A Template Is All You Meme

Anonymous ACL submission

Abstract

A templatic meme possesses a base semantics that can be tailored by whomever posts it on social media. Machine learning systems that treat memes as just images with text struggle to be performant, which is likely due to such systems having insufficient context. There can be more to memes than the obvious image and text. To aid understanding of memes, we release a knowledge base of memes, composed of more than 5,200 meme templates, detailed information about each one, and 54,000 examples of template instances (templatic memes). To demonstrate the semantic signal of meme templates, we formulate a majority-based, non-parametric classifier that leverages our knowledge base. Our method outperforms more expensive techniques but exposes an underlying issue with meme datasets, where template information is leaked from the training data and models can exploit this knowledge in a way we may not want them to. To control the impact of this template awareness, we reorganize datasets to account for the influence of meme templates. Our re-split datasets discourage undesirable shortcuts to meme understanding, resulting in increased model robustness. This work sets the state-of-the-art for five of the six tasks that we consider.[1]

1 Introduction

Memes are a modern form of communication capable of conveying complicated messages in a succinct manner. The AI research community and datasets treat memes as static images that sometimes have text (Du et al., 2020; Qu et al., 2022). This is only part of the story as memes have many definitions, such as a unit of cultural transmission, or a unit of imitation and replication (Dawkins, 1976). However, all memes possess the trait of referencing a cultural moment shared by a group of people. Despite their inherent basis in Internet culture, they exhibit sociolinguistic traits typical 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.

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.

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 an instance of the popular *Is This a Pigeon?*[2] template, along with two novel memes that we generated ourselves: the first and the second images on the left. 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.

We distinguish between templatic memes and

[1] Our code and data are available at [REDACTED].

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

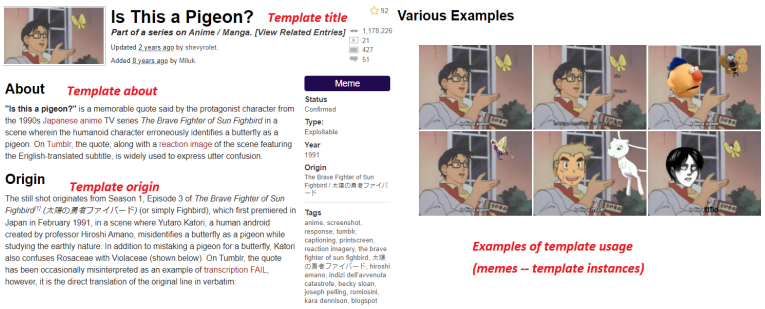

Figure 2: Example entry from KYM where we have labeled relevant data fields in red.

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 or even memes (see Appendix A.2).

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 the base semantics of a specific template, and in order to understand a new template, one 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 undesirable content (Pramanick et al., 2021a), such as misinformation and hate speech. Memes usually express concepts humorously, and humor has been shown to increase the persuasiveness of an idea (Walter et al., 2018). Thus, it is important that we develop systems that can understand memes to prevent the spread of harmful content.

Here, we create and release the Know Your Meme Knowledge Base (KYMKB), a general-purpose database rich with images and information about meme templates, which we scraped from KYM. We hypothesize that knowledge about templatic memes and the KYMKB provide context that was not used in previous work and can aid in meme understanding. To demonstrate the value of the KYMKB and the saliency of the signal created by templatic memes, we develop a meme classification method, Template-Label Counter (TLC). TLC is a majority-based classifier that assigns templates to memes based on the distance between their vector representations. We can then assign the most frequent label for a given template to a novel meme if it is an instance of that template. We find that TLC outperforms fine-tuning pretrained models (PLM), while also being more computationally efficient.

The success of TLC reveals an issue in how meme datasets are created, which we call *template awareness*, where template information is leaked to the model from the training data, which may not be desirable. To investigate this, we developed the Template-Aware Splitter (TSplit), which (re)organizes dataset entries based on their distance in feature space from the KYMKB. TSplit discourages models from bypassing meme understanding and improves model robustness. Our examination of meme templates and of our knowledge base and methods results in state-of-the-art (SOTA) performance on five of the six tasks that we consider. Our contributions are as follows:

1. We release the KYMKB, a knowledge base with 54,000 meme-related images and information about them.

2. We propose TLC, an efficient and effective majority-based classifier.

3. We propose TSplit for (re)splitting meme datasets and increasing model robustness.

2 Related Work

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

[3] https://knowyourmeme.com/

Memes as harmful content This includes MultiOFF (Suryawanshi et al., 2020), a dataset of offensive memes related to the 2016 US presidential election. The MAMI dataset (Fersini et al., 2022) is from SemEval-2022 Task 5: 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. Recently, Hwang and Shwartz (2023) released a dataset of meme explanations to aid in resolving metaphors in memes.

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 Most closely related to our work is Tommasini et al. (2023), who 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 did not include images, made no attempt to leverage the graph in a downstream task, nor is it clear how their graph could be applied to actual meme analysis due to a lack of explanations and demonstrations.

