad Frog*.[7] The former is a template originally used in a manner similar to emoticons, while the latter is a popular instance of Pepe that became its own template expressing sorrow or disappointment. By collecting examples, user-curated information, and distinct but related templates, the KYMKB is organized for the dynamic nature of memes.

4 Template-Meme Analysis

To investigate the power of templates, we conducted exploratory data analysis on four meme datasets, MultiOff, Memotion 3, FigMemes, and MAMI. Specifically, we encoded the KYMKB templates, fit a nearest neighbor lookup, as this is an intuitive and commonly used vector-similarity measure (Buitinck et al., 2013), query the 500 closest neighbors in the datasets, and manually inspect the similarities between template-meme pairs. Henceforth, we use CLIP as our encoder as it is a commonly-used PLM for vision and language learning problems and memes (Pramanick et al., 2021b), but the encoding function is ultimately arbitrary and we refer to it as f. In our analysis, we group memes into four categories, detailed below:

1. **Base** - the meme is the base template or cropped/distorted version of it.
2. **Instance** - the meme is a templatic instance customized with text or overlaid images.
3. **Relevant** - the meme employs more than one template or an obvious reference to a template.
4. **Irrelevant** - the meme/image does not appear related to its template pair.

Dataset	Base	Instance	Relevant	Irrelevant
MultiOff	10.4	30	21.3	38.3
Memotion 3	13.4	48.2	11.6	26.8
FigMemes	39.2	15.2	28.8	16.8
MAMI	5.7	24.1	20.3	49.9

Table 1: Percentages of memes grouped by visual similarities to their nearest neighbor in the KYMKB.

Figure 3 shows a sample of our template-meme pairs and Table 1 summarizes our groupings in percentages. In many cases, the memes we find are our base templates or distorted versions of them, such as the first two columns in the figure. We observed many instances of templates tuned by the poster, such as the third column. To validate our grouping, we sampled 50 template-meme pairs and asked two of our colleagues to annotate them based on our four categories. For the *Base*, *Instance*, *Relevant*, and *Irrelevant* groupings, we computed a Cohen's Kappa of 1.0, 0.45, -0.11, and 0.80 respectively, with the average being 0.54, moderate agreement. It is reasonable that our annotators could be confused by the *Instance* and *Relevant* grouping as this distinction is subjective, however, we note the perfect agreement and near perfect agreement we captured for the *Base* and *Irrelevant* grouping.
We are able to match templates to relevant instances despite different appearances. For example, the KYMKB includes *Pepe the Frog*, a template with many different versions, which is also a symbol of the alt-right movement (Glitsos and Hall, 2019). When we query FigMemes, we capture an instance of a happy Pepe inhaling gasoline, communicating the idea that only death can bring the poster happiness. Going a step further, we see two templatic concepts merging into a single meme. *Mocking SpongeBob*[8] is a popular template, which is used to express contempt. The nearest neighbor to this template in FigMemes is an instance where SpongeBob has been amalgamated with the angry Pepe. Querying the KYMKB retrieves enough informa-

[6]https://knowyourmeme.com/memes/pepe-the-frog

[7]https://knowyourmeme.com/memes/feels-bad-man-sad-frog

[8]https://knowyourmeme.com/memes/mocking-spongebob

Dataset	Original Training Size	TSplit Training Size	Original Validation Size	TSplit Validation Size	Original Test Size	TSplit Dummy Test Size
MultiOff	445	381	149	96	149	117
Memotion 3	5600	4674	1400	1169	1500	1157
FigMemes	3084	2333	515	260	1542	1006
MAMI	8000	7353	2000	1839	1000	808

Table 2: Original dataset split sizes and example TSplit dataset splits derived from training and validation data. The dummy test data was discarded. `ViT-L/14@336px` was used as the encoder. See Table 7 for other encoders.

tion from the *about* sections to interpret this meme as the alt-right angrily expressing derision, consistent with /pol/ (Hine et al., 2017), the domain from which FigMemes was created.

KYMKB may match a meme or image that is not a template instance. The 7th column shows the template of *You Get Used To It*[9] matched to a picture that appears to be a still from another anime.[10] The FigMemes image is not an instance of the aforementioned template, but this is subjective as it is not possible to know every template nor does the KYMKB encompass all meme knowledge. We note that for MAMI this especially seems to be the case, where for nearly half of the examples we considered, our template-meme pairs did not appear to be related (see Table 1). Not all memes reference a template and meme datasets contain non-meme images (see Figure 6).

5 Template-Aware Splitter

Given that we are able to successfully match memes to templates in the KYMKB, we hypothesize that many meme datasets are composed of nothing more than templatic instances, which may already be contained within our knowledge base. This would then create a strong signal we could leverage in a downstream task.

To investigate this, we demonstrate another use case and develop a tool from the KYMKB called the Template-Aware Splitter (TSplit), which quantifies *templateness* by comparing base templates to their examples (see Figure 2). TSplit embeds a template ($ref = f(KYMKB_{[i]})$) and its examples ($examp = f(KYMKB_{[i][j]})$), then computes the Euclidean distance between them ($dists_{[i]} = dist(ref, examp_{[j]})$). These distances are used to compute a threshold value, such as the median distance ($threshold_{[i]} = median(dists_{[i]})$). We explore four threshold methods: maximum, median, mean, and 25th percentile distances. Using the maximum provides a lenient definition of a template instance, allowing greater visual variance (see Figure 3), while the 25th percentile offers a stricter interpretation. TSplit encodes a meme, matches it to the closest template in the KYMKB, and compares the distance to the template's threshold. If the distance exceeds the threshold, the meme is assigned a unique identifier (UI); otherwise, it is recorded as an instance of the template. While our focus is on templatic memes, TSplit also handles non-templatic memes via UIs.

We use TSplit to reorganize datasets and assign memes to a template or declare them non-templatic. TSplit samples our templates and UIs without replacement, such that a template or UI cannot appear in both the training and the test data. Our goal is to decouple the data distribution from template robustness, inspired by work done in QA, NLI, and bias detection on adversarial dataset creation and modelling (Jia and Liang, 2017; Gururangan et al., 2018; Baly et al., 2020). Note that TSplit is task agnostic (see Section A.2).

To examine the effect TSplit has on dataset sampling, we tested various versions of TSplit on six meme classification datasets: FigMemes with seven labels of different types of figurative language, MultiOff a binary dataset where the labels can be offensive or non-offensive, Memotion 3, where Task A has three labels for sentiment analysis and Task B has four for different types of emotion, and MAMI with Task A being binary misogyny detection and Task B having four labels with different types of misogyny being expressed (see Tables 18 and 19). We choose these datasets because they were made to address the issue of harmful meme detection and to frame our analysis of meme templates in the light of this important goal. Each dataset used Google Vision to provide OCR text overlaid on memes.[11]

To avoid overlapping templates and unique identifiers between dataset splits, we construct an array of distinct objects, meaning detected templates or

[9] https://knowyourmeme.com/memes/you-get-used-to-it

[10] https://en.wikipedia.org/wiki/Hyperdimension_Neptunia

[11] https://cloud.google.com/use-cases/ocr

unique identifiers. We randomly shuffle the array and then create a test split index. Everything that appears before the index is an object that can appear in the training data and everything after can appear in the test data. We use the following formula to create this index: $\lfloor (\frac{t_{size}}{d_{size}}) * o_{size} \rfloor$, where t_{size} is the number of test examples in the original dataset, d_{size} is the total size of the dataset, and o_{size} is the number of distinct detected objects (templates or unique identifiers). We maintain the original dataset ratios (for example training 60%, validation 20%, and test 20%), but because we sample the resplit datasets based on detected templates, it is difficult to maintain the exact numbers.

Finally, it is possible that a template in the KYMKB does not have any examples. In such cases, we use a global threshold value calculated by taking the maximum, median, mean, or 25^{th} percentile value of all template thresholds.

5.1 Down-sampling Experimental Setup

Here, we describe different experimental setups we investigated with TSplit. Our goal is to study the effect of meme templates on modelling, specifically on the classification datasets we have examined thus far. In these experiments we fine-tuned three different CLIP encoders. We opted for fine-tuning as prompting with (multimodal) LLMs does not result in meme understanding (Hwang and Shwartz, 2023), and this is supported by our own experiments (see Section A.14).

In this first suite of experiments, we downsampled the training and validation data by using TSplit to reorganize them but kept the original test split untouched and used it for evaluation. In other words, we constructed a training, validation, and dummy test split from the original training and validation split based on their size ratios relative to the original sizes. We then used the resampled training data for fine-tuning and the resampled validation data for model selection, and discarded the dummy test split, using the original test data for the final evaluation. Therefore, less data was used in this setting for model optimization compared to the original data splits (see Table 2).

We compare TSplit against two baselines: (1) CLIP fine-tuned on the original dataset splits (Original$_{ViT-X}$), using both OCR-extracted text and visual content, and (2) the same model with the training and validation data randomly downsampled to match TSplit's split sizes (Baseline$_{ViT-X}$; see Table 2). Both baselines include a feed-forward layer after the CLIP text and image encoders. We also fine-tune the same model on our resplit datasets. Training is performed for 20 epochs using AdamW (Loshchilov and Hutter, 2019) with a learning rate of $1e^{-5}$. Test evaluation uses the best-performing checkpoint on the validation split. For datasets without a validation split, we sample 20% of the training data. We experiment over five seeds, reporting the mean and standard deviation.[12]

Results and Discussion Table 3 summarizes our experimental results. While our baselines performed strongly, TSplit$_{max}$ with the largest encoder outperformed them, achieving an F1 score 66.16 for MultiOff in the initial grouping, SOTA performance. For Memotion 3 (tasks A and B), performance remained consistent across configurations. Prior research highlights random (down)sampling as a strong baseline (Ein-Dor et al., 2020; Lu et al., 2024), likely due to the "noise" it introduces, which can enhance robustness.

Fine-tuning on the original split also established the state-of-the-art performance for FigMemes, achieving an F1 score of 47.79. However, this was subsequently surpassed by TSplit$_{max}$ using the largest encoder, which achieved an F1 score of 48.85. Notably, this dataset exhibited the greatest variance in model performance, with random downsampling proving to be comparatively less effective. We attribute this disparity to the inherent difficulty of the task and the highly templatic nature of the dataset (see Section 4). The model benefits from either more data or template-based sampling, with TSplit demonstrating both sample efficiency and strong overall performance.

We observed that a variant of TSplit outperformed both of our baselines on four of the six datasets analyzed, while TSplit$_{max}$ from the first grouping emerges as the overall strongest performer. By accounting for *templateness*, the model acquires more generalized meme knowledge that transfers effectively across splits, achieving this in an efficient manner. Notably, the sizes of the TSplit dataset splits used for optimization are significantly smaller than the original dataset (see Table 2). Despite the reduction in data, this sampling approach matches or exceeds the performance of models trained on the full dataset. For FigMemes, as well as MAMI (tasks A and B), optimization was performed using nearly 1,000 fewer memes,

[12]We used 40 GB and 80 GB NVIDIA A100 Tensor Core GPUs.

Split	MultiOff	Memotion 3 (A)	Memotion 3 (B)	FigMemes	MAMI (A)	MAMI (B)	Overall
Encoder: `ViT-L/14@336px`							
Original$_{ViT-L/14@336px}$	$63.64_{2.31}$	$26.38_{1.57}$	$81.7_{2.33}$	$47.79_{1.43}$	$\mathbf{71.82}_{4.05}$	$56.6_{1.56}$	$57.99_{17.76}$
TSplit$_{max}$	$\mathbf{66.16}_{4.11}$	$26.68_{1.29}$	$83.98_{1.76}$	$47.76_{0.8}$	$68.72_{2.21}$	$\mathbf{58.69}_{1.56}$	$\mathbf{58.66}_{17.98}$
TSplit$_{median}$	$64.54_{3.0}$	$\underline{28.35}_{1.73}$	$82.93_{3.15}$	$\mathbf{48.85}_{1.31}$	$68.63_{1.25}$	$58.19_{2.13}$	$58.58_{17.02}$
TSplit$_{mean}$	$64.21_{3.87}$	$28.14_{1.21}$	$82.57_{1.49}$	$47.43_{1.52}$	$68.62_{2.42}$	$57.41_{1.88}$	$58.06_{17.12}$
TSplit$_{percentile}$	$62.04_{3.36}$	$26.79_{2.08}$	$83.24_{3.39}$	$44.63_{1.96}$	$68.87_{0.99}$	$58.12_{0.98}$	$57.28_{17.89}$
Baseline$_{ViT-L/14@336px}$	$65.03_{2.66}$	$26.36_{1.05}$	$\underline{84.31}_{1.5}$	$45.21_{2.52}$	$70.68_{2.86}$	$57.73_{1.05}$	$58.22_{18.56}$
Encoder: `ViT-B/32`							
Original$_{ViT-B/32}$	$60.34_{2.21}$	$\mathbf{28.46}_{1.67}$	$82.65_{1.04}$	$38.26_{1.59}$	$\underline{68.17}_{1.05}$	$54.0_{0.82}$	$\underline{55.31}_{18.03}$
TSplit$_{max}$	$\underline{60.4}_{2.17}$	$26.02_{0.72}$	$82.0_{2.44}$	$37.44_{1.3}$	$67.59_{1.54}$	$\underline{55.28}_{1.37}$	$54.79_{18.55}$
TSplit$_{median}$	$58.61_{3.08}$	$26.66_{0.72}$	$82.24_{1.89}$	$37.74_{2.21}$	$65.79_{2.08}$	$54.05_{1.8}$	$54.18_{18.12}$
TSplit$_{mean}$	$58.24_{2.17}$	$25.98_{0.94}$	$\mathbf{84.35}_{2.22}$	$38.98_{0.83}$	$65.61_{2.75}$	$54.61_{1.85}$	$54.63_{18.63}$
TSplit$_{percentile}$	$60.09_{2.15}$	$25.83_{0.84}$	$81.61_{2.05}$	$\underline{39.22}_{2.15}$	$67.16_{2.79}$	$54.49_{1.81}$	$54.73_{18.17}$
Baseline$_{ViT-B/32}$	$60.07_{3.29}$	$27.78_{1.38}$	$83.31_{0.48}$	$39.47_{1.45}$	$67.11_{1.77}$	$53.5_{1.38}$	$55.21_{18.06}$
Encoder: `ViT-B/16`							
Original$_{ViT-B/16}$	$\underline{65.82}_{2.69}$	$\underline{28.28}_{1.49}$	$82.72_{0.83}$	$\underline{42.98}_{1.88}$	$\underline{68.63}_{0.71}$	$55.03_{1.56}$	$\underline{57.24}_{17.79}$
TSplit$_{max}$	$62.87_{3.14}$	$26.7_{1.91}$	$\underline{83.7}_{1.64}$	$40.19_{2.6}$	$68.43_{1.53}$	$55.24_{1.62}$	$56.19_{18.61}$
TSplit$_{median}$	$60.46_{3.34}$	$27.04_{1.02}$	$82.17_{2.32}$	$40.07_{2.68}$	$67.77_{2.01}$	$\underline{56.54}_{1.36}$	$55.68_{17.96}$
TSplit$_{mean}$	$63.96_{3.24}$	$25.92_{0.73}$	$83.38_{1.93}$	$39.88_{1.21}$	$67.61_{1.9}$	$56.18_{0.75}$	$56.16_{18.76}$
TSplit$_{percentile}$	$61.88_{1.25}$	$26.13_{0.83}$	$82.62_{1.33}$	$41.3_{1.82}$	$67.6_{2.6}$	$54.84_{1.65}$	$55.73_{18.2}$
Baseline$_{ViT-B/16}$	$62.98_{1.52}$	$26.44_{0.25}$	$82.43_{1.87}$	$37.81_{2.01}$	$66.07_{2.68}$	$54.6_{0.76}$	$55.06_{18.48}$

Table 3: **Challenging setup**: *TSplit uses the original test sets for evaluation and much less data for both training and model selection (see Table 2). Templateness*-based down-sampling compared against fine-tuning on the original dataset splits and against randomly downsampling to match the TSplit split sizes. We group results by their encoder (*Original* subscript), where encoders are organized by size in descending order. The best performer in each group is underlined, while the best for each dataset is in **bold**. See Table 18 for evaluation measures. The *Overall* column is the mean and standard deviation of each row for insight into aggregated performance.

yet this reduction led to improved performance.