The current work Our work differs from the above in a number of aspects. We are the first to specifically exploit meme templates and to distinguish between templatic memes and non-templatic ones. Second, our KYMKB is much larger: it is composed of more than 54,000 images, while MAMI, the largest dataset above, is composed of 11,000 memes. The KYMKB is not labeled for a specific task, but contains information about templatic memes, such as the title, meaning, and origin. While our classifier does perform inference with a multimodal model, CLIP (Radford et al., 2021), for encoding, it does not rely on expensive fine-tuning or brittle prompting. We use a distance-based lookup to find the most likely template and choose the most frequent label associated with a template for a novel meme. This method reveals the issue of template awareness, which we believe affects all meme datasets. We therefore (re)organize datasets based on *templateness*, or the Euclidean distance of meme-vector representations from the KYMKB. We find this can improve meme analysis.

3 The Know Your Meme Knowledge Base

Know Your Meme, or the "Internet Meme Database", can be thought of as the Wikipedia for memes. Users create web pages with a meme template and document information about the meme, e.g., its origin and meaning, and add examples of its usage (see Figure 2). The community reviews and eventually 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 for meme-related knowledge, but it has been under-utilized 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

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

[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).

from KYM that are approved by the community, scrapping 5,220 base templates and 49,531 examples (see Appendix A.3).

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 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

Our knowledge base enables insightful exploratory data analysis with well-known algorithms that can be used "off the shelf", giving us access to information about a novel meme by considering the text connected to the base template, such as the *about* section. To demonstrate this, we fit a nearest neighbor lookup on encoded templates in the KYMKB, as this is an intuitive and commonly used vector-similarity measure (Buitinck et al., 2013). We then query it on six existing meme classification tasks (see Tables 3 and 4). In the main text, we investigate FigMemes, as we consider it a difficult dataset, but additional analysis can be found in Appendix A.4. 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.

Figure 3 shows a sample of our results. We note that in 39.2% of cases, the meme in the FigMemes dataset is a base template or a distorted or cropped version of it, such as the first two columns in the figure. We also observe that an additional 15.2% are instances of templates tuned by the 4chan poster, such as the third column. This suggests that for this dataset, we can then easily access detailed information about its memes via the KYMKB.

In 16.8% of cases, the KYMKB matches a meme or an image that is not a template instance. The 7^{th} column shows the template of *You Get Used To It*[8] matched to a picture that appears to be a still from another anime.[9] We believe 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 do we argue that the KYMKB encompasses all meme knowledge.

We are able to match templates to relevant instances despite different appearances, which make up the remaining 28.8% of the examples we analyzed. 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*[10] 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 with multiple neighbors retrieves enough information in the form of the *about* sections to interpret this meme as the alt-right angrily expressing derision,

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

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

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

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

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

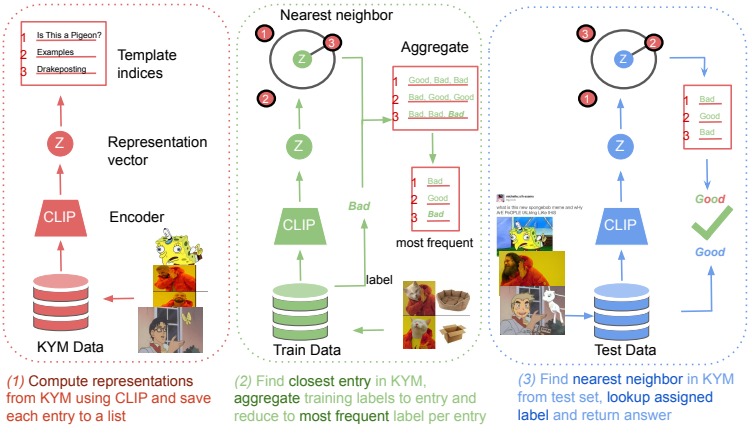

Figure 4: TLC encodes meme knowledge from the KYMKB and computes a nearest neighbor index. We then encode the training data and query our lookup, recording and keeping each template's most frequent class. Finally, we encode the test data, query the index, and assign the closest template's label to the test data.

consistent with /pol/ (Hine et al., 2017), the domain from which FigMemes was created.

5 Template-Label Counter

We hypothesize that many meme datasets are often nothing more than examples of popular templates we have collected in 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 4).

Injecting meme knowledge Considering the success we had in Section 4, we again opted for nearest neighbor indexing as a measure of similarity. We formalize this as a ranking task and first create a reference to our templates, $ref = f(X_{KYMKB})$.

Injecting dataset knowledge The next step is to learn the idiosyncrasies of a dataset, such as the 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 index to the most frequent label, as below.

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

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. Different encoders can also be used. 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 the two via the Hadamard product, and normalizing both vectors and using the average of the two modalities as the final input vector (Yu et al., 2023). There is also 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.6). Note that TLC is reliant on the KYMKB.

5.1 Classification Experiments

Baselines and experimental setup We test various versions of TLC on six meme classification tasks: FigMemes, MultiOff, Memotion 3 Tasks A

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	NA	NA
Best previous: vision only	24.0	NA	NA	*47.69*	NA	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 1: 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 Appendix A.8 for TLC hyperparameter configurations and Appendix A.7 for modelling information from previous work.

and B, and MAMI Tasks A and B (see Tables 3 and 4). Our baseline model is the majority class from the training split for each task.

Results and Discussion Table 1 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 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 with fine-tuning a multimodal PLM for Memotion 3 (B).

We can see that 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, jumping by more than ten points for MultiOff. We find that concatenating the image and the text modalities tends to be the strongest TLC configuration. We interpret this as support for 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 and we ignore the examples. In most other cases, TLC$_{Templates}$ is within one point if not higher than TLC$_{Templates+Instances}$. We can create a strong model grounded in meme knowledge by encoding one-tenth of the available images, supporting our claim 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 the expensive training of a large model. We further note that meme templates cross cultural and linguistic boundaries, as indicated by our strong performance on both Memotion 3 tasks, a multilingual dataset of memes in Hindi and English. The power of templates gives 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 created via crawlers and 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 5).

TLC's strength and simplicity reveals a problem in the creation of meme datasets. By taking the most common label for a given template, we assume that a template can only convey a fixed message; for classification, this means that, e.g., a template can only ever be harmful or not harmful. This aligns with our argument that the template grounds the meme in a base semantics, but contradicts the reality that a meme's meaning can be tuned by the poster. By over-fitting to the majority class, TLC is naïve but competitive as compared to more expensive methods. This demonstrates the power of meme templates, but by design TLC cannot interpret novel templates as it just exploits the manner in which meme datasets were created.

6 Template-Aware Splitter

In a sense, TLC utilizes leaked knowledge as it has not learned to interpret memes, but instead exploits the template signal that has been neglected in the

literature. We call this *template awareness*, where the model uses template information that appears in both the training and the test data in a way we may not want it to, e.g., memorizing a template's most frequent label as a shortcut to meme understanding.

To investigate this, we demonstrate another use case of and develop a tool from the KYMKB, which we call the Template-Aware Splitter (TSplit). TSplit quantifies the notion of *templateness* by comparing each base template to its examples (see Figure 2) contained in our knowledge base. Formally, we embed a template ($ref = f(KYMKB_{[i]})$, where i is a template index) and its examples ($examp = f(KYMKB_{[i][j]})$, where j is an example index) and compute and record the Euclidean distance between a template and each of its examples ($dists_{[i]} = dist(ref, examp_{[j]})$). These distances are then used to compute a threshold value, for example, the median distance from template to instance ($threshold_{[i]} = median(dists_{[i]})$). We consider four ways to compute the threshold value: maximum, median, mean, and 25^{th} percentile values of distance from a template to its examples. Using the maximum value is a lenient view of what constitutes a template instance, as they can be very different in appearance (see Figure 3), while the 25^{th} percentile would be a stricter view. TSplit encodes a meme and finds its closest template in the KYMKB. If the distance exceeds the template's threshold, we assign it a unique identifier; otherwise, we record it as an instance of that template.

We use TSplit to reorganize the datasets. We first assign each meme in a dataset to a template or we declare it non-templatic. To reorganize the datasets, TSplit samples our templates and unique identifiers without replacement, such that a template instance 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). We record the ratio of each split of the original dataset (for example, training 60%, validation 20%, test 20%) and we maintain this ratio in our reorganized dataset (see Appendix A.10).

6.1 Classification Experiments

Baselines and experimental setup We perform fine-tuning experiments to examine the effect of template awareness. We opted for fine-tuning as prompting with (multimodal) LLMs does not result in meme understanding (Hwang and Shwartz, 2023), and this is confirmed and supported by our own experiments (see Appendix A.9). Our baseline model is CLIP fine-tuned for classification on the original datasets, called Original$_{ViT-X}$, using both the OCR text and the meme. To use CLIP for classification, we add a single feed-forward layer on top of the text and image encoders. We compare this baseline to the same model but fine-tuned on our resplit versions of the same datasets. We train for 20 epochs with AdamW (Loshchilov and Hutter, 2019) and a learning rate of $1e^{-5}$. We perform test evaluation using the checkpoint that performed best on the validation dataset from any given epoch. If a validation split does not exist, we create one by sampling 20% of the training data. We experiment over five seeds, reporting the mean and the standard deviation with three encoders of varying sizes.[11]

Results and Discussion Table 2 shows our results. In the case of MultiOff, a relatively small binary dataset, 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. Interestingly, TLC set the new SOTA for this dataset (an F1 of 61.89), but was beaten by both our baseline (64.65) and TSplit (63.58).

For Memotion 3 (A), a difficult task that contains many templatic memes, all versions of TSplit outperform our baseline, even though 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 capture a distribution decoupled from templates, we consistently attain scores of approximately 33–35, and as with TLC, we once more get multilingualism without even trying. Task B, however, tells a different story, where the TSplit datasets appear more difficult than the original for all encoders. TLC tells us that we can exploit template-awareness for this task. Therefore, the performance naturally drops when the model cannot depend on this information.