Controlling for *templateness* proves to be sample-efficient not only for datasets with a high proportion of templatic memes, such as FigMemes, but also for less templatic datasets like MAMI (see Section 4). TSplit$_{max}$, when paired with the largest encoder, performed well on MultiOff, FigMemes, and MAMI (B), establishing itself as the overall strongest performer. TSplit$_{max}$ implies a high threshold for the maximum distance between a template and its examples. This is intuitive, as templatic memes can vary greatly from their base form (Figure 3), and even non-templatic memes may implicitly reference a template (Figure 6).

TSplit leverages the size and strong base performance of the encoder. Among both the baselines and TSplit configurations, `ViT-L/14@336px`—the largest encoder—demonstrates the strongest overall performance. Notably, random downsampling with this encoder can outperform optimization on the full dataset, which is markedly not the case for the other CLIP models. TSplit utilizes this encoder not only for classification but also to reorganize splits and selectively discard data. The model's strong performance suggests a latent understanding of meme content, which TSplit naturally leverages.

5.1.1 Analysis: TSplitting the Entire Dataset

To better understand sampling with TSplit, we examine the strength of *templateness* by applying TSplit and fine-tuning across the entire dataset. In the previous section, we observed that we are able to down-sample the training and validation splits with TSplit, and still fine-tune competitive models evaluated on the same test data. Here, however, we resample everything, comparing this model against Original$_{ViT-X}$. Table 5 shows an example of the TSplit split sizes across the entire dataset. Our fine-tuning procedure is the same.

Results and Discussion Table 4 shows our results. The TSplit versions of MultiOff appear more challenging than the original. Looking into the original test split revealed that it contained fewer templatic memes, whereas the TSplit test split included more. This suggests that templatic structures are harder for the model to learn in smaller datasets and that the original split was less challenging.

For Memotion 3 (A), a difficult dataset with many templatic memes, previous work required the use of a "Hinglish" BERT-based model to reach an F1 of 33.28 in Mishra et al. (2023). If we attempt to decouple the distribution from templates, we get multilingualism without even trying, attaining an F1 of 35.3 with TSplit$_{mean}$ from the first grouping.

Controlling for template awareness makes dif-

Split	MultiOff	Memotion 3 (A)	Memotion 3 (B)	FigMemes	MAMI (A)	MAMI (B)	Overall
			Encoder: `ViT-L/14@336px`				
Original$_{ViT-L/14@336px}$	$\underline{63.64}_{2.31}$	$26.38_{1.57}$	$\underline{81.7}_{2.33}$	$47.79_{1.43}$	$71.82_{4.05}$	$56.6_{1.56}$	$57.99_{17.76}$
TSplit$_{max}$	$59.32_{0.67}$	$35.0_{1.46}$	$80.61_{0.66}$	$\mathbf{50.83}_{1.37}$	$\mathbf{86.26}_{0.67}$	$\mathbf{61.18}_{3.28}$	$\mathbf{62.2}_{17.3}$
TSplit$_{median}$	$57.03_{2.88}$	$\mathbf{35.3}_{1.13}$	$80.43_{1.06}$	$50.4_{2.11}$	$82.89_{1.17}$	$58.52_{2.62}$	$60.76_{16.59}$
TSplit$_{mean}$	$60.8_{2.85}$	$35.27_{1.34}$	$80.99_{0.29}$	$50.22_{2.66}$	$85.63_{1.79}$	$59.67_{1.46}$	$62.1_{17.22}$
TSplit$_{percentile}$	$59.61_{2.68}$	$35.25_{1.13}$	$79.85_{0.91}$	$48.55_{1.51}$	$84.41_{1.35}$	$61.09_{2.48}$	$61.46_{16.94}$
			Encoder: `ViT-B/32`				
Original$_{ViT-B/32}$	$\underline{60.34}_{2.21}$	$28.46_{1.67}$	$\mathbf{82.65}_{1.04}$	$38.26_{1.59}$	$68.17_{1.05}$	$54.0_{0.82}$	$55.31_{18.03}$
TSplit$_{max}$	$58.53_{4.08}$	$\underline{32.13}_{2.07}$	$81.29_{0.89}$	$\underline{40.98}_{2.67}$	$82.47_{1.83}$	$58.36_{2.53}$	$\underline{58.96}_{18.69}$
TSplit$_{median}$	$58.22_{2.18}$	$31.86_{1.36}$	$81.05_{0.72}$	$36.71_{1.66}$	$\underline{83.47}_{2.14}$	$59.36_{3.37}$	$58.44_{19.65}$
TSplit$_{mean}$	$57.42_{4.11}$	$31.35_{1.92}$	$80.97_{0.55}$	$37.17_{2.31}$	$81.69_{2.96}$	$\underline{59.61}_{3.26}$	$58.04_{19.3}$
TSplit$_{percentile}$	$54.57_{6.21}$	$30.91_{1.93}$	$79.84_{0.88}$	$39.24_{3.95}$	$83.09_{1.59}$	$56.45_{2.9}$	$57.35_{19.17}$
			Encoder: `ViT-B/16`				
Original$_{ViT-B/16}$	$\mathbf{65.82}_{2.69}$	$28.28_{1.49}$	$\underline{82.72}_{0.83}$	$42.98_{1.88}$	$68.63_{0.71}$	$55.03_{1.56}$	$57.24_{17.79}$
TSplit$_{max}$	$57.5_{5.44}$	$32.34_{2.51}$	$80.74_{0.51}$	$43.83_{1.59}$	$\underline{83.93}_{1.0}$	$56.68_{2.89}$	$59.17_{18.45}$
TSplit$_{median}$	$56.19_{2.73}$	$\underline{33.11}_{2.48}$	$80.79_{0.41}$	$\underline{43.85}_{2.97}$	$82.5_{1.79}$	$56.77_{0.92}$	$58.87_{17.98}$
TSplit$_{mean}$	$50.4_{5.4}$	$32.04_{1.89}$	$80.62_{0.68}$	$43.69_{2.77}$	$83.21_{1.2}$	$59.55_{1.34}$	$58.25_{18.64}$
TSplit$_{percentile}$	$58.52_{5.41}$	$31.42_{2.1}$	$80.54_{0.82}$	$43.04_{2.91}$	$82.85_{1.34}$	$\underline{60.22}_{3.06}$	$\underline{59.43}_{18.48}$

Table 4: **Easy setup:** *TSplit resamples all data based on templatness, even resampling the test data.* We compare TSplit against fine-tuning and evaluating on the original dataset splits (see Table 5).

Dataset	Training Size	Validation Size	Test Size
MultiOff	473	119	151
Memotion 3	5700	1426	1374
FigMemes	3246	362	1533
MAMI	7996	1999	1005

Table 5: Example TSplit dataset split sizes derived from the entire dataset. The encoder was `ViT-L/14@336px`.

ficult tasks easier by forcing the model to learn general, meme-specific properties. Aggarwal et al. (2024) recently achieved an F1 of 85.0 on MAMI (A) using OCR text, focusing their investigation on hate speech in memes. Previously reported generic results in Zhang and Wang (2022) relied on fine-tuning and ensembling multimodal models, including CLIP (see Table 18), suggesting that both tasks are difficult, and our own results show that CLIP alone is a poor performer. While TSplit is based on the concept of meme templates, our UIs account for non-templatic memes or images, common in meme datasets. By removing meme/image conceptual overlap between the train and the test split, our approach is general and the model cannot rely on leaked information or spurious artifacts, resulting in two tasks that no longer require ensembling.

A tolerant view of template instances results in consistent, strong performance, especially with the largest encoder, where once more TSplit$_{max}$ is the overall strongest system, consistent with our earlier findings. The efficacy of TSplit$_{max}$ is corroborated not only by the results presented here but also by two additional experimental setups. In the first set of experiments, we investigated the regularization properties of TSplit. For this analysis, the models were fine-tuned over 20 epochs prior to the final evaluation and were therefore very likely overfit. TSplit$_{max}$, when paired with the largest encoder, once more emerged as the most robust performer overall. Furthermore, we conducted experiments involving the application of TSplit to the entire dataset, followed by downsampling the optimization data based either on *templateness* or through random downsampling of the reorganized splits to match the size of the original downsampling experiments (see Table 2). Notably, downsampling based on *templateness* was capable of discarding up to 2,500 samples while achieving comparable, if not superior, performance relative to random downsampling or standard fine-tuning. These findings provide further evidence of the sample efficiency of meme templates (see Section A.2).

5.2 TSplit Error Analysis

We performed error analysis by examining correct and incorrect predictions made by TSplit on the FigMemes dataset. Note that we examined samples that were misclassified by Original$_{\texttt{ViT-L/14@336px}}$, i.e., difficult memes.

TSplit appears to be robust to errors in the extracted OCR text, as it predicted the correct label for the first and second entries in Figure 4, despite the OCR text being incorrect and the memes being arguably non-templatic. The model also excelled in cases involving templatic memes that follow established formats or memes that employ a templatic character, such as Pepe the Frog. For ex-

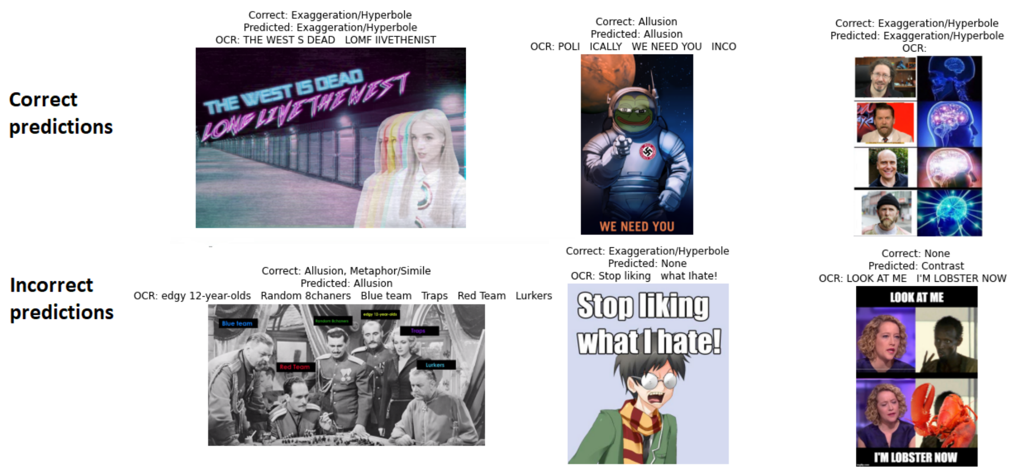

Correct predictions
Incorrect predictions
Correct: Exaggeration/Hyperbole
Predicted: Exaggeration/Hyperbole
OCR: THE WEST S DEAD LOMF IIVETHENIST
THE WEST IS DEAD LONG LIVE THE WEST
Correct: Allusion
Predicted: Allusion
OCR: POLI ICALLY WE NEED YOU INCO
WE NEED YOU
Correct: Exaggeration/Hyperbole
Predicted: Exaggeration/Hyperbole
OCR:
Correct: Allusion, Metaphor/Simile
Predicted: Allusion
OCR: edgy 12-year-olds Random 8chaners Blue team Traps Red Team Lurkers
Correct: Exaggeration/Hyperbole
Predicted: None
OCR: Stop liking what Ihate!
Stop liking what I hate!
Correct: None
Predicted: Contrast
OCR: LOOK AT ME I'M LOBSTER NOW
LOOK AT ME
I'M LOBSTER NOW

Figure 4: Representative examples of correct and incorrect predictions from TSplit on the FigMemes dataset.

ample, "Galaxy Brain"[13] memes were accurately classified, demonstrating the model's strength in recognizing hyperbolic and exaggerated content within familiar templates. Similarly, the "Pepe Astronaut" meme was correctly classified as an allusion. TSplit's familiarity with memes like Pepe made it robust to erroneous OCR text.

TSplit struggles with multilabel, non-templatic memes, such as the first entry in the second row, the model only predicted Allusion, missing the additional Metaphor/Simile label. Non-meme images also pose an issue, such as the second entry, which appears to be a cartoon referencing Harry Potter. A similar misclassification arose with the "Look at Me, I'm the Captain Now"[14] meme, classified as "Contrast" rather than "None". This template relies on juxtaposition for humor, but the model appeared to over-rely on structural cues, misinterpreting its intent. This may be due to the meme referencing an obscure Jordan Peterson interview[15] and the template being heavily manipulated with overlaid images of Cathy Newman and a lobster.

The most challenging cases involved non-templatic memes or images that are arguably not memes. The World War II "Teams" meme illustrates this difficulty. While the model correctly identified "Allusion" based on the historical imagery, it failed to capture the metaphorical elements introduced by labels like "Blue Team," "Red Team", and "Lurkers." These labels transform the meme into a commentary on social dynamics of internet subcultures, adding a layer of metaphor that the model missed. Incorporating culturally and historically informed optimization data is an interesting topic we leave for future research.

6 Conclusion and Future Work

We analyzed the power of meme templates. To anchor our analysis to memes, we created the KYMKB, containing more than 54,000 images and 5,200 base templates with detailed information about each one. We first showed that a comparison of templates to memes creates a strong signal that we can leverage in downstream tasks. We therefore proposed TSplit and found it is sample-efficient and can result in robust models. It was not our goal to create state-of-the-art meme classifiers, but we believe our methods are convincing demonstrations of the strength of meme templates – memes may be difficult to analyze, but they are not random.

The potential applications of meme templates are vast. Our findings demonstrate their significant impact on topics such as retrieval (see Section 4), sample efficiency, model performance, and even multilingual capabilities. However, our work represents only an initial exploration of this area. Future research could further investigate these applications and other critical directions, contributing to a deeper understanding of the role and utility of meme templates in computational research.

Our resources provide unified, inexpensive tools for future research grounded in meme knowledge, and specifically, meme templates. In future work, we will apply the KYMKB to more datasets and languages in a cross-language setup and explore automatically augmenting the KYMKB with new memes and with new templates.

[13] https://knowyourmeme.com/memes/galaxy-brain
[14] https://knowyourmeme.com/memes/look-at-me-im-the-captain-now
[15] https://www.youtube.com/watch?v=aMcjxSThD54

Acknowledgments

We would like to thank Tobin Bates, Max Glockner, Derek Hommel, and Timour Igamberdiev for sharing inspiring discussions with us and Irina Bigoulaeva and Seolhwa Lee for their invaluable feedback on an early draft of our manuscript. LB and IG acknowledge the German Federal Ministry of Education and Research (BMBF) under the promotional reference 13N15897 (MISRIK) and the Hessian Ministry of Science and the Arts (HMWK) within the projects "The Third Wave of Artificial Intelligence - 3AI" (200009-62200024), hessian.AI and within their joint support of the National Research Center for Applied Cybersecurity ATHENE. PEC was supported by the Pioneer Centre for AI, DNRF grant number P1.

Limitations

KYM is in our view the best resource for meme-related knowledge, but this does not mean that it is the only resource, nor does it mean that all meme posters necessarily agree on the interpretation of a template or a meme. Like all forms of communication, there is ambiguity in what a given instance means. Not all memes are templatic, but it is our belief that the most popular memes are, at least based on how meme datasets are created. TSplit quantifies the notion of templateness for the sake of computation, however, we believe that truly determining the templateness of a given meme is not trivial and it is certainly not the case that KYM contains all known templates. We have devised a measure by which to determine templateness, but it is only applicable within the limited scope of ML, where memes are viewed as images (see Section A.1). We have performed an examination of multimodal LLM prompting performance and found such a paradigm to be insufficient for meme understanding (see Section A.14) and our findings are consistent with Hwang and Shwartz (2023). More recent vision language models, such as Otter (Li et al., 2023) or IDEFICS,[16] might be stronger, we are skeptical due to their incremental nature and the poor performance of LLaVA (Liu et al., 2023), even supplemented by the KYMKB. However, we did not test this ourselves. We have not considered pay-to-use corporate artifacts for reasons of reproducibility and accessibility.