Controlling for template awareness makes difficult tasks easier by forcing the model to learn general properties and not take shortcuts to meme understanding. 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, achiev-

[11] We used a 40 GB NVIDIA A100 Tensor Core GPU.

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

Table 2: Fine-tuning results comparing TSplit against the original dataset. The only difference between TSplit versions is how the *templateness* threshold is chosen. 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 evaluation measures remain unchanged (see Table 3).

ing SOTA performance, an F1 of 47.91. For MAMI A and B, TSplit outperforms our baseline in all cases, and once again we achieve SOTA on Task A (an F1 of 83.63). Previously reported results relied on fine-tuning and ensembling multimodal models, including CLIP (see Appendix 3), 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 unique identifiers account for non-templatic memes or images, common in meme datasets and in MAMI. By removing meme/image conceptual overlap between the train and the test split, the model cannot rely on leaked information or spurious artifacts, resulting in a task that is easier and no longer requires ensembling.

A tolerant view of what constitutes a template instance results in strong performance. The maximum distance between a template and its examples implies a high threshold. TSplit_{max} assigns more templates to memes and fewer unique identifiers, resulting in less template leakage. Templatic memes can appear quite different from their base (see Figure 3) and even non-templatic memes may reference a template indirectly (see Figure 5). We therefore find this result intuitive.

7 Conclusion and Future Work

We created the KYMKB, containing more than 54,000 images and 5,200 base templates with detailed information about each one. To demonstrate the power of templatic memes, we conducted exploratory data analysis, showing that a comparison of templates to memes in existing datasets creates a strong signal that we can leverage through the KYMKB to inject models with meme-specific information. To demonstrate this, we proposed TLC, a majority-based classifier and found it competitive with far more expensive methods. TLC revealed the issue of template awareness, where models exploit the template signal at test time, which may not be desirable. We therefore proposed TSplit and found that it discourages using shortcuts to meme understanding and can result in easier tasks and more robust models. It was not our goal to create SOTA meme classifiers, but we believe our methods are convincing demonstrations of the value of the KYMKB and the strength of meme templates.

While each dataset has its own idiosyncrasies, our resources provide unified, inexpensive tools for future research grounded in meme knowledge. Template awareness may not be a problem for all meme analysis tasks, but we believe it is an issue that researchers should be conscious of. Any sampling method for a given task will come with its own biases, yet we feel ours offer advantages over random sampling. Memes may be hard to analyze, but they are not random.

In future work, we will apply the KYMKB to more datasets and languages in a cross-language setup. We further plan to explore automatically augmenting the KYMKB with new memes and with new templates.

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. TLC assumes, however, that each meme is a template instance, which is not always the case. However, we believe that 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 Appendix A.1). We have performed an examination of multimodal LLM prompting performance and found such a paradigm to be insufficient for meme understanding (see Appendix A.9) and our findings are consistent with Hwang and Shwartz (2023). More recent vision language models, such as Otter (Li et al., 2023) or IDEFICS,[12] 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.

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. We developed TLC, which is computationally efficient and can flag dangerous memes in an interpretable manner. 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.

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

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? WWW '23, page 3734–3743, New York, NY, USA. Association for Computing Machinery.

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? *arXiv preprint arXiv:2402.04967*.

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*, KDD '16. 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. *arXiv preprint arXiv:1909.11740*.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

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. *arXiv preprint arXiv:2010.11929*.

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.

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.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

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. *arXiv preprint arXiv:2305.18391*.

Patrick 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*, volume 33, pages 9459–9474. Curran Associates, Inc.

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 arXiv: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, Yuheng Li, and Yong Jae Lee. 2023. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

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 arXiv:2303.09892*.

Abel L Peirson and E Meltem Tolunay. 2018. Dank learning: Generating memes using deep neural networks. *arXiv preprint arXiv: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 arXiv: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*, 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. *arXiv preprint arXiv:1409.1556*.

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 arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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 arXiv: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.

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*,[13] a template that references a popular YouTube video[14] where a player in World of Warcraft[15] 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.[16] 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 Appendix A.5, 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. TLC, for example, relies on the concept that memes are images in order to perform classification, but as we point out in Section 5.1, 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.

A.2 Non-Templatic Memes

In this Appendix section, we provide examples of images/memes which we consider to be non-templatic (see Figure 5). The first and third examples are a visual joke and pun respectively. The text in the first does make reference to the *Doggo*[17] and language from the *Cheezburger*[18] templates. The second is a still from a .gif that references the film *the Sword in the Stone*[19] and can actually be found on KYM.[20] 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

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

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

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

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

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

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

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

[20] https://knowyourmeme.com/photos/546232-reaction-images

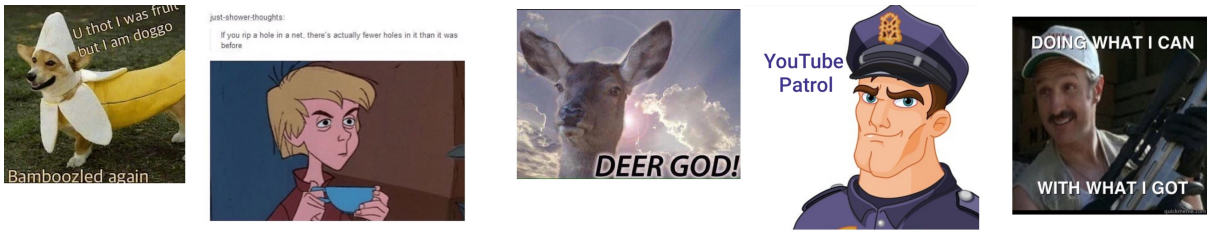

Figure 5: Examples of non-template images found in FigMemes.

example is a still from the movie *Tremors II: Aftershock*[21] where the image is the correct character but the text is quoted anachronistically from another part of the film.[22]

A.3 Scraping details

We use the Wayback Machine[23] (WM) to adhere to KYM's terms of use.[24] WM's snapshots of the Internet are incomplete, making it impossible to completely capture KYM; 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 and we are committed to regularly updating the knowledge base ourselves as new entries become available. All information relevant to the scraping process is preserved in a .json file, linking templates to their examples (see Figure 6).

A.4 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, but we make all the code and the resulting image files freely available.

Figure 7 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. This is consistent with TLC's optimal settings where we found that keeping the modalities separate or concatenating them to be the strongest version of our method.

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 so obvious, we match text or characters, such as *Why So Serious* or the *Joker*,[25] or concepts that exist in only meme or Internet culture. For example, consider the first column under the concatenation setting in Figure 7. We observe the character of

[21] https://en.wikipedia.org/wiki/Tremors_2:_Aftershocks)

[22] https://www.youtube.com/watch?v=eMODPOmB-cA

[23] https://web.archive.org/

[24] https://knowyourmeme.com/terms-of-service

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

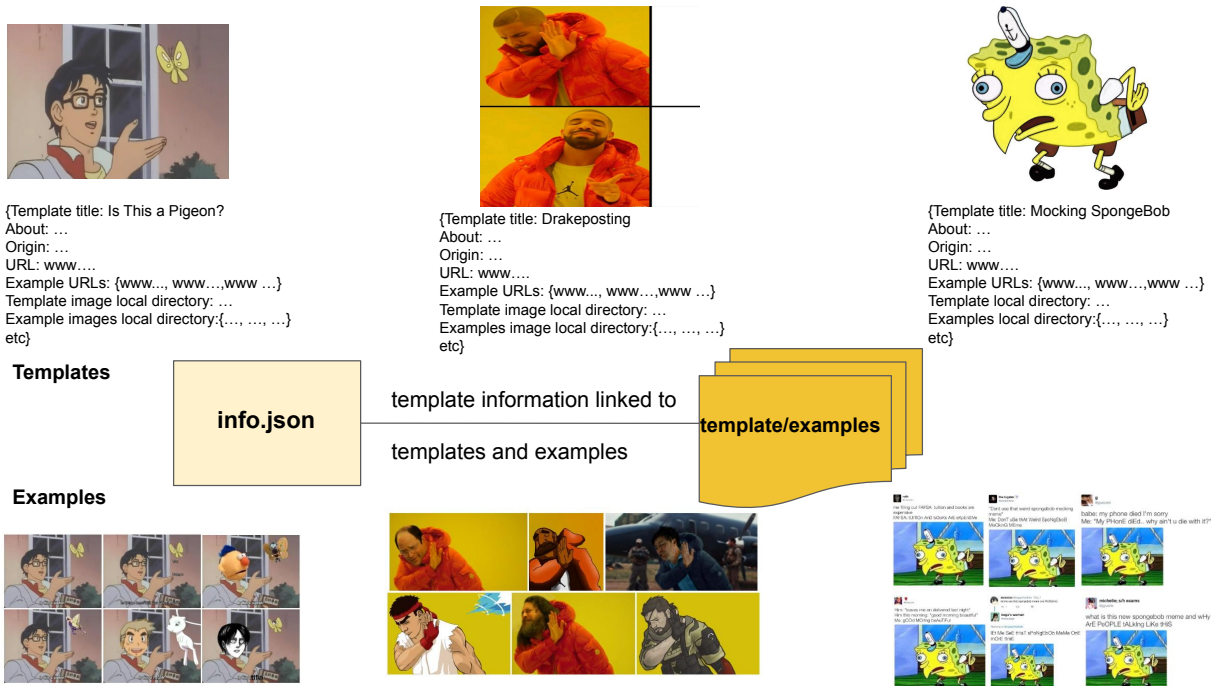

Figure 6: The KYMKB records all textual information about a meme in a .json file, including the text found on KYM, the URLs used in the scraping process, and local locations of all template and example images in the knowledge base.

Wojak in the *I Support the Current Thing* meme template,[26] [27] 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*.[28] 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.[29] 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 3). Here, for conciseness, we consider only templates and FigMemes, as we consider it a difficult dataset and it has the most labels; however, we make all resulting image files available with the KYMKB along with the code to reproduce them.