While our analysis demonstrates that grounding in templates can have multilingual effects, we have not addressed the cultural or linguistic limitations or potential biases inherent in our approach or our knowledge base. Future work should include an in-depth investigation of KYMKB's diversity in terms of culture, language, and humor.

There is an argument to made for template instances being equally distributed between dataset splits, as a model has no hope of interpreting a novel template at test time. This is not our own view because it is our belief that we should be testing model robustness on memes that are as novel as possible. However, we enable this functionality in TSplit to encourage future research on model-template understanding.

The distinction between different meme types is not always clear and is arguably subjective. In future work, we will use the KYMKB to develop a taxonomy of memes in order to aid the development of meme-aware systems.

Ethics Statement

It is possible that the resources and insights we have developed and discussed may be misused to spread harmful memes more effectively. Abuse is an unfortunate drawback of all tools and technology. We hope that our work is used instead to create systems that have stronger meme understanding such that we can automatically and accurately flag dangerous memes to halt their spread on social media. To this end, we created and demonstrated how the KYMKB can be used practically and analytically. To investigate model-meme understanding, we conducted thorough classification experiments with adversarial meme datasets via TSplit and found that we can use it to discourage models from taking undesirable shortcuts to meme understanding, which results in more robust models. Throughout this work, we have emphasized the dangers that memes can present, pointed out how our field is lacking in its approach to memes, and taken what we believe are the first steps on a long road to intelligent systems that understand memes.

We do not publish data, but following similar work that provided access to real-world content (Fan et al., 2019; Hanselowski et al., 2019; Zlatkova et al., 2019), we make our scraping and organization code available to recreate the KYMKB and use Wayback Machine[17] (WM) as a best effort to ensure reproducibility, as the WM URLs will not foreseeably change. If KYM does not wish

[16] https://huggingface.co/blog/idefics

[17] https://wayback-api.archive.org/

for their content to be accessed via WM, they have the option to opt out,[18] such that their content is removed.

References

Piush Aggarwal, Pranit Chawla, Mithun Das, Punyajoy Saha, Binny Mathew, Torsten Zesch, and Animesh Mukherjee. 2023. Hateproof: Are hateful meme detection systems really robust? In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 3734–3743. ACM.

Piush Aggarwal, Jawar Mehrabanian, Weigang Huang, Özge Alacam, and Torsten Zesch. 2024. Text or image? what is more important in cross-domain generalization capabilities of hate meme detection models? In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 104–117, St. Julian's, Malta. Association for Computational Linguistics.

Ramy Baly, Giovanni Da San Martino, James Glass, and Preslav Nakov. 2020. We can detect your bias: Predicting the political ideology of news articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4982–4991, Online. Association for Computational Linguistics.

Meghana Bhange and Nirant Kasliwal. 2020. HinglishNLP at SemEval-2020 task 9: Fine-tuned language models for Hinglish sentiment detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 934–939, Barcelona (online). International Committee for Computational Linguistics.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 785–794. ACM.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *European Conference on Computer Vision*.

R Dawkins. 1976. *The Selfish Gene*. Oxford University Press, Oxford, UK.

[18]https://help.archive.org

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. SemEval-2021 task 6: Detection of persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98, Online. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Yuhao Du, Muhammad Aamir Masood, and Kenneth Joseph. 2020. Understanding visual memes: An empirical analysis of text superimposed on memes shared on twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 153–164.

Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online. Association for Computational Linguistics.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Elisabetta Fersini, Francesca Gasparini, Giulia Rizzi, Aurora Saibene, Berta Chulvi, Paolo Rosso, Alyssa Lees, and Jeffrey Sorensen. 2022. SemEval-2022 task 5: Multimedia automatic misogyny identification. In *Proceedings of the 16th International*

Workshop on Semantic Evaluation (SemEval-2022), pages 533–549, Seattle, United States. Association for Computational Linguistics.

Laura Glitsos and James Hall. 2019. The pepe the frog meme: an examination of social, political, and cultural implications through the tradition of the darwinian absurd. *Journal for Cultural Research*, 23(4):381–395.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Andreas Hanselowski, Christian Stab, Claudia Schulz, Zile Li, and Iryna Gurevych. 2019. A richly annotated corpus for different tasks in automated fact-checking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 493–503, Hong Kong, China. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Gabriel Hine, Jeremiah Onaolapo, Emiliano De Cristofaro, Nicolas Kourtellis, Ilias Leontiadis, Riginos Samaras, Gianluca Stringhini, and Jeremy Blackburn. 2017. Kek, cucks, and god emperor trump: A measurement study of 4chan's politically incorrect forum and its effects on the web. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1):92–101.

Cille Hvass Holm. 2021. What do you meme? the sociolinguistic potential of internet memes. *Leviathan: Interdisciplinary Journal in English*, 7:1–20.

EunJeong Hwang and Vered Shwartz. 2023. MemeCap: A dataset for captioning and interpreting memes. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1433–1445, Singapore. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Vasiliki Kougia, Simon Fetzel, Thomas Kirchmair, Erion Çano, Sina Moayed Baharlou, Sahand Sharifzadeh, and Benjamin Roth. 2023. Memegraphs: Linking memes to knowledge graphs. In *Document Analysis and Recognition - ICDAR 2023*, pages 534–551, Cham. Springer Nature Switzerland.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. 2023. Otter: A multi-modal model with in-context instruction tuning. *ArXiv preprint*, abs/2305.03726.

Hongzhan Lin, Ziyang Luo, Jing Ma, and Long Chen. 2023. Beneath the surface: Unveiling harmful memes with multimodal reasoning distilled from large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9114–9128, Singapore. Association for Computational Linguistics.

Chen Liu, Gregor Geigle, Robin Krebs, and Iryna Gurevych. 2022. 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. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yao Lu, Jiayi Wang, Raphael Tang, Sebastian Riedel, and Pontus Stenetorp. 2024. Strings from the library of babel: Random sampling as a strong baseline for prompt optimisation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2221–2231, Mexico City, Mexico. Association for Computational Linguistics.

Shreyash Mishra, S Suryavardan, Parth Patwa, Megha Chakraborty, Anku Rani, Aishwarya Reganti, Aman Chadha, Amitava Das, Amit Sheth, Manoj Chinnakotla, Asif Ekbal, and Srijan Kumar. 2023. Memotion 3: Dataset on sentiment and emotion analysis of codemixed hindi-english memes. *ArXiv preprint*, abs/2303.09892.

Abel L Peirson and E Meltem Tolunay. 2018. Dank learning: Generating memes using deep neural networks. *ArXiv preprint*, abs/1806.04510.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Shraman Pramanick, Dimitar Dimitrov, Rituparna Mukherjee, Shivam Sharma, Md. Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021a. Detecting harmful memes and their targets. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2783–2796, Online. Association for Computational Linguistics.

Shraman Pramanick, Shivam Sharma, Dimitar Dimitrov, Md. Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021b. MOMENTA: A multimodal framework for detecting harmful memes and their targets. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4439–4455, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jingnong Qu, Liunian Harold Li, Jieyu Zhao, Sunipa Dev, and Kai-Wei Chang. 2022. DisinfoMeme: A Multimodal Dataset for Detecting Meme Intentionally Spreading Out Disinformation. *ArXiv preprint*, abs/2205.12617.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Shivam Sharma, Ramaneswaran S, Udit Arora, Md. Shad Akhtar, and Tanmoy Chakraborty. 2023. MEMEX: Detecting explanatory evidence for memes via knowledge-enriched contextualization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5272–5290, Toronto, Canada. Association for Computational Linguistics.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Will Styler. 2020. The linguistics of memes. `https://wstyler.ucsd.edu/talks/meme_linguistics.html#/`; accessed 26-June-2023.

Shardul Suryawanshi, Bharathi Raja Chakravarthi, Mihael Arcan, and Paul Buitelaar. 2020. Multimodal meme dataset (MultiOFF) for identifying offensive content in image and text. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 32–41, Marseille, France. European Language Resources Association (ELRA).

Riccardo Tommasini, Filip Illievski, and Thilini Wijesiriwardene. 2023. IMKG: the internet meme knowledge graph. In *The Semantic Web - 20th International Conference, ESWC 2023, Hersonissos, Crete, Greece, May 28 - June 1, 2023, Proceedings*, volume 13870 of *Lecture Notes in Computer Science*, pages 354–371. Springer.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *ArXiv preprint*, abs/2302.13971.

Nathan Walter, Michael J Cody, Larry Zhiming Xu, and Sheila T Murphy. 2018. A Priest, a Rabbi, and a Minister Walk into a Bar: A Meta-Analysis of Humor Effects on Persuasion. *Human Communication Research*, 44(4):343–373.

Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, Candace Ross, Adam Polyak, Russell Howes, Vasu Sharma, Puxin Xu, Hovhannes Tamoyan, Oron Ashual, Uriel Singer, Shang-Wen Li, Susan Zhang, Richard James, Gargi Ghosh, Yaniv Taigman, Maryam Fazel-Zarandi, Asli Celikyilmaz, Luke Zettlemoyer, and Armen Aghajanyan. 2023. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. *ArXiv preprint*, abs/2309.02591.

Jing Zhang and Yujin Wang. 2022. SRCB at SemEval-2022 task 5: Pretraining based image to text late sequential fusion system for multimodal misogynous meme identification. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 585–596, Seattle, United States. Association for Computational Linguistics.

Naitian Zhou, David Jurgens, and David Bamman. 2024. Social meme-ing: Measuring linguistic variation in memes. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3005–3024, Mexico City, Mexico. Association for Computational Linguistics.

Dimitrina Zlatkova, Preslav Nakov, and Ivan Koychev. 2019. Fact-checking meets fauxtography: Verifying claims about images. 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 2099–2108, Hong Kong, China. Association for Computational Linguistics.

A Appendix

A.1 What's in a Meme?

Memes are not just images that sometimes have text. The KYMKB captures this fact and how far we as a community are from meme understanding. Consider *Leeroy Jenkins*,[19] a template that references a popular YouTube video[20] where a player in World of Warcraft[21] makes a brash decision while yelling his name, Leeroy Jenkins. This results in a party of players losing a fight to a monster.

An instance of this template is not merely some image, but rather hollering *Leeroy Jenkins* or using the audio from the original template when performing a reckless act that will likely have negative consequences. A concrete example of this can be seen in a recent YouTube video.[22] We are unaware of any approach which considers memes in audio form. Despite this template originating in 2005, it is still referenced almost 20 years later, demonstrating the longevity of popular templates. The video in question is a compilation of memes, but is not composed of still images sometimes with text, but rather audio and video. At the time of writing, this video has more than 6.6 million views, which we feel is compelling evidence that this is a more realistic representation of memes than what can be found in the literature. This video is not an edge case either, but rather a case that has not been considered in previous work, exemplified by the relevant YouTube channel having 18 other such videos, each with more than one million views. Such examples may seem anomalous, but we argue otherwise and we believe that such an interpretation is a consequence of the narrow scope of the literature. In Section A.12, we provide a detailed discussion about additional *edge case* examples contained within the KYMKB.

In order to make our work digestible, we have conformed to the notion of memes that the AI community has converged to. TSplit, for example, relies on the concept that memes are images in order to perform classification, but our method is meant to demonstrate the usefulness of templates and a shortcoming of the literature. Templatic memes are only the tip of the iceberg when it comes to understanding this form of communication and the KYMKB provides a wealth of knowledge we can utilize to create systems capable of interpreting memes.

[19]https://knowyourmeme.com/memes/leeroy-jenkins

[20]https://www.youtube.com/watch?v=mLyOj_QD4a4&t=1s

[21]https://worldofwarcraft.blizzard.com

[22]https://www.youtube.com/watch?v=UdWv202brqo (at 1:25).

A.2 Template-Aware Splitter details

In this Appendix section, we provide additional results from fine-tuning with TSplit. Our experimental conditions are the same as Section 5 except here we perform inference with the model fine-tuned for 20 epochs to highlight TSplit's regularization effects. These results can be found in Table 6.

In the case of MultiOff, we see that smaller encoders struggle with our resplit datasets, while the larger encoder overfits to the original split. In such cases, TSplit appears to have a regularization effect. For Memotion 3 (A), a difficult task with many templatic memes, TSplit outperforms our baseline and consistently attains scores of approximately 33–35, and we again get multilingualism without even trying. Task B tells a different story, where the TSplit datasets appear harder than the original.

FigMemes remains a challenging task, and the largest models are required to be performant, but are still prone to overfitting. We again note the regularization properties of TSplit, beating our baseline with an F1 of 47.91.

A.2.1 TSplitting the Enitre Dataset and Downsampling

In this section, we present additional experiments to further investigate the efficacy of TSplit sampling. Specifically, we resample the entire dataset using TSplit, followed by downsampling and selective discarding of optimization data in two distinct ways: 1) by randomly downsampling to match the sizes outlined in Table 10, and 2) by sampling templates that may appear in the training or validation data based on dividing the random downsampled size by the original training size, i.e., $\text{train\_ratio} = \frac{\text{downsample\_size}}{\text{original\_training\_size}}$. As detailed in Section 5, we accomplish this by generating an array of uniquely detected templates and unique identifiers (UIs) that are eligible to appear in the training data. This array is then shuffled, and a cutoff index is determined. All detected templates or UIs preceding the cutoff index is discarded, while the remaining data is used for training. Once we have the ratio, we then multiply it by the number of detected training templates, as follows:

Split	MultiOff	Memotion 3 (A)	Memotion 3 (B)	FigMemes	MAMI (A)	MAMI (B)	Overall
			Encoder: `ViT-L/14@336px`				
Original$_{ViT-L/14@336px}$	$59.77_{3.14}$	$26.77_{1.04}$	$\underline{79.8_{0.85}}$	$44.93_{3.03}$	$66.72_{1.22}$	$51.93_{1.18}$	$54.99_{16.75}$
TSplit$_{max}$	$62.16_{2.14}$	$32.96_{1.26}$	$77.95_{1.04}$	$\mathbf{47.91}_{1.57}$	$\mathbf{83.63}_{2.36}$	$\mathbf{59.65}_{4.66}$	$\mathbf{60.7}_{17.12}$
TSplit$_{median}$	$58.27_{2.57}$	$34.73_{1.15}$	$78.49_{0.97}$	$42.51_{9.36}$	$82.15_{2.33}$	$56.33_{1.4}$	$58.75_{17.24}$
TSplit$_{mean}$	$61.57_{3.98}$	$\mathbf{35.04}_{1.46}$	$78.1_{1.74}$	$45.36_{3.5}$	$81.22_{2.62}$	$56.7_{3.35}$	$59.66_{16.47}$
TSplit$_{percentile}$	$\underline{63.58_{3.0}}$	$33.53_{2.59}$	$77.38_{1.94}$	$40.26_{10.87}$	$80.52_{0.92}$	$53.32_{2.93}$	$58.1_{17.56}$
			Encoder: `ViT-B/32`				
Original$_{ViT-B/32}$	$\underline{60.43_{1.63}}$	$29.08_{0.9}$	$\mathbf{80.26}_{1.3}$	$35.92_{2.34}$	$66.06_{1.99}$	$51.97_{1.23}$	$53.95_{17.46}$
TSplit$_{max}$	$56.02_{2.38}$	$33.2_{1.27}$	$78.04_{1.27}$	$37.96_{1.52}$	$79.74_{1.42}$	$55.53_{3.04}$	$56.75_{17.76}$
TSplit$_{median}$	$53.19_{4.32}$	$33.92_{1.64}$	$77.8_{0.87}$	$\underline{39.8_{4.32}}$	$79.87_{1.81}$	$\underline{56.08_{2.38}}$	$\underline{56.92}_{18.97}$
TSplit$_{mean}$	$57.73_{3.44}$	$32.54_{1.39}$	$76.84_{0.55}$	$36.68_{1.71}$	$\underline{80.13_{1.68}}$	$55.16_{2.61}$	$56.51_{17.99}$
TSplit$_{percentile}$	$55.68_{4.92}$	$\underline{34.49_{1.7}}$	$76.41_{2.2}$	$38.25_{2.72}$	$77.17_{3.65}$	$55.31_{1.94}$	$56.22_{16.55}$
			Encoder: `ViT-B/16`				
Original$_{ViT-B/16}$	$\mathbf{64.65}_{2.12}$	$27.28_{0.65}$	$\underline{79.49_{3.0}}$	$40.58_{2.0}$	$67.61_{1.96}$	$51.5_{2.65}$	$55.18_{17.51}$
TSplit$_{max}$	$58.23_{4.87}$	$\underline{34.95_{1.42}}$	$77.47_{0.64}$	$39.4_{4.93}$	$80.1_{0.83}$	$\underline{57.13_{4.11}}$	$57.88_{17.06}$
TSplit$_{median}$	$57.31_{2.91}$	$33.33_{1.54}$	$78.34_{1.59}$	$40.96_{0.7}$	$\underline{80.74_{1.29}}$	$54.36_{2.69}$	$57.51_{17.52}$
TSplit$_{mean}$	$59.74_{3.2}$	$34.16_{2.64}$	$77.28_{1.83}$	$40.16_{3.19}$	$80.31_{1.21}$	$55.99_{3.38}$	$57.94_{17.14}$
TSplit$_{percentile}$	$59.4_{6.11}$	$34.5_{1.89}$	$77.95_{1.34}$	$\underline{42.71_{2.02}}$	$79.28_{0.71}$	$54.33_{3.45}$	$\underline{58.03}_{16.95}$

Table 6: TSplit compared against the original dataset. We group results by their encoder (*Original* subscript), where the encoders are organized by size in descending order. The best performer in each group is underlined. The best performer for each dataset is in **bold**. The *Overall* column is the mean and standard deviation of each row for insight into performance on aggregate.

Dataset	ViT-B/16 Training Size	ViT-B/32 Training Size	ViT-B/16 Validation Size	ViT-B/32 Validation Size	ViT-B/16 Dummy Test Size	ViT-B/32 Dummy Test Size
MultiOff	367	354	93	90	134	150
Memotion 3	4930	4723	1233	1181	837	1096
FigMemes	2293	2327	256	260	1050	1012
MAMI	7213	7295	1804	1824	983	881

Table 7: Example TSplit dataset split sizes for `ViT-B/16` and `ViT-B/32`. The dummy test data was discarded.

$$\text{cutoff} = \lfloor \text{train\_templates} \times \text{train\_ratio} \rfloor$$

Similarly, to create the validation data, we sample from the training data based on the ratio of the randomly downsampled split size to the size of the original split. This is an aggressive means of downsampling the optimization data based on templateness, where we might discard more then $2,500$ samples (see Table 8).

Dataset	Training	Validation	Test	Discard
MultiOff	461	69	137	76
Memotion3	3650	913	1609	2328
FigMemes	2683	143	1376	939
MAMI	6145	1537	740	2578

Table 8: Dataset sizes for MultiOff, Memotion3, FigMemes, and MAMI with full TSplit downsampling.

Our experimental setup remains the same, but we only experimented with the `ViT-L/14@336px` CLIP model. We compare these two downsampled TSplit versions against fine-tuning on the original splits.

The results presented in Table 9 reveal several notable trends across datasets and sampling methods. Fine-tuning on the original splits achieves the strongest performance for MultiOff ($\mathbf{63.6}$) and Memotion 3 (B) ($\mathbf{81.7}$).

Template-based downsampling demonstrates notable improvements, even for datasets that do not contain a high proportion of templates. For example, TSplit$_{\text{max}}$ from the last grouping achieves the best performance for MAMI (A) ($\mathbf{85.71}$). Similarly, TSplit$_{\text{median}}$ stands out as the top performer for MAMI (B) ($\mathbf{63.05}$) and achieves the highest overall mean score (61.92), highlighting its robustness across datasets.

Interestingly, for FigMemes, the improvements with template-based downsampling are marginal, but we see a boost if we first TSplit the entire dataset and combine this with random downsampling. In contrast, for Memotion 3 (A), TSplit$_{\text{median}}$ (35.51) performs best, underscoring that the optimal sampling method may vary depending on the dataset's characteristics.

Overall, template-based TSplit methods tend to outperform random downsampling or even using more data for optimization, that is, standard fine-tuning. These findings indicate that while fine-tuning on the original splits remains competitive for certain datasets, template-based sampling offers a more sample-efficient and effective alternative.

Split	MultiOff	Memotion 3 (A)	Memotion 3 (B)	FigMemes	MAMI (A)	MAMI (B)	Overall
Original$_{ViT-L/14@336px}$	$\mathbf{63.64}_{2.31}$	$26.38_{1.57}$	$\mathbf{81.7}_{2.33}$	$47.79_{1.43}$	$71.82_{4.05}$	$56.6_{1.56}$	$57.99_{17.76}$
Random Downsampling							
TSplit$_{max}$	$60.21_{1.9}$	$35.12_{1.0}$	$\underline{80.74}_{0.62}$	$48.34_{3.17}$	$84.32_{3.36}$	$61.45_{2.88}$	$\underline{61.7}_{17.13}$
TSplit$_{median}$	$58.18_{7.07}$	$33.13_{2.03}$	$80.34_{0.45}$	$\mathbf{50.15}_{2.19}$	$82.09_{1.73}$	$59.3_{2.9}$	$60.53_{16.95}$
TSplit$_{mean}$	$59.91_{5.28}$	$\underline{35.51}_{0.81}$	$80.44_{0.47}$	$47.2_{0.98}$	$83.15_{1.74}$	$\underline{61.65}_{2.63}$	$61.31_{16.88}$
TSplit$_{percentile}$	$\underline{62.59}_{4.64}$	$35.48_{1.04}$	$80.66_{0.42}$	$44.56_{9.82}$	$\underline{84.94}_{1.55}$	$60.2_{2.56}$	$61.4_{17.72}$
TSplit Downsampling							
TSplit$_{max}$	$58.83_{2.05}$	$33.77_{1.35}$	$\underline{80.8}_{0.88}$	$46.19_{2.02}$	$\mathbf{85.71}_{2.53}$	$61.62_{2.54}$	$61.15_{18.11}$
TSplit$_{median}$	$60.14_{3.57}$	$\mathbf{35.68}_{2.06}$	$80.36_{0.62}$	$\underline{48.45}_{2.63}$	$83.85_{1.75}$	$\mathbf{63.05}_{3.18}$	$\mathbf{61.92}_{16.81}$
TSplit$_{mean}$	$\underline{60.79}_{3.81}$	$33.48_{1.6}$	$79.84_{0.43}$	$46.35_{2.09}$	$84.15_{0.45}$	$59.39_{1.87}$	$60.67_{17.63}$
TSplit$_{percentile}$	$60.18_{4.35}$	$35.24_{1.44}$	$80.59_{0.92}$	$47.55_{1.26}$	$83.81_{2.4}$	$60.66_{2.06}$	$58.1_{17.56}$

Table 9: Fine-tuning on the original splits is compared against T-Splitting the entire dataset followed by random downsampling, and T-Splitting the entire dataset with downsampling based on templateness. The best performer within each group is underlined, and the best performer for each dataset is **bolded**. The *Overall* column represents the mean and standard deviation for each row, providing insight into the aggregate performance.

A.2.2 Template-Split Analysis

Split and threshold	MultiOff	Memotion 3	FigMemes	MAMI
Train$_{max}$	228 / 4	1407 / 141	1335 / 59	1961 / 136
Test$_{max}$	55 / 2	303 / 28	568 / 29	198 / 11
Train$_{median}$	204 / 88	1170 / 2051	1155 / 547	1764 / 1191
Test$_{median}$	50 / 23	249 / 441	525 / 204	172 / 123
Train$_{mean}$	194 / 100	1106 / 2614	1122 / 655	1700 / 1510
Test$_{mean}$	47 / 26	228 / 569	497 / 264	170 / 150
Train$_{percentile}$	167 / 176	890 / 3562	964 / 1198	1506 / 2789
Test$_{percentile}$	41 / 44	205 / 748	419 / 506	138 / 291

Table 10: Example of how TSplit reorganizes datasets, where we show the number of detected templates / number of unique identifiers in each split for each thresholding method. `ViT-L/14@336px` was used as the CLIP encoder.

In this section, we examine how TSplit samples templates and unique identifiers. As Table 10 shows, using the maximum distance from template to examples as the threshold value for each template results in TSplit detecting more templates, while using the 25th percentile results in more unique identifiers.

To empirically verify how TSplit resplits datasets, we sampled 1000 memes from FigMemes from both the training and test split of our reorganized datasets. We then manually inspected the memes, looking for overlapping templates, template references, or meme-related artifacts. We did this under all four thresholding techniques, where the CLIP encoder was `ViT-L/14@336px`.

Under TSplit$_{max}$ and TSplit$_{median}$, we observed that certain templatic characters, such as Pepe the Frog, appeared in both splits. Pepe was present in another base template as well as in a non-templatic meme, but no single template appeared across both splits. In contrast, under TSplit$_{mean}$, we noted similar templatic characters and observed one overlapping template, Picardía / Thumbs Up Emoji Man, which is frequently used on 4chan to mock political ideologies.[23] This template's flexibility in appearance suggests that a strict interpretation of template instances would naturally result in its presence in both splits. Under TSplit$_{percentile}$, while no explicit template overlap was noted, the same templatic characters—such as Pepe, Picardía, SpongeBob, and Spiderman—appeared across both splits.[24] Although these examples are not instances of a formal template, they evoke a similar emotional resonance, akin to the templates they reference.

A.3 Template-Label Counter

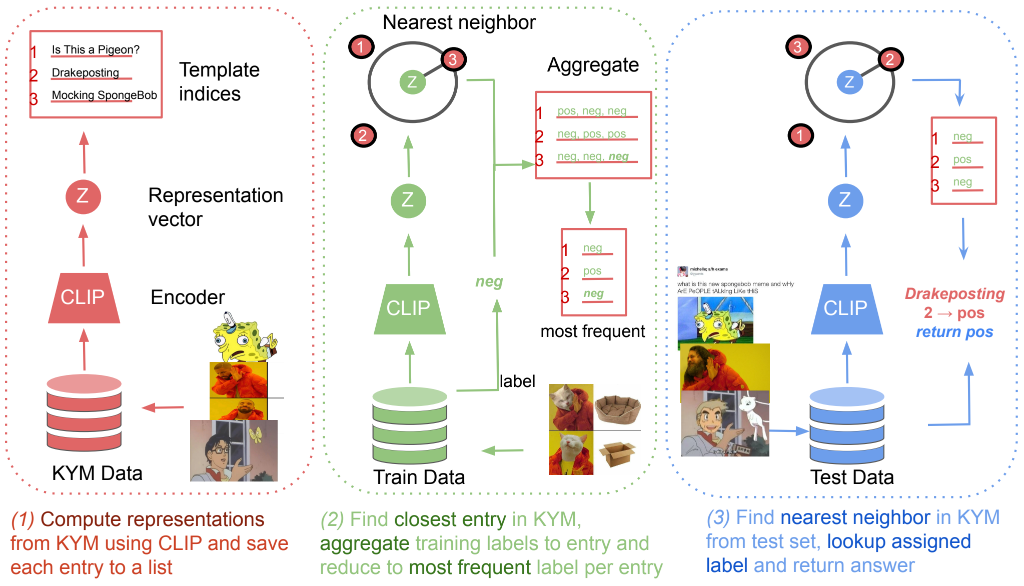
We hypothesize that many meme datasets are often nothing more than examples of popular templates we have collected in the KYMKB. We should therefore be able to compare memes to templates, select the most similar template, and obtain a meme-specific context. To test this, we matched templates to memes in the training split of a dataset. We can then assign a meme's label to another meme if they share the same template, i.e., a novel meme in the test split of that dataset (see Figure 5).

Meme knowledge We again opted for nearest neighbor indexing as a similarity measure and formalize this as a ranking task, where we first create a template reference, $ref = f(X_{KYMKB})$.

Dataset knowledge To learn a dataset's labeling scheme, we encode the training data, $query_{train} = f(X_{train})$, and query our index, selecting the closest template and recording the label for each training instance. TLC then reduces each

[23] `https://knowyourmeme.com/memes/picardia-thumbs-up-emoji-man`

[24] `https://knowyourmeme.com/memes/spider-man-pointing-at-spider-man`



Is This a Pigeon?
Drakeposting
Mocking SpongeBob
Template indices
Representation vector
CLIP
Encoder
KYM Data
(1) Compute representations from KYM using CLIP and save each entry to a list
Nearest neighbor
Aggregate
pos, neg, neg
neg, pos, pos
neg, neg, neg
most frequent
neg
label
Train Data
(2) Find closest entry in KYM, aggregate training labels to entry and reduce to most frequent label per entry
Drakeposting
2 → pos
return pos
Test Data
(3) Find nearest neighbor in KYM from test set, lookup assigned label and return answer

Figure 5: TLC encodes the KYMKB and computes a nearest neighbor index. We then encode the training data and query our lookup, recording each template's most frequent class. We query the index and assign the closest template's label to test samples. As an example, we use *pos* and *neg* as labels for a sentimental analysis task.

index (template) to the most frequent label:

$$\arg\max_{ref} count(rank(ref, query_{train}))$$

Here our $rank$ function sorts entries in the KYMKB in ascending order based on their Euclidean distance from a query vector.

Testing meme and dataset knowledge The final step is to encode test data, $query_{test} = f(X_{test})$, and then query our lookup. We then assign the most frequent label for a template to a test instance, $\hat{y} = rank(ref, query_{test})$. If we find a template not seen during training, we backoff to the most frequent label in the training data.

Hyperparameter values TLC has the option to ignore the meme itself and instead to match the *about* section of templates to the OCR text of a novel meme. Alternatively, we can choose to consider base templates or also examples for encoding knowledge about the meme. Multiple neighbors can be searched over, selecting the most common template or label among them. We can also use multiple modalities, combining the *about* section from the template/example and the OCR text, respectively, with the template and the novel meme embeddings. We experimented with concatenating the CLIP embeddings of both modalities, fusing via the Hadamard product, normalizing and averaging the two modalities as the final input vector (Yu et al., 2023), and a type of late fusion, where the text and the image representations vote separately and we then aggregate. After the hyperparameter values are set, TLC is deterministic (see Appendix A.5). Note that TLC is reliant on the KYMKB.

A.4 Classification Experiments

Baselines and experimental setup We test various versions of TLC on six meme classification datasets: FigMemes with seven labels of different types of figurative language, MultiOff a binary dataset where the labels can be offensive or non-offensive, Memotion 3, where Task A has three labels for sentiment analysis and Task B has four for different types of emotion, and MAMI with Task A being binary misogyny detection and Task B having four labels with different types of misogyny being expressed (see Tables 18 and 19). We choose these datasets because they were made to address the issue of harmful meme detection and to frame our analysis of meme templates in the light of this important goal. Our baseline is a majority class classifier. Each dataset used Google Vision to provide OCR text overlaid on memes.[25]

Results and Discussion Table 11 shows our results. We display the best-performing version of TLC, comparing embedding text versus templates versus templates and examples. We also show the best result from previous work, where a PLM was

[25]https://cloud.google.com/use-cases/ocr

Method	MultiOff	Memotion 3 (A)	Memotion 3 (B)	FigMemes	MAMI (A)	MAMI (B)
Majority	37.92	21.5	72.59	5.72	33.33	18.2
Best previous: text only	*54.0*	NA	NA	34.06	83.0	NA
Best previous: vision only	24.0	NA	NA	*47.69*	*85.0*	NA
Best previous: vision+text	50.0	*33.28*	*74.74*	46.69	83.4	*73.1*
TLC$_{Text}$	51.83	35.4	77.6	21.14	61.86	35.93
TLC$_{Templates}$	**61.89**	**37.77**	79.89	**29.8**	69.24	39.99
TLC$_{Templates+Instances}$	58.58	37.04	**80.49**	28.97	**70.0**	**40.21**

Table 11: Classification results for the best-performing version of TLC (**in bold**) compared against the best performing method from the related work (*in italics*). *Instances* refers to template examples in the KYMKB. See Section A.7 for TLC hyperparameter configurations and Appendix A.13 for modelling information from previous work.

fine-tuned on OCR text, the meme itself, or a multimodal representation of the two. TLC beats our majority class classifier, but this baseline is competitive a fine-tuned PLM for Memotion 3 (B).