Figures 8 and 9 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

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

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

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

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

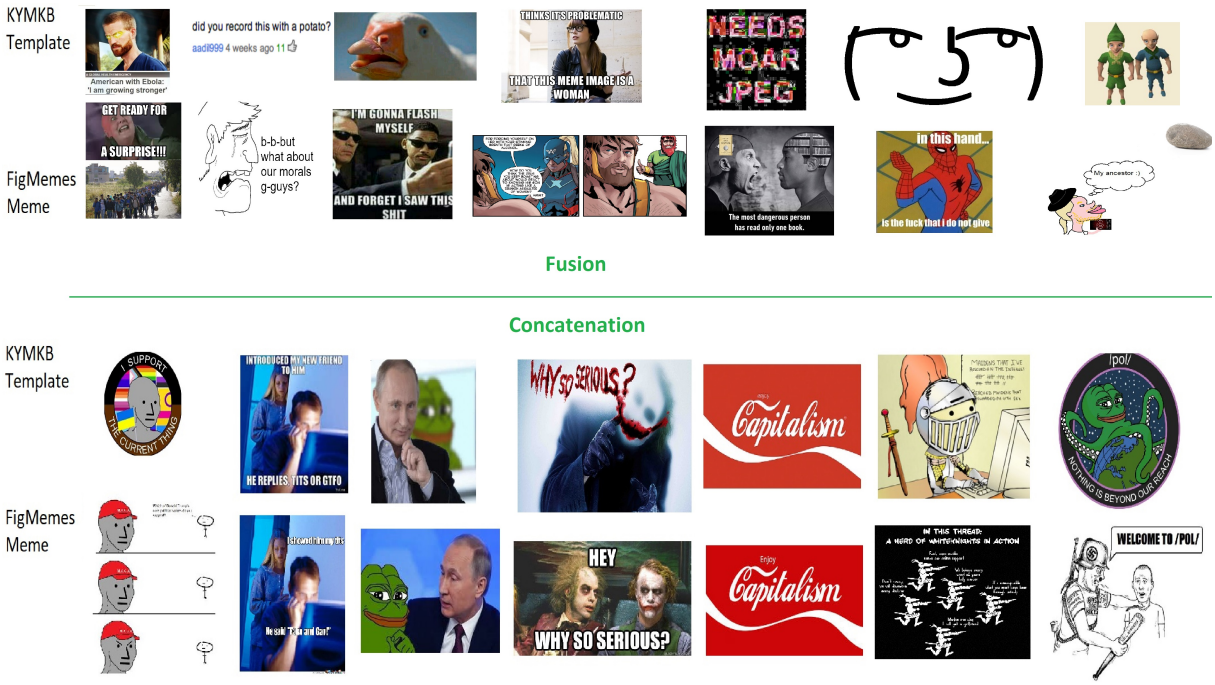

Figure 7: KYMKB templates matched via similarity search to FigMemes images.

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 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.[30] 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.

A.5 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*,[31] which can involve posting an image of Rick Astley from the Never Going to Give You Up music video,[32] 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.[33] We would argue this is an intertextual meme instance referencing the Rickroll template.

Loss[34] is another famous template where an instance is an action, not an image. The template is a reference to the Ctrl+Alt+Del Comic[35] gaming webcomic, which made an uncharacteristically 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,

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

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

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

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

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

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

Figure 8: 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.

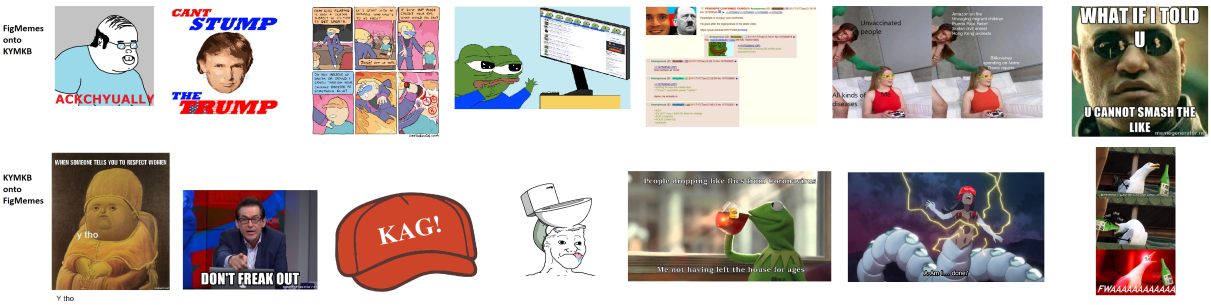

Figure 9: In the first row, we show the templates closest to seven KMeans centroids fit on the FigMemes, while in the second row, we show FigMeme images closest to seven centroids derived from the KYMKB. We only use the image modality, which results in seven distinct images.

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 10).