TLC's performance consistently improves as we consider more modalities. Encoding the *about* section of a template and the OCR text from a novel meme is strong on its own, especially in the case of Memotion 3. As we add template and meme images, the performance improves by more than ten points for MultiOff. We find that concatenating the image and the text modalities tends to be the strongest TLC configuration, supporting our hypothesis that the base semantics of a meme is explained in the *about* section, but is also captured by the template. We can naturally obtain a better representation of the exact meaning by using meme-specific information from OCR.

The base template is sufficient to encode meme knowledge and is more efficient than also embedding examples. For MultiOff, we see a boost of more than two points when we only consider templates. In other cases, TLC$_{Templates}$ is within one point if not higher than TLC$_{Templates+Instances}$. Encodng one-tenth of the available images results in a strong model, demonstrating that meme datasets can be instances of the KYMKB templates.

In the case of Memotion 3 and MultiOff, our approach is a stronger method than fine-tuning a PLM. We further note that meme templates cross cultural and linguistic boundaries, as indicated by our strong performance on both Memotion 3 datasets, a multilingual dataset of memes in Hindi and English. Templates give us multilinguality for free.

TLC assumes memes belong to a template, but our prediction has no meaning for a picture (which is not a meme). Many meme datasets are not curated to remove non-memes, containing both memes and images. This can be verified in the datasets or by reading the paper. Figure 1 in FigMemes shows an example of a visual metaphor/simile, which is a picture, not a meme (see (f)).

In Figure 1 of MAMI, all examples are not templatic memes and are understandable without knowledge of memes (see Figure 6).This is supported by our template-meme analysis, where we saw that a large percentage of memes in MAMI are unrelated to templates (see Section 4).

A.5 Template-Label Counter details

In this Appendix section, we provide additional details about TLC that could not be provided in the main text due to space limitations. For TLC, each template is associated with an array of observed training labels derived from instances of that template. This array is then reduced to the most frequently occurring label, which serves as the template's representative label. When the same template appears again, such as in the test data, this representative label is assigned to any novel memes associated with the template. There are actually multiple ways we can go about voting if we consider multiple neighbors. First, we could consider multiple templates and then take their most common label, only keeping and recording that label. We refer to this as *template vote*. In cases where we only consider templates and not examples, this would mean often backing off to the most majority class in the dataset because we will find distinct templates. Alternatively, we could keep all labels for a given template and then reduce to its most frequent label, which we refer to as *label vote*. We consider all cases. We find that the template style of voting is the strongest and it is about this configuration that we report results. The only exception to this is MAMI, where we found *label vote* to be the best configuration. This finding is intuitive because MAMI is composed largely of memes which are not templatic and therefore it is the label signal, not the template signal, which is most beneficial for classification. As we are not dealing with probabilities but with a majority, this is reflected in our late fusion implementation. We use *label vote* for both the template and its about section, combine all their labels, find the most common between the two, and keep that label as the final prediction for a given template. If we come across a template not featured in the training data, we back off to the most frequent label in the training split. For the datasets we explored, our implementation of late fusion was not a strong performer. This is intuitive because, as we have shown, using text representations is not as strong as image representations. Voting independently and then aggregating both modalities weakens image performance and is not as strong as other multimodal methods.

Additionally, in FigMemes, the authors tried many different models, for example, fine-tuning BERT (Devlin et al., 2019), which yielded a macro-averaged F1 of 32.62. TLC$_{Templates}$ is competitive with this model, but far cheaper. In Table 3 from their work, we see a great deal of variation, demonstrating the difficulty of the task.

A.6 Template Label Counter: Out of Distribution

TLC's main drawback is its assumption that an input meme or image is employs a meme template. In order overcome this, we add an additional implementation of TLC that uses the same notion of *templateness* as TSplit. That is, if the meme exceeds the maximum distance between template and examples, we declare it non-templatic. In this case, we back off to the most common label in the training data or randomly choose a label from the training data, which we refer to as TLC$_{maj}$ and TLC$_{rand}$, respectively. We compare this out of distribution implementation against making a prediction that assumes all memes are templatic, which we refer to as TLC$_{norm}$. This implementation also allows for more traditional hyperparameter search. We ran extensive experiments with these alternative models, where we again examined performance with different encoders, number of neighbors, and combinations of features. Table 12 shows our results, where we show performance on the test set based on best performance on the validation data. If a validation split did not exist, we created one by sampling 20% of the training data.

It is unsurprising that TLC$_{norm}$ is the strongest performer for both tasks of Memotion 3, as this dataset contains many templatic memes and TLC was already a strong performer here. FigMemes continues to be a difficult dataset with a complex labeling scheme which requires fine-tuning to be performant. All versions of TLC continue to struggle with MAMI, which we attribute to the low ratio of templatic memes in this dataset.

A.7 Additional classification results

In this section, we provide additional results from our experiments that could not be put into the main text due to space limitations. Each table contains the results for a different type of modality or combination of modalities. Namely, we keep the modalities separate, we concatenate the embeddings, we fuse the embeddings via an element-wise product, or we normalize and average the embeddings. In each setting, we search over one to five neighbors as described in Section A.4. In the tables below, we present results organized by encoder, different CLIP models, namely `ViT-L/14@336px`, `ViT-B/32`, and `ViT-B/16`,[26] organized in each table in that order and also by the number of neighbors used for voting. The best configuration was chosen for Table 11 in the main text. Note that TLC$_{About/OCR}$ is only present in cases where the modalities are not combined, because in the other cases text embeddings are combined with the template or the meme embeddings.

We find that `ViT-L/14@336px` usually results in the strongest performer, consistent with our findings with TSplit, but there are exceptions. In the case of MultiOff and Memotion 3 (B) and Memotion 3 (A), for example, ViT-B/16 and ViT-B/32, respectively, were the best backbones for our method.

It is only in cases where we consider both templates and examples (TLC$_{Templates+Instances}$) that neighbor voting improves the final prediction. We believe that this is an intuitive finding for two reasons: (*i*) similar templates have unique, but broad semantics and convey concepts with related emotion charges, e.g., negative or positive sentiment. Therefore, templates that are similar would be nearby in the feature space. And (*ii*) template instances are many, conveying a specific meaning, and can be noisy or combinations of distinct templates, as we demonstrated in Section 4. This crowded and noisy feature space results in neigh-

[26]`https://github.com/openai/CLIP/blob/main/clip/clip.py`

Method	MultiOff	Memotion 3 (A)	Memotion 3 (B)	FigMemes	MAMI (A)	MAMI (B)
TLC$_{norm}$	37.92	**28.73**	**79.14**	9.79	43.24	**37.6**
TLC$_{maj}$	52.78	18.88	77.6	9.08	**52.92**	29.89
TLC$_{rand}$	**54.2**	24.25	78.43	**15.84**	48.66	28.91

Table 12: Classification results for the best-performing versions of TLC OOD and TLC$_{norm}$ (**in bold**) compared against each other.

bors that may be nearby markedly different templates.

We compute all evaluation measures using scikit-learn twice, where we set zero division equal to zero and to one, taking the max result between the two. We do this to avoid cases with zero in the denominator which can happen when precision (true positive + false positive) or recall (true positive + false negative) is equal to zero. This would make the f-score undefined. However, it is possible that this results in a sample-averaged F1 of 1.00 if we make no predictions for a given label, artificially inflating the weighted- or macro-averaged F1 score. In this case, we report the lower value.

In the earlier version of our work, we considered MEMEX in the main text, but removed it as it is no longer relevant to our analysis. However, we keep the results here for transparency.

Method	MultiOff	Memotion 3 (A)	Memotion 3 (B)	FigMemes	MEMEX	MAMI (A)	MAMI (B)
ViT-L/14@336px							
TLC$_{About/OCR}$ 1	44.43	27.12	76.58	21.14	46.02	60.43	35.19
TLC$_{Templates}$ 1	54.75	30.72	78.35	28.67	44.22	65.05	39.61
TLC$_{Templates+Instances}$ 1	58.58	34.59	76.91	27.99	43.01	67.44	38.92
TLC$_{Templates+Instances}$ 2	38.9	31.77	73.95	15.09	41.25	43.28	22.27
TLC$_{Templates+Instances}$ 3	43.56	32.15	74.49	18.54	41.11	51.51	26.05
TLC$_{Templates+Instances}$ 4	48.29	32.39	74.92	21.68	42.45	56.78	27.93
TLC$_{Templates+Instances}$ 5	45.66	33.0	75.65	23.05	43.87	60.89	32.73
ViT-B/32							
TLC$_{About/OCR}$ 1	48.29	27.06	77.6	20.86	43.56	58.7	33.8
TLC$_{Templates}$ 1	48.15	35.79	77.51	24.68	43.01	59.31	37.5
TLC$_{Templates+Instances}$ 1	48.33	28.68	76.74	28.4	43.64	63.68	38.35
TLC$_{Templates+Instances}$ 2	39.19	31.67	73.96	11.21	42.12	39.44	22.14
TLC$_{Templates+Instances}$ 3	40.94	32.63	75.13	15.44	42.12	45.71	23.87
TLC$_{Templates+Instances}$ 4	43.92	33.4	75.21	18.57	42.12	48.57	26.67
TLC$_{Templates+Instances}$ 5	43.2	34.1	75.81	21.04	42.12	52.3	29.74
ViT-B/16							
TLC$_{About/OCR}$ 1	51.83	35.4	76.2	20.29	46.25	59.04	35.44
TLC$_{Templates}$ 1	42.68	33.33	78.36	26.32	43.01	63.68	37.99
TLC$_{Templates+Instances}$ 1	51.32	36.42	78.13	26.87	43.71	63.31	37.34
TLC$_{Templates+Instances}$ 2	39.19	31.69	74.15	13.16	40.7	41.66	24.05
TLC$_{Templates+Instances}$ 3	39.99	51.58	74.56	15.95	42.25	48.43	27.21
TLC$_{Templates+Instances}$ 4	41.76	32.08	74.79	19.18	42.55	54.72	29.63
TLC$_{Templates+Instances}$ 5	42.3	32.45	75.11	22.27	42.55	56.98	32.23

Table 13: TLC classification results where the text and the image modalities are kept separate. The results are organized by encoder and the number of neighbors used for voting.

Method	MultiOff	Memotion 3 (A)	Memotion 3 (B)	FigMemes	MEMEX	MAMI (A)	MAMI (B)
ViT-L/14@336px							
TLC$_{Templates}$ 1	43.64	37.77	77.51	25.04	44.56	65.29	39.99
TLC$_{Templates+Instances}$ 1	45.29	28.5	78.6	23.81	41.07	69.09	38.07
TLC$_{Templates+Instances}$ 2	38.9	26.04	74.09	14.15	43.47	50.47	23.67
TLC$_{Templates+Instances}$ 3	43.2	27.07	75.57	18.24	42.88	54.58	27.92
TLC$_{Templates+Instances}$ 4	48.78	28.4	77.39	21.42	43.15	57.19	31.76
TLC$_{Templates+Instances}$ 5	48.74	29.18	77.36	23.53	43.33	60.4	34.74
ViT-B/32							
TLC$_{Templates}$ 1	52.56	27.62	76.35	26.59	44.84	60.42	37.87
TLC$_{Templates+Instances}$ 1	51.35	29.75	77.32	25.75	42.4	64.1	37.51
TLC$_{Templates+Instances}$ 2	41.61	33.02	74.95	12.99	40.7	46.44	23.86
TLC$_{Templates+Instances}$ 3	46.59	34.4	75.56	17.83	43.58	53.05	28.68
TLC$_{Templates+Instances}$ 4	52.24	34.45	76.25	19.59	42.38	57.71	31.84
TLC$_{Templates+Instances}$ 5	53.09	32.86	76.24	22.47	42.86	58.51	33.42
ViT-B/16							
TLC$_{Templates}$ 1	61.89	34.65	76.56	25.74	41.3	61.59	38.41
TLC$_{Templates+Instances}$ 1	53.98	35.76	78.65	23.65	43.77	62.33	37.09
TLC$_{Templates+Instances}$ 2	47.01	32.6	74.75	13.16	40.7	48.06	24.14
TLC$_{Templates+Instances}$ 3	49.07	33.44	76.06	18.48	43.37	53.84	29.29
TLC$_{Templates+Instances}$ 4	49.06	27.28	76.89	19.54	42.12	57.64	31.2
TLC$_{Templates+Instances}$ 5	50.83	35.28	77.62	20.8	43.0	59.78	33.17

Table 14: TLC classification results where the text and the image modalities are concatenated. The results are organized by encoder and the number of neighbors used for voting.

Method	MultiOff	Memotion 3 (A)	Memotion 3 (B)	FigMemes	MEMEX	MAMI (A)	MAMI (B)
ViT-L/14@336px							
TLC$_{Templates}$ 1	43.13	30.56	79.89	19.44	44.99	54.06	33.43
TLC$_{Templates+Instances}$ 1	51.26	36.48	80.17	18.76	48.14	57.99	35.4
TLC$_{Templates+Instances}$ 2	43.92	32.63	74.78	25.76	41.05	39.91	22.27
TLC$_{Templates+Instances}$ 3	38.71	33.2	75.63	13.02	40.9	45.13	23.82
TLC$_{Templates+Instances}$ 4	45.46	33.65	77.22	13.14	40.86	48.21	26.38
TLC$_{Templates+Instances}$ 5	45.32	35.93	77.06	15.24	41.1	48.2	27.89
ViT-B/32							
TLC$_{Templates}$ 1	49.68	26.88	78.62	21.37	42.97	60.68	31.73
TLC$_{Templates+Instances}$ 1	52.83	27.36	78.05	18.46	44.6	53.11	32.84
TLC$_{Templates+Instances}$ 2	41.57	32.26	74.59	41.83	41.05	38.81	20.13
TLC$_{Templates+Instances}$ 3	42.29	29.95	75.32	27.24	41.05	42.97	22.96
TLC$_{Templates+Instances}$ 4	45.85	30.74	75.04	28.97	41.5	45.77	25.36
TLC$_{Templates+Instances}$ 5	45.25	33.82	74.82	14.72	43.66	46.01	26.24
ViT-B/16							
TLC$_{Templates}$ 1	49.29	29.56	77.35	19.57	43.47	56.3	32.81
TLC$_{Templates+Instances}$ 1	50.09	29.52	80.49	19.64	43.49	54.18	33.51
TLC$_{Templates+Instances}$ 2	44.65	25.71	74.48	41.26	40.7	36.84	20.67
TLC$_{Templates+Instances}$ 3	48.14	32.56	75.89	9.84	40.7	41.12	22.95
TLC$_{Templates+Instances}$ 4	50.02	34.14	76.36	12.0	41.52	46.16	24.85
TLC$_{Templates+Instances}$ 5	47.44	33.78	75.96	14.44	42.12	47.78	26.11

Table 15: TLC classification results where the text and the image modalities are fused via the Hadamard product. The results are organized by encoder and the number of neighbors used for voting.

Method	MultiOff	Memotion 3 (A)	Memotion 3 (B)	FigMemes	MEMEX	MAMI (A)	MAMI (B)
ViT-L/14@336px							
TLC$_{Templates}$ 1	48.72	27.81	76.88	29.8	44.4	62.7	37.77
TLC$_{Templates+Instances}$ 1	52.89	37.04	77.58	25.5	46.01	63.01	36.57
TLC$_{Templates+Instances}$ 2	46.48	33.06	75.84	16.43	44.21	51.55	27.12
TLC$_{Templates+Instances}$ 3	40.96	26.11	75.96	18.78	45.2	58.25	30.93
TLC$_{Templates+Instances}$ 4	46.07	26.63	75.5	22.08	45.52	61.23	32.41
TLC$_{Templates+Instances}$ 5	47.9	25.99	77.13	23.45	45.36	62.79	33.83
ViT-B/32							
TLC$_{Templates}$ 1	57.09	33.55	78.04	25.07	42.45	60.65	35.95
TLC$_{Templates+Instances}$ 1	49.07	35.22	77.75	23.36	43.99	63.21	36.96
TLC$_{Templates+Instances}$ 2	43.2	32.18	74.07	13.71	41.1	50.66	28.94
TLC$_{Templates+Instances}$ 3	42.12	32.55	75.27	16.76	42.15	56.41	28.41
TLC$_{Templates+Instances}$ 4	41.71	25.7	75.96	19.2	42.31	59.1	32.54
TLC$_{Templates+Instances}$ 5	44.02	25.37	76.46	20.08	42.93	63.12	33.12
ViT-B/16							
TLC$_{Templates}$ 1	47.54	34.57	75.15	24.39	42.6	64.43	38.72
TLC$_{Templates+Instances}$ 1	47.7	27.46	78.59	24.04	42.82	61.49	35.41
TLC$_{Templates+Instances}$ 2	50.02	33.25	74.88	13.41	43.84	51.08	24.79
TLC$_{Templates+Instances}$ 3	44.36	32.12	76.03	18.93	43.97	56.67	28.13
TLC$_{Templates+Instances}$ 4	49.19	25.41	76.11	20.61	44.34	58.51	30.04
TLC$_{Templates+Instances}$ 5	49.22	33.77	77.29	22.0	44.34	59.34	32.06

Table 16: TLC classification results where the text and the image modalities are normalized and averaged. The results are organized by encoder and the number of neighbors used for voting.

Method	MultiOff	Memotion 3 (A)	Memotion 3 (B)	FigMemes	MAMI (A)	MAMI (B)
TLC$_{best}$	61.89	37.77	80.49	29.8	70.0	40.21
TLC$_{TSplit}$ 1	37.12	22.8	70.93	5.96	29.5	18.08
TLC$_{TSplit}$ 2	49.27	29.7	72.84	15.86	63.82	28.54
TLC$_{TSplit}$ 3	45.05	**32.08**	76.75	21.46	69.12	32.99
TLC$_{TSplit}$ 4	**51.81**	28.8	74.78	28.67	75.2	**37.53**
TLC$_{TSplit}$ 5	48.7	31.62	**77.39**	**29.55**	**75.73**	37.35

Table 17: TLC classification results after TSplit preprocessing (the highest result is in **bold**).

A.8 TSplit then TLC

In this Appendix section, we present TLC results after preprocessing via TSplit (see Table 17). We searched over one to five neighbors, using only the base template. We took a tolerant view of what constitutes a template instance, using TSplit$_{max}$.

TLC assumes all memes are templatic and relies on the template signal, assuming that the most common label for a given template in the training data is the correct label for an instance of that template in the test data. However, TSplit ensures that templates do not overlap between dataset splits, effectively removing TLC's predictive power. Under these conditions, TLC is far more likely to assign the wrong template and therefore likely the wrong label.

When a dataset contains many templatic memes, such as Multioff or Memotion 3, we see that TLC's performance drops markedly, even if we consider additional base templates (additional neighbors). However, when a dataset does not contain as many templatic memes, MAMI, or is very difficult, FigMemes, the effect on performance is not as dramatic, and voting with multiple templates allows us to achieve similar if not improved performance relative to the best performing version of TLC from Table 11. In the case of MAMI, this is because TLC alone was already assigning templates erroneously most of the time. For FigMemes, this is due to the difficulty of the task, where assigning a fixed interpretation for a given template is too simple an approach for a nuanced labeling system (see Section A.13) and also a nuanced means of communication like memes.

We searched over one to five neighbors and include the best TLC results from Table 11 for reference. Performance suffers when a dataset is highly templatic, while it is not as affected or even improves when the dataset does not contain as many templatic memes. `ViT-L/14@336px` was used as the CLIP encoder.

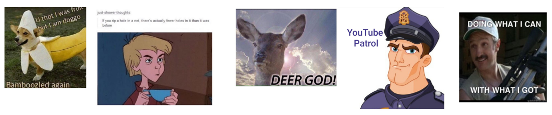

U thot I was fruit but I am doggo
Bamboozled again
just-shower-thoughts:
If you rip a hole in a net, there's actually fewer holes in it than it was before
DEER GOD!
YouTube Patrol
DOING WHAT I CAN
WITH WHAT I GOT

Figure 6: Examples of non-template images found in FigMemes.

A.9 Non-Templatic Memes

In this Appendix section, we provide examples of images/memes which we consider to be non-templatic (see Figure 6). The first and third examples are a visual joke and pun respectively. The first does make reference to the *Doggo*[27] and language from the *Cheezburger*[28] templates. The second is a still from a .gif that references the film *The Sword in the Stone*[29] and can actually be found on KYM.

This is a bit of an edgecase, but we believe this communicates the idea of confusion or realization triggered by the text above the image, which is interpretable without knowledge of a template. The fourth example is arguably not a meme and we actually are not sure of the interpretation without additional context. A possible reading, given the domain of FigMemes, is a criticism of users who comment on YouTube, forcing their views of social norms on the politically incorrect, but this a forced interpretation. The fifth example is a still from the movie *Tremors II: Aftershock*, where the image is the correct character but the text is quoted anachronistically from another part of the film. All information relevant to the scraping process is preserved in a .json file, linking templates to their examples (see Figure 7).

A.10 Scraping details

We use WM for all scraping and use the most recent snapshots available. We first visit the main page of KYM on WM with Selenium[30] and autoscroll through the entire meme table. We collect and write .htmls files for each cell in the template table. We then process the .html files, parsing each meme template and collecting the text appearing in the table with Beautiful Soup.[31] We collect all urls seen and write them to a .json file. This ensures that we do not revisit seen pages if the scraper is

[27]https://knowyourmeme.com/memes/doggo

[28]https://knowyourmeme.com/memes/sites/cheezburger

[29]https://en.wikipedia.org/wiki/The_Sword_in_the_Stone

[30]https://selenium-python.readthedocs.io/

[31]https://beautiful-soup-4.readthedocs.io/en/latest/

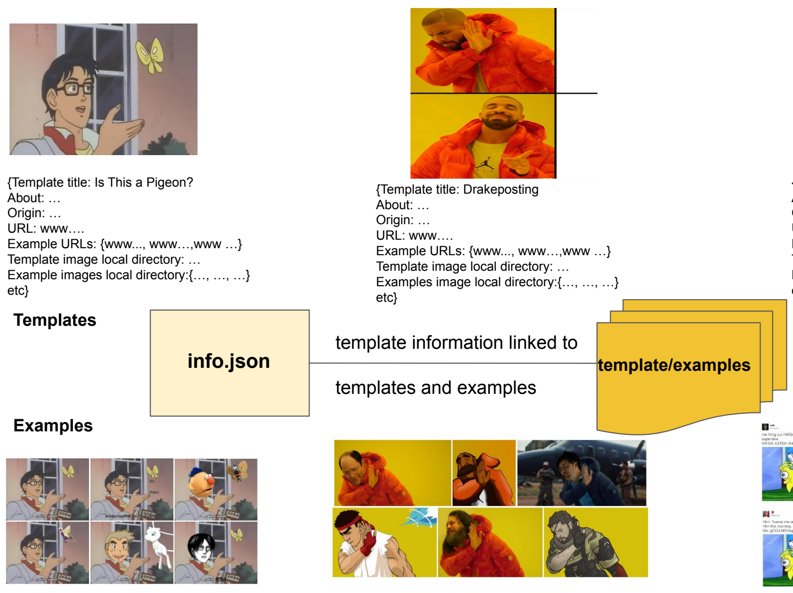

{Template title: Is This a Pigeon?
About: …
Origin: …
URL: www.…
Example URLs: {www..., www…,www …}
Template image local directory: …
Example images local directory:{…, …, …}
etc}
{Template title: Drakeposting
About: …
Origin: …
URL: www.…
Example URLs: {www..., www…,www …}
Template image local directory: …
Examples image local directory:{…, …, …}
etc}
{Template title: Mocking SpongeBob
About: …
Origin: …
URL: www.…
Example URLs: {www..., www…,www …}
Template local directory: …
Examples local directory:{…, …, …}
etc}
Templates
info.json
template information linked to
templates and examples
template/examples
Examples

Figure 7: The KYMKB records all textual information about a meme in a .json file, including the text found on KYM, the URLs used in the scraping process, and local locations of all template and example images in the knowledge base.

interrupted and it very likely will be. To collect template instances (examples), we look at each image url and determine if WM has a snapshot available and collect the image and write to disk if it exists. Again, we record all visited urls to avoid revisiting and associate each image with its parent template see Figure 7.

WM's snapshots of the Internet are incomplete, making it impossible to completely capture KYM via WM; of the roughly 8,400 confirmed entries at the time of writing, we were only able to scrape 5,220. However, we are passionate about memes and we are devoted to making the KYMKB as complete as possible. We therefore release all our scraping code such that other members of the community can contribute to and improve meme analysis grounded in templates.

A.11 More template-meme analysis

In this Appendix section, we showcase additional examples of how the KYMKB can be easily used with simple, well-known algorithms, such as nearest neighbor indexing and k-means clustering to gain insight into a meme dataset. Specifically, we can use retrieval to examine how well templates in our knowledge base map onto memes in a dataset. Often we find that the memes in a dataset are simply the base templates or examples already contained in the knowledge base. We argue that examination of cluster centroids yields insight into which templates best reflect the type of memes in a dataset. For example, FigMemes was collected from 4chan /pol/, and by investigating cluster centroids we unintentionally arrived at the /pol/ template. We emphasize that we did nothing but consider the template closest to a centroid and arrive at a template we ourselves were previously unaware of. Details can be found below.

Retrieval Here we provide further examples and details regarding the retrieval-based examination of the KYMKB from the main text, Section 4. After querying the 500 closest neighbors, we then randomly select k pairs, where k is equal to the number of labels in a given dataset. The pairs, as in the main text, are composed of the template and its nearest neighbor in the dataset. For conciseness, we only consider FigMemes here as it is a difficult dataset with the most labels.

Figure 8 shows a sample of our findings. Combining embeddings via fusion or normalizing and averaging the vectors results in matches where the relation between a template and a meme is nuanced or nonexistent.

We again find that either only considering the image modality or concatenating the image and the text representations results in the strongest signal, and indeed, using this configuration for retrieval makes it difficult to appear as though we are not cherry-picking. We clearly match either a base template to a meme or a base template to an obvious instance of that template. In cases when it is not

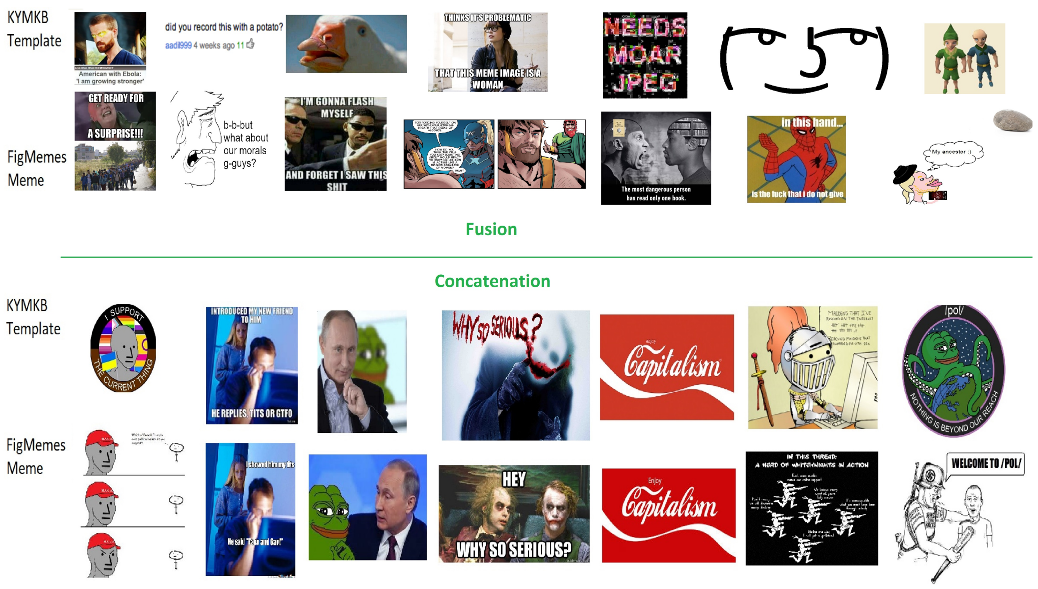

KYMKB Template
FigMemes Meme
Fusion
Concatenation
KYMKB Template
FigMemes Meme
did you record this with a potato?
THINKS IT'S PROBLEMATIC
THAT THIS MEME IMAGE IS A WOMAN
NEEDS MOAR JPEG
GET READY FOR
A SURPRISE!!!
b-b-but what about our morals g-guys?
I'M GONNA FLASH MYSELF
AND FORGET I SAW THIS SHIT
The most dangerous person has read only one book.
in this hand...
is the fuck that i do not give
I SUPPORT THE CURRENT THING
INTRODUCED MY NEW FRIEND TO HIM
HE REPLIES ITS OR GTFO
WHY SO SERIOUS?
HEY
WHY SO SERIOUS?
Capitalism
Enjoy Capitalism
IN THIS THREAD: A HERD OF WHITEKNIGHTS IN ACTION
/pol/
NOTHING IS BEYOND OUR REACH
WELCOME TO /POL/

Figure 8: KYMKB templates matched via similarity search to FigMemes images.

so obvious, we match text or characters, such as *Why So Serious* or the *Joker*,[32] or concepts that exist in only meme or Internet culture. For example, consider the first column under the concatenation setting in Figure 8. We observe the character of Wojak in the *I Support the Current Thing* meme template,[33] [34] a template that criticizes social media users for being a simpleton or lacking critical thinking skills. We match this template to a meme criticizing Trump supporters for the same faults, despite drastically different appearances. In the sixth column, we match the template of White Knight to an image that derides *White Knighting*.[35] This template and its entry in the KYMKB provide sufficient background to interpret the FigMemes image, which is arguably not even a meme. Finally, in the seventh column, we match the template of /pol/ to a meme obviously about the 4chan board.[36] We share this information not to explain memes, but to demonstrate the ease and the power of using the KYMKB to retrieve information about not only memes, but also images related to Internet culture. If one is not familiar with these concepts, it is difficult to even know what to search for; however, this is different with KYMKB.

Clustering In order to investigate the saliency of templatic memes in the context of meme datasets, we conduct distance-based clustering using KMeans where we fit the algorithm on both the KYMKB, with or without examples, and on the dataset in question, encoding all memes using CLIP. We then manually examine the closest meme or template to each centroid, respectively. We set k to be equal to the number of labels in each dataset (see Table 18). Here, for conciseness, we consider only templates and FigMemes, as we consider it a difficult dataset and it has the most labels/.

Figures 9 and 10 show a sample of our results. If we attempt to combine the image and the text embeddings, either via fusion or normalization and averaging, we find that this often results in repeated images, that is, a meme or a template is close to multiple centroids. However, if we concatenate the embeddings or only use the images representation, we find that we are left with centroids that point to k distinct image files, where k is again equal to the number of labels in a given dataset: seven in the case of FigMemes.