Instances of the *Planking*[36] 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*.[37] 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.[38] 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.[39] 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.6 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. 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 exam-

[36] https://knowyourmeme.com/memes/planking

[37] https://knowyourmeme.com/memes/thinking-face-emoji

[38] https://knowyourmeme.com/memes/oof-roblox-death-sound

[39] https://www.roblox.com/

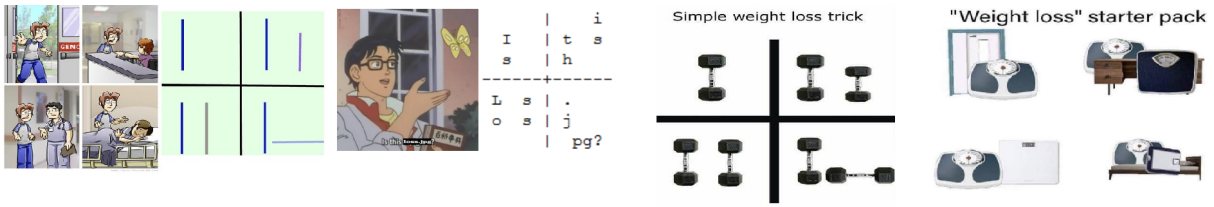

Figure 10: The first image is the original template of *Loss*, while the other three images are *Loss* instances, all of which are visual puns that cannot be understood without knowing the original template. The second image is another intertextual meme where *Loss* and *Is This a Pigeon?* have been amalgamated.

ples, 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. We do not report results related to late fusion. However, we make all our results available with our source code.

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.7 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 as "Best previous" in Table 1. 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) pretrained 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

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
MEMEX	Relevant Explanation	2	3403	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 3: Summary of the previous works 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
MEMEX	BERT	ViT	Meme Transformer / Meme LSTM (novel model)	0.55 to 0.72
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 4: Continued summary of the previous works 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.

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%), Anthropomorphism (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.

Finally, MEMEX is a binary task, baseless (30%) vs. valid (70%), of whether or not a explanation document is relevant for a given meme. In their first stage of annotation they report a Cohen's Kappa of 0.55, moderate agreement, but report a Cohen's Kappa of 0.72, substantial agreement in the second stage. MEMEX formulates a new task, corpus, and model for meme understanding. Their model relies on BERT embeddings fed into a novel "meme-aware" Transformer model (Vaswani et al., 2017), whose outputs are then decoded by a "meme-aware" LSTM layer. At time of writing, their validation split is not public. Note that we do not present results on this dataset in the main text.

See Tables 3 and 4 above for a summary of this information.

A.8 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 5.1. 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,[40] organized in each table in that order and also by the number of neighbors used for voting. The best configuration was chosen for Table 1 in the main text. Note that $\text{TLC}_{About/OCR}$ is

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

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, 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 neighbors 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 5: 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 6: 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 7: 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 8: 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.

A.9 Prompting with LLMs

As the TLC method is a non-parametric approach, we ask ourselves 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*,[41] *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 KYM?
- Does the inclusion of the retrieved template *about* section in the prompt improve LLM performance?

We perform an extensive ablations to answer the above questions and the main results are shown in Table 9. Our overall setup is illustrated in Figure 11. 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.

Accounting for negation To test if the model explicitly states which label is not part of the answer, we used a 3-token window around the label in the output. When recomputing the scores for the best FigMemes LLaVA model grounded in KYMKB, the scores changed from 25.4 to 25.3 in Table 9, meaning a negligible difference when checking for negation.