Naturally, when we consider centroids fit on KYMKB, their closest meme in FigMemes reflects the nature of that dataset. These memes express sexist or politically charged, but still toxic rhetoric, which 4chan /pol/ is known for. Somewhat surprisingly, when we determine the centroids from the dataset and query the closest template in the KYMKB, we again see the nature of the dataset reflected, where we had expected to be met with

[32]Note that this text and character have taken on lives on their own in meme culture. https://knowyourmeme.com/memes/why-so-serious

[33]https://knowyourmeme.com/memes/npc-wojak

[34]https://knowyourmeme.com/memes/i-support-the-current-thing

[35]https://knowyourmeme.com/memes/white-knight

[36]https://knowyourmeme.com/memes/sites/pol

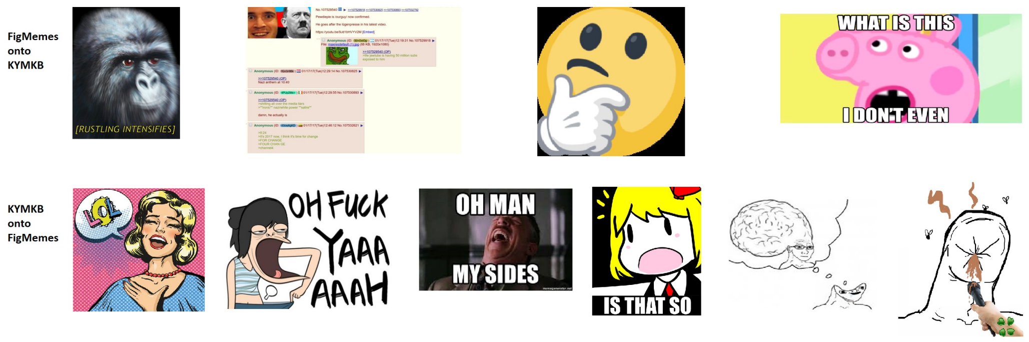


Figure 9: In the first row, we show the templates closest to seven KMeans centroids fit on the FigMemes, while in the second row, we show FigMeme images closest to seven centroids derived from KYMKB. We combine the text and the image representations by normalizing and averaging the two modalities. This results in multiple centroids close to the same meme/template.

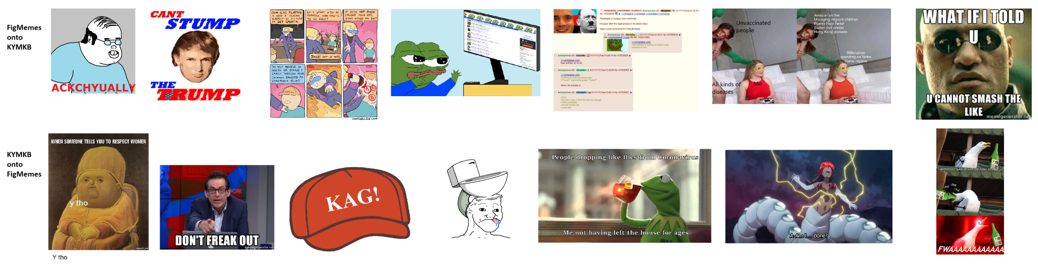


Figure 10: In the first row, we show the templates closest to seven KMeans centroids fit on the FigMemes, while in the second row, we show FigMeme images closest to seven centroids derived from the KYMKB. We only use the image modality, which results in seven distinct images.

potentially political, but not toxic templates. The resulting image files express salient traits of derision, sexism, or conservative political beliefs. Interestingly, if we combine modalities or only consider image representations, one meme centroid is closest to the same template in both cases, that is the *Is He /Our Guy/?* template.[37] This 4chan-specific template is used to confirm whether a celebrity shares similar beliefs as the "politically incorrect" community, e.g., supporting Nazism. It is surprising that an examination of centroids in this way provides such a succinct summary of the domain of the dataset.

Annotation In this Appendix section, we provide additional details on the annotation conducted in Section 4. Our annotators were two colleagues of ours that are knowledgeable about memes. They were warned that the template-meme pairs might be offensive.

A.12 Meme "edge cases"

Below, we provide a discussion and background on examples of meme templates contained in the KYMKB that defy the narrow scope of memes being static images. The templates we discuss are by no means exhaustive and we provide this section purely as additional motivation for our argument that the AI community must not limit itself simply to static images.

One of the oldest templates is *Rickroll*,[38] which can involve posting an image of Rick Astley from the Never Going to Give You Up music video,[39] but more frequently an instance of this template is a bait-and-switch prank where posters trick others into viewing the music video. This has since evolved where the prank is now to trick others into stating the title of the song.[40] We would argue this is an intertextual meme instance referencing the Rickroll template.

Loss[41] is another famous template where an instance is an action, not an image. The template is a reference to the Ctrl+Alt+Del Comic[42] gaming webcomic, which made an uncharacteristically

[37]https://knowyourmeme.com/memes/is-he-our-guy

[38]https://knowyourmeme.com/memes/rickroll

[39]https://www.youtube.com/watch?v=dQw4w9WgXcQ

[40]https://knowyourmeme.com/photos/1901413-rickroll

[41]https://knowyourmeme.com/memes/loss

[42]https://cad-comic.com/

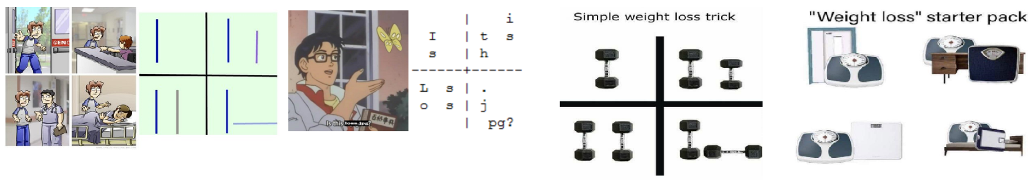


Figure 11: The first image is the original template of *Loss*, while the other three images are *Loss* instances, all of which are visual puns that cannot be understood without knowing the original template. The second image is another intertextual meme where *Loss* and *Is This a Pigeon?* have been amalgamated.

serious update about a miscarriage. The idea that this webcomic could approach such a serious topic amused many social media users, and they began mocking the strip by posting references to the panel as a joke, bringing it to its meme status. The strip was referenced so ubiquitously that the positions of the characters in the strip, that is, one vertical line, two vertical lines of different heights, two vertical lines of the same height, and one vertical and one horizontal line became an instance of this template. The phrase *Is this Loss?* became a meme by itself, as users wondered whether certain posts or memes were instances of the *Loss* template (see Figure 11).

Instances of the *Planking*[43] template is again a behavior where a person lies flat on their stomach with their arms to their sides in an unusual place, has their photo taken, and uploads this for the amusement of others.

Another tricky template is that of *Thinking Face Emoji*.[44] An instance of this template would be ironically or sarcastically posting a thinking face emoji. However, this could be simply using the Unicode "U+1F914" or posting a picture of the emoticon for extra emphasis.

A recent example of a meme that is not an image is the *OOF / Roblox Death Sound* template.[45] An instance of this template is featuring or remixing the audio clip in videos or music, referencing an amusing sound effect from the popular MMORPG Roblox.[46] Players of this game found the audio clip so amusing that it is referenced to suggest humorously express empathy for another's misfortune and shared experience.

A.13 Datasets and previous work details

In this Appendix section, we provide additional information about the datasets we examined, such as their respective label inventories, distributions, and reported inter-annotator agreement scores. We also provide information on the models reported in the respective work. We do this for ease of reference and simply reproduce reported information where possible. When this information is not available, we report the information we are able to access.

MultiOff is a binary classification task, offensive (40%) vs. not offensive (60%), composed of memes related to the 2016 US Presidential Election. They report two Fleiss Kappas both before and after getting feedback from their annotators. The first is between 0.2 and 0.3 (fair agreement), while the other, after feedback, is between 0.4 and 0.5 (moderate agreement). Their text-based model is a combination of GloVe embeddings (Pennington et al., 2014) and a CNN, while their image-based model was VGG 16 (Simonyan and Zisserman, 2015) pre-trained on ImageNet (Deng et al., 2009), and their multimodal method consistent of a stacked LSTM (text) and VGG 16 (image) combined via early fusion.

Memotion 3 is composed of two multilabel tasks (A and B). The test split is not publicly available, so we consider only the training and validation split. Task A is sentiment analysis for memes, where labels can be very positive (5%), positive (26%), neutral (42%), negative (23%), or very negative (5%). Task B considers memes with humorous (39%), sarcastic (37%), offensive (19%), and motivational (5%) messages. They do not report inter-annotator agreement scores, settling disagreements via majority vote. This work reports only multimodal results, fusing and fine-tuning a BERT-based Hindi and English model (Bhange and Kasliwal, 2020) for textual features and the ViT model (Dosovitskiy et al., 2021) for image features. Note that their test split is not public at time of writing.

FigMemes is a multilabel task of determining the type of figurative language used in a meme. There are seven labels, composed of Allusion (17%), Exaggeration (19%), Irony (20%), Anthropomor-

[43] https://knowyourmeme.com/memes/planking
[44] https://knowyourmeme.com/memes/thinking-face-emoji
[45] https://knowyourmeme.com/memes/oof-roblox-death-sound
[46] https://www.roblox.com/

Dataset	Task	Number of Labels	Size	Multilabel?	Multilingual?	Evaluation Measure
FigMemes	Figurative Language	7	5141	Yes	No	Macro-F1
MultiOff	Offensive Language	2	743	No	No	Macro-F1
MAMI Task A	Misogyny Detection	2	11k	No	No	Macro-F1
MAMI Task B	Types of Misogyny	4	11k	Yes	No	Weighted-F1
Memotion 3 Task A	Sentiment Analysis	3	10k	No	Yes	Weighted-F1
Memotion 3 Task B	Types of Emotion	4	10k	Yes	Yes	Weighted-F1

Table 18: Summary of the previous work we examine.

Dataset	Text Model	Vision Model	Multimodal Model	Agreement
FigMemes	DeBERTa	CLIP	CLIP	0.42
MultiOff	GloVe + CNN	VGG 16	Stacked LSTM + GloVe (text) / VGG 16 (image) (early fusion)	0.2 - 0.3 to 0.4 - 0.5
MAMI Task A	NA	NA	Ensemble of XGBoost + CLIP + UNITER + BERT	0.5767
MAMI Task B	NA	NA	Ensemble of XGBoost + CLIP + UNITER + BERT	0.3373
Memotion 3 Task A	NA	NA	Hinglish BERT (text) / ViT (image)	Majority vote
Memotion 3 Task B	NA	NA	Hinglish BERT (text) / ViT (image)	Majority vote

Table 19: Continued summary of the previous work we examine. In the case of multimodal models, we provide them as *text model* / *vision model*. For agreement, we provide multiple scores to indicate that the researchers consulted their annotators, which led to an increase in agreement.

phism (9%), Metaphor (20%), Contrast (10%), and None (30%) (see the work for more information). They report a Fleiss Kappa of 0.42, indicating moderate agreement. The authors fine-tuned DeBERTa (He et al., 2021) for their text classifier and used various CLIP fine-tuning strategies for their image and multimodal experiments.

Task A in MAMI looks at whether memes are misogynous or not. The task has a balanced binary label distribution and the authors report a Fleiss-k of 0.5767. Task B examines different types of misogyny expressed in a meme. There are four labels, Shaming (17%), Stereotype (38%), Objectification (31%), Violence (13%), and the remaining do not express misogyny. The authors report a Fleiss-k of 0.3373, showing that is too is quite a difficult task. The best performing methods on this dataset are reported in Zhang and Wang (2022), which involved multimodal fine-tuning and ensembling of XGBoost (Chen and Guestrin, 2016), CLIP, UNITER (Chen et al., 2020), and BERT.

See Tables 18 and 19 above for a summary of this information.

A.14 Prompting with LLMs

We now explore if the KYMKB can also be used to aid a vision language model by grounding the model in a meme-specific context. We experiment with LLaVA (Liu et al., 2023) which employs LLaMA (Touvron et al., 2023) and CLIP to combine both the textual and visual input modalities.

We mainly conduct few-shot in-context-learning (ICL) experiments using one completion and three examples for text-only modality inputs, as the model has not been trained to handle multiple input images. We investigate the following scenarios, with and without RAG-style (Lewis et al., 2020) prompting:

- Which input modality is the most useful? Text, image, or both?
- Does providing a description of the labels in the prompt help? (i.e the meaning of "anthropomorphic" in FigMemes)
- Does performance increase if we retrieve the nearest meme template title and add it to the prompt? (i.e *Drakeposting*,[47] *Is This a Pigeon?*, etc.)
- Does LLM performance increase when we consider all examples of a template and not just the base template as candidates for retrieval?
- Does it help to discard retrieved meme information from the KYMKB if a given input meme is too different (not considered in distribution) from its closest entry in the KYMKB?
- Does the inclusion of the retrieved template *about* section in the prompt improve LLM performance?

[47] https://knowyourmeme.com/memes/drakeposting

We perform an extensive ablations to answer the above questions and the main results are shown in Table 20. Our overall setup is illustrated in Figure 12. We highlight the following results by answering the points raised above when grounding LLaVA in the KYMKB:

- Model performance is higher when we exclude the input image but use the extracted OCR on the input meme.
- Adding the nearest template title improves performance.
- Adding an explanation of target labels hurts performance.
- Excluding information from retrieved KYMKB entities for input memes that are too dissimilar improves performance. Similar to TSplit, dissimilar here means exceeds the threshold value from template to example in the KYMKB.
- Including the *About* section of the nearest neighbor in the KYMKB as input to LLaVA aids in understanding.

Parsing LLM output To calculate the F1 score from the LLM output and ground truth labels, we convert the output and ground truth into binary arrays with n elements with n being the number of classes for the given multilabel task. For multiclass and binary classification tasks, we instead operate with $n - 1$ labels. To ensure the model does not start to regurgitate the answer list and input, we restrict decoding to 64 tokens. Similarly, we also ask the model to output the answer "none" if no category is suitable and that "The Assistant must only answer by listing the labels that describe the meme and nothing else.". Finally, if the model does output all the labels, an erroneous answer is given, or no output is recorded, we opt for returning a list of n elements each being 0.

Examples of LLaVA output To illustrate the content of the output, we show three examples of LLaVA output on the FigMemes dataset below:

- "the meme follows the label set of allusion, exaggeration, and irony. it is not clear which of these labels best fit the ""who killed hannibal?"" meme in this specific image."
- "the meme ""distracted boyfriend"" can be classified as an example of anthrop, metaphor, and contrast. in this meme, the man in the image represents the ""distracted boyfriend"" metaphorically, as he appears to be looking at another woman rather than pay"
- "the meme follows the label set of allusion, exaggeration, and irony. the labels "i once was blind but now i see " seem to be an allusion to the popular song "i used to be blind""

Discussion Table 20 shows our results. In the case of MultiOff and Memotion 3 (B), we see that grounding our LLaVA model in the KYMKB improves performance and that the textual modality alone is better than including visual inputs. We believe this is because the offensive or emotional charge signal is stronger in the text. This finding is consistent with (Aggarwal et al., 2024), who showed that text alone is enough to detect hate speech in memes. Additional meme-context from the KYMKB can naturally aid in this. For Memotion 3 (A) and FigMemes on the other hand are more challenging tasks. The model struggles, even with the aid of the KYMKB. This is consistent with our findings for TSplit. In general, LLaVA performs worse than TSplit, thus motivating further study into learning more robust models.

The only exception to the overall trend is MAMI (A) which we believe is due to the LLaMA's strong understanding of misogyny in text. KYMKB is inherently not a knowledge base of misogyny and not a suitable resource for such a message and for the non-templatic memes in MAMI.

A.14.1 Classification Experiments

In this section, we investigate several ablations including how to format the prompt. We examine the following:

- How do different input modalities impact downstream tasks?
- Should there be a threshold mechanism for selecting which memes are relevant for meme retrieval?
- Does including more information from KYM help downstream tasks and do multiple examples help ICL?

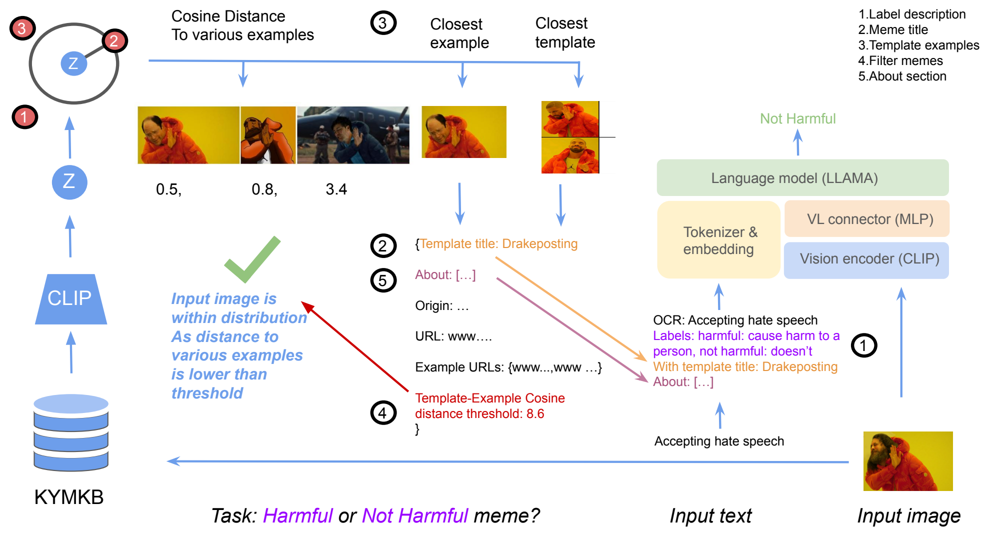

Cosine Distance
To various examples
Closest example
Closest template
1.Label description
2.Meme title
3.Template examples
4.Filter memes
5.About section
Z
Z
CLIP
KYMKB
0.5,
0.8,
3.4
Not Harmful
Language model (LLAMA)
Tokenizer & embedding
VL connector (MLP)
Vision encoder (CLIP)
Input image is within distribution As distance to various examples is lower than threshold
{Template title: Drakeposting
About: [...]
Origin: ...
URL: www....
Example URLs: {www...,www ...}
Template-Example Cosine distance threshold: 8.6
}
OCR: Accepting hate speech
Labels: harmful: cause harm to a person, not harmful: doesn't
With template title: Drakeposting
About: [...]
Accepting hate speech
Task: Harmful or Not Harmful meme?
Input text
Input image

Figure 12: The complete setup of using LLAVA with KYMKB. All experiments are with multiple examples as illustrated in Figure 21. Initially, both the OCR text and input meme are passed to LLaVA alongside k examples with ground truth labels and a selection of labels to choose from. Then we add 5 steps with further modifications as shown in the top right corner of this figure and according to the additional steps in Table 20. In Step 1, we add a description of each label as a prefix to LLaVA, the 2nd step is RAG in that we look up the nearest neighbor with CLIP in the KYMKB and retrieve its meme template name, and add this to the text prompt. As a 3rd step, we increase the number of entries in the KYMKB by adding each template example, increasing the coverage of memes we want to look up. We then filter out retrieved results that are larger than the allowed threshold for each meme template. This is measured by comparing the maximum distance between the base template and examples and the distance between the input meme and the base template. In the fifth and final step, we include the retrieved *about* section in the prompt too.

A.14.2 Prompt ablations

Before investigating how LLAVA (Liu et al., 2023) might best perform in prompting experiments on memes, we will describe the overall prompting setup. To make LLaVA suitable for different tasks on memes without fine-tuning, we perform few-shot in context learning with 3 randomly drawn examples and ground truth answers. These are given to the model when inferring the answer for a given input. The process is illustrated in Table 21 However the model struggles with multiple images, even when training and evaluated on the same domain. As such we restrict ourselves to providing the few-shot examples as text input only.

We write a prompt that instructs the model to choose between one or multiple choices from a list of possible answers, which are the labels for a given classification task.

A.14.3 Non-retrieval ablations

In this section, we focus our attention on whether both image and OCR text or if either one would suffice in helping LLaVA in its downstream classification task. We investigate model performance without retrieving external knowledge by prompting the model with either 1) the OCR text (OR), 2) the original image (IM), or 3) both. We also include explanations of the target labels (LD) to examine if this improves performance. We report the micro-averaged F1 for each class per task and the average and weighted F1 micro score for all classes. We examine MultiOff, FigMemes, and both MAMI tasks.

Method	All.	Exa.	Iro.	Ant.	Met.	Con.	F1	F1(W)
OC	22.7	25.2	29.9	11.5	18.2	12.4	22.8	21.6
OC+LD	26.4	24.1	31.6	14.2	5.4	16.3	23.0	20.6
IM	20.0	29.2	27.7	13.4	17.5	16.3	22.5	21.9
IM+LD	20.2	30.7	26.2	12.9	21.8	18.0	23.2	22.9
IM+OC	21.1	30.8	32.7	15.6	17.4	17.1	25.6	23.8
IM+OC+LD	22.4	25.6	28.1	16.5	11.0	19.5	21.1	21.1

Table 22: LLaVA performance in terms of F1 scores on FigMemes. "OC" refers to the text in the meme, "IM" is the meme itself, "ID" refers to our predicted meme template title, "LD" is the label description, i.e a description of what each label means. The different categories are All(usion), Exa(ggeration), Iro(ny), Ant(hropopomorphism), Met(aphor) and Con(trast).

Method	**MultiOff**	**Memotion 3 (A)**	**Memotion 3 (B)**	**FigMemes**	**MAMI (A)**	**MAMI (B)**
ICL: Vision and text	41.6	26.8	73.8	25.6	48.9	35.1
ICL: Vision only	43	26.6	72.1	22.5	47.6	33.8
ICL: text only	49	27.3	71.3	22.8	47.7	34.2
Using text only						
+ Label description	45	27.7	72.4	23	49.4	33.8
+ Template title	49	27.1	74.1	22.3	51.4	36.6
+ Template examples	47	29.7	73.4	23	48.6	34.4
+ Filter OOD memes	53.7	30.5	74.8	24.7	53.7	36.4
+ *About* section	**56.4**	**31.1**	**75.7**	**25.4**	**56.8**	**37.3**

Table 20: Classification results for the best performing version of LLaVA grounded in the KYMKB (**in bold**) compared against the best-performing method from related work (*in italics*). + refers to adding additional input information to the previous row (i.e + template title also includes giving LLaVA the label description in addition to the template title). See remaining tables for full ablations.

	Few-shot ICL
Shot	You are given the following memes with input Optical Character Recognition text: `[Text]` Input Image: `[Image]` and the following labels `[Label]` with explanation `[Label Detail]` The meme can be described as `[GT Labels]`
Shot	The next meme has the following input ...
Input	The final meme has the following input OCR: `[Text]`
Answer	The meme can be described as* `<answer>`

Table 21: Prompt setup for zero-shot and few-shot prompting with LLaVA

Method	sh.	st.	ob.	vi.	F1.	F1(W)
OC	24.3	44.4	39.6	7.0	34.2	34.1
OC+LD	24.4	45.6	35.7	9.7	33.8	33.5
IM	22.2	37.9	43.2	21.1	33.8	34.8
IM+LD	23.9	42.9	42.4	18.9	34.5	36.3
IM+OC	22.4	41.2	43.9	15.9	35.1	35.5
IM+OC+LD	27.8	42.5	46.4	19.3	37.0	38.1

Table 23: LLaVA performance in terms of F1 scores on MAMI (B). "OC" refers to the text on the meme, "IM" refers to usage of the image itself, "ID" refers to our predicted meme template title, "LD" refers to label description, i.e a description of what each label means. The categories to predict are Sh(aming), St(ereotype), Ob(jectification) and Vi(olence). We measure the F1 score and the Weighted F1 score, F1(W)

Method	Of.	N-Of.	F1.	F1(W)
OC	38.7	56.3	49.0	45.6
OC+LD	19.6	58.2	45.0	34.6
IM	17.5	56.4	43.0	36.9
IM+LD	2.1	51.2	34.9	21.2
IM+OC	10.3	56.7	41.6	28.4
IM+OC+LD	8.2	55.0	39.6	26.4

Table 24: LLaVA performance in terms of F1 scores on MultiOff. "OC" refers to the text on the meme, "IM" refers to usage of the image itself, "ID" refers to our predicted meme template title, "LD" refers to label description, i.e a description of what each label means. The two categories to predict are Of(fensive) and Non-offensive (N-Of)

A.14.4 Clip-based template retrieval

As an additional ablation, we test if increasing the size of the CLIP model used for retrieval affects the performance of LLaVA. This model uses a 14x14 patch size on a downscale image of resolution 336x336,

Method	All.	Exa.	Iro.	Ant.	Met.	Con.	F1	F1(W)
OC+ID+LD	27.3	24.5	30.2	13.6	5.3	12.8	22.3	20.1
IM+ID+LD	21.3	27.9	26.7	16.6	8.4	19.7	21.2	20.5
IM+OC+ID+LD	20.2	25.7	26.8	18.5	10.6	16.7	20.8	20.2

Table 25: LLaVA performance in terms of F1 scores on FigMemes. "OC" refers to the text in the meme, "IM" refers to usage of the image itself, "ID" refers to our predicted meme template title, "LD" refers to label description, i.e a description of what each label means.

Method	sh.	st.	ob.	vi.	R	F1(W)
OC+ID+LD	24.1	47.7	42.5	13.0	36.6	37.1
IM+ID+LD	23.1	42.7	45.4	20.5	35.4	37.4
IM+OC+ID+LD	22.8	38.6	46.2	22.4	34.7	36.5

Table 26: LLaVA performance in terms of F1 scores on MAMI (B)

Method	Of.	N-Of.	F1.	F1(W)
OC+ID+LD	42.4	54.2	49.0	47.1
IM+ID+LD	15.7	56.2	42.3	31.4
IM+OC+ID+LD	21.6	53.5	41.6	34.1

Table 27: LLaVA performance in terms of F1 scores on MultiOff

A.14.5 Extended clip based template retrieval

As memes can deviate from their template (see Figure 3, we ask ourselves if including examples of templates from the KYMKB can be used to aid LLaVA in its downstream classification task. To do this, we include both KYMKB templates and example as candidates for retrieval, as the examples may be more similar to the meme in a dataset. Note that we can still access the same information, such as the *about* section, as when using the base template entry because of the KYMKB's structure.

Method	All.	Exa.	Iro.	Ant.	Met.	Con.	F1	F1(W)
OC	22.7	25.2	29.9	11.5	18.2	12.4	22.8	21.6
OC+ID	25.2	26.2	32.0	6.1	17.2	9.2	23.6	21.7
OC+ID+LD	26.4	24.1	31.6	14.2	5.4	16.3	23.0	20.6
IM+ID	20.2	30.7	26.2	12.9	21.7	18.0	23.2	22.9
IM+ID+LD	18.1	26.0	27.4	16.8	5.0	14.4	19.5	18.5
IM+OC+ID	20.1	30.4	30.2	11.3	20.5	15.6	24.3	23.0
IM+OC+ID+LD	18.8	24.7	28.2	14.2	9.0	14.0	19.0	19.1

Table 28: LLaVA performance in terms of F1 scores for FigMemes. "OC" refers to the text on the meme, "IM" refers to usage of the image itself, "ID" refers to our predicted meme template title, "LD" refers to label description, i.e a description of what each label means.

Method	sh.	st.	ob.	vi.	F1.	F1(W)
OC	22.3	46.3	38.8	5.0	33.9	33.8
OC+ID	20.3	44.3	43.6	2.6	34.3	34.1
OC+ID+LD	24.8	46.9	36.8	9.7	34.8	34.4
IM+ID	20.8	38.9	43.5	20.9	34.1	35.0
IM+ID+LD	24.0	41.7	43.4	20.6	34.8	36.2
IM+OC+ID	19.7	41.9	46.6	17.4	35.9	36.5
IM+OC+ID+LD	25.6	40.2	45.0	14.8	34.9	35.8

Table 29: LLaVA performance in terms of F1 scores on the MAMI dataset (Sub-Task B).

Method	Of.	N-Of.	F1.	F1(W)
OC	38.7	56.3	49.0	45.6
OC+ID	28.1	55.4	45.0	38.7
OC+ID+LD	34.7	55.4	47.0	42.8
IM+ID	13.5	53.6	39.6	29.1
IM+ID+LD	15.7	56.1	42.3	31.4
IM+OC+ID	10.2	56.0	40.9	28.1
IM+OC+ID+LD	17.3	55.7	42.3	32.3

Table 30: LLaVA performance in terms of F1 scores on the MultiOff dataset.

A.14.6 Clip-based template retrieval filtering

When we also include the examples of a template from the KYMKB, another question naturally arises: what if there are no suitable entries for a given prompt? To handle such a scenario, we create several filters based on the distance between the base template and its instances. We base this on summary statistics like the interquartile range (IQR), three sigma, mean absolute deviation (MAD), and the maximum distance from template to example, which we find to be the most useful. This corresponds to step 4 in 12. That is, for each template we detect, is this meme in fact within the distribution of this template. Note that here we do not use the template examples to detect the template at first, only the templates. We do however use the template examples to make our calculations, which is done using each pairwise distance between the meme template and their examples.

Method	All.	Exa.	Iro.	Ant.	Met.	Con.	F1	F1(W)
IQR								
IM+OC+ID	18.0	30.8	31.5	16.4	18.5	14.4	24.7	22.9
Three Sigma								
IM+OC+ID	17.8	31.2	30.8	15.1	21.3	14.6	24.8	23.2
MAD								
IM+OC+ID	20.8	31.3	30.8	14.7	20.2	16.1	25.2	23.7
Max								
IM+OC+ID	20.5	32.0	31.4	15.5	20.2	14.2	25.4	23.8

Table 31: LLaVA performance in terms of F1 scores on FigMemes. "OC" refers to the text in the meme, "IM" refers to using the meme itself, "ID" refers to our retrieved meme template title, "LD" refers to label description, i.e a description of what each label means.

Method	sh.	st.	ob.	vi.	F1.	F1(W)
IQR						
IM+OC+ID	22.5	41.9	46.7	14.8	36.4	36.6
Three Sigma						
IM+OC+ID	22.2	43.1	45.2	14.5	36.0	36.4
MAD						
IM+OC+ID	21.4	42.6	45.4	15.1	35.9	36.3
Max						
IM+OC+ID	19.7	42.8	49.0	8.1	36.4	36.3

Table 32: LLaVA performance in terms of F1 scores on the MAMI dataset (Sub-Task B).

Method	Of.	N-Of.	F1.	F1(W)
IQR				
OC+ID+LD	43.9	60.6	53.7	50.4
IM+OC+ID	12.3	57.0	42.3	29.7
Three Sigma				
OC+ID+LD	46.3	56.1	51.7	50.1
IM+OC+ID	8.3	56.4	40.9	27.1
MAD				
OC+ID+LD	42.4	54.2	49.0	47.1
IM+OC+ID	12.0	55.6	40.9	29.0
Max				
OC+ID+LD	35.4	50.0	43.6	41.1
IM+OC+ID	10.1	55.3	40.2	32.7

Table 33: LLaVA performance in terms of F1 scores on the MultiOff dataset.

A.14.7 Extended meme information

We choose the maximum distance algorithm as the default method of selecting relevant meme content based on its simplicity and performance in filtering experiments above. We now investigate including the *about* section as additional information to add to our prompt and if this grounds the LLM in meme knowledge.

Method	All.	Exa.	Iro.	Ant.	Met.	Con.	F1	F1(W)
IQR								
OC+ID+KYM	24.7	28.2	33.3	14.3	20.0	16.5	25.4	24.4
Max								
IM+OC+ID+KYM	20.3	30.4	30.1	12.2	20.6	19.1	24.9	23.6

Table 34: LLaVA performance in terms of F1 scores on FigMemes. "OC" refers to the text in the meme, "IM" refers to usage of the meme itself, "ID" refers to our retrieved template title, "LD" is the label description, i.e a description of what each label means.

Method	sh.	st.	ob.	vi.	F1.	F1(W)
IQR						
OC+ID+KYM	24.9	45.3	47.2	8.3	37.8	37.3
IM+OC+ID+KYM	21.5	41.0	43.7	14.3	34.5	35.0

Table 35: LLaVA performance in terms of F1 score on the MAMI dataset (Sub-Task B).

Method	Of.	N-Of.	F1.	F1(W)
IQR				
OC+ID+KYM	55.2	57.5	56.4	56.1

Table 36: LLaVA performance in terms of F1 scores on the MultiOff dataset (Sub-Task B).