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,

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

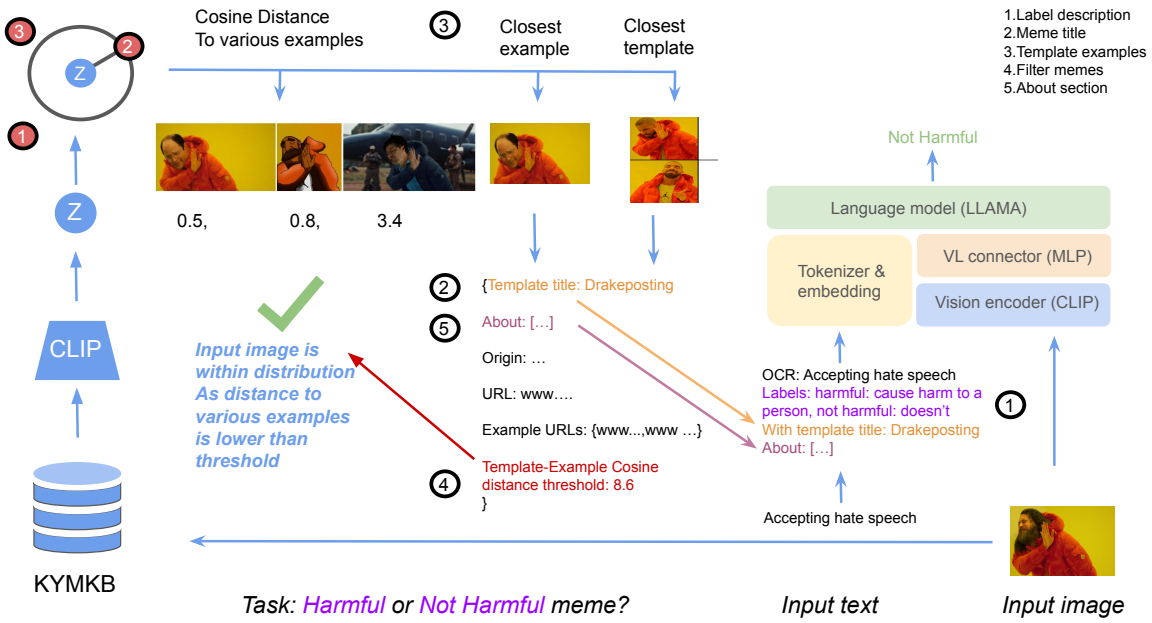

Figure 11: The complete setup of using LLAVA with KYMKB. All experiments are with multiple examples as illustrated in Figure 10. Initially, both the OCR text and input meme are passed to LLaVA alongside k examples with ground truth labels and a selection of labels to choose from. Then we add 5 steps with further modifications as shown in the top right corner of this figure and according to the additional steps in Table 9. 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 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 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.

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 9 shows our results. We compare our findings to TLC, and the overall trend is that LLaVA is not as strong as TLC. In the case of MultiOff and Memotion 3 (B), we see that grounding our LLaVA model in 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 both TLC and TSplit. In general, LLaVA performs worse than TLC, thus motivating further study into learning more robust models, as we showed with our TSplit approach. The only real exception to the overall trend is that of 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.9.1 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?

Method	MultiOff	Memotion 3 (A)	Memotion 3 (B)	FigMemes	MAMI (A)	MAMI (B)
TLC	*61.89*	*37.77*	*80.49*	*29.8*	*70.0*	*40.21*
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 9: Classification results for the best performing version of LLaVA grounded in 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.

- 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?

A.9.2 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 10 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.9.3 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.

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

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 11: 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	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 12: 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 13: 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.9.4 Clip-based template retrieval

As an additional ablation, we test if increasing the size of the CLIP model used for retrieval also affects the performance of LLaVA as it does with TLC. 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 14: 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 15: 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 16: LLaVA performance in terms of F1 scores on MultiOff

A.9.5 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 17: 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 18: 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 19: LLaVA performance in terms of F1 scores on the MultiOff dataset.

A.9.6 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 11. 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 20: 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 21: 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 22: LLaVA performance in terms of F1 scores on the MultiOff dataset.

A.9.7 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 23: 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 24: 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 25: LLaVA performance in terms of F1 scores on the MultiOff dataset (Sub-Task B).

A.10 Template-Aware Splitter details

In this Appendix section, we provide additional details on the effects how TSplit reorganizes the datasets we examined the main text. In Table 26, we show examples of TSplit samples detected templates vs. unique identifiers. As mentioned in the main text, using the maximum distance from template to examples as the threshold value for each template results in TSplit detecting more templates, using the 25^{th} percentile results in more unique identifiers.

To avoid overlapping templates and unique identifiers between datasets, we construct an array of distinct objects, meaning detected templates or unique identifiers. We randomly shuffle the array and then set up 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, but because we sample the resplit datasets based on detected templates, it is difficult to maintain the exact numbers.

To empirically verify that TSplit prevents template instances from appearing in both the training and test split of a dataset, we sample 100 memes from FigMemes from both the training and test split of our reorganized datasets. We then manually inspected the memes, looking for overlapping templates. We did this under all four thresholding techniques discussed in the main text where the CLIP encoder was once more ViT-L/14@336px, the overall strongest performer.

Under TSplit$_{max}$ and TSplit$_{median}$, we note templatic characters, such as Pepe the Frog, overlapped in both splits. In such cases, it was Pepe appearing in another template or in a non-templatic meme, but did not note what we would call the same template appearing in both splits.

Under TSplit$_{mean}$, we noted similar characters,

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
$\text{Train}_{percentile}$	167 / 176	890 / 3562	964 / 1198	1506 / 2789
$\text{Test}_{percentile}$	41 / 44	205 / 748	419 / 506	138 / 291

Table 26: Example of how TSplit reorganizes the datasets from the main text, where we show the number of detected templates / number of unique identifiers in each split. The threshold method is in subscript of the split. ViT-L/14@336px was used as the CLIP encoder.

but also one overlapping template, which was Picardía / Thumbs Up Emoji Man,[42] a template that is commonly used on 4chan to mock political ideologies. As this template can take on vastly different appearances, it is intuitive that a strict view of what constitutes a template instance in how the threshold is calculated would result in these instances slipping through our filter.

Under $\text{TSplit}_{percentile}$, we did not note overlapping templates explicitly, but notice the same templatic characters in both splits, such as Pepe, Picardia, SpongeBob, and Spiderman.[43] This is subjective, but while these examples are not necessarily instances of a template, they carry a similar emotional charge as a template, and some users might consider them templatic instances.

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.

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

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