A Template Is All You Meme

Anonymous ACL submission

Abstract

Memes are a modern form of communication and meme templates possess a base semantics that is customizable by whomever posts it on social media. Machine learning systems struggle with memes, which is likely due to such systems having insufficient context to understand memes, as there is more to memes than the obvious image and text. Here, to aid understanding of memes, we release a knowledge base of memes and information found on www.knowyourmeme.com, which we call the Know Your Meme Knowledge Base (KYMKB), composed of more than 54,000 images. The KYMKB includes popular meme templates, examples of each template, and detailed information about the template. We hypothesize that meme templates can be used to inject models with the context missing from previous approaches. To test our hypothesis, we create a non-parametric majority-based classifier, which we call Template-Label Counter (TLC). We find TLC more effective than or competitive with fine-tuned baselines. To demonstrate the power of meme templates and the value of both our knowledge base and method, we conduct thorough classification experiments and exploratory data analysis in the context of seven meme analysis tasks.[1]

WARNING: For demonstration purposes, we discuss and show memes that some may find offensive. These memes in no way reflect our views.

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).

[1] Our code and data are available at https://github.com/[REDACTED]. **Disclaimer:** Our work should only ever be used for academic purposes.

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 suggests ChatGPT can exploit the NLP community for data. The third is a reference to Pokemon, demonstrating that entities can be alluded to with overlaid images rather than text.

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). A meme's meaning can therefore be obfuscated to those not belonging to the in-group, which can make it difficult to understand for many humans, let alone machines.

Meme templates are common patterns or elements, such as text or images, that are used to create novel memes.[2] They can difficult to parse because they can be combined in different ways and each 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 instances of the popular *Is This a Pigeon?*[3] template, along with two novel memes that we generated ourselves: the first and the sec-

[2] https://knowyourmeme.com/memes/meme-templates

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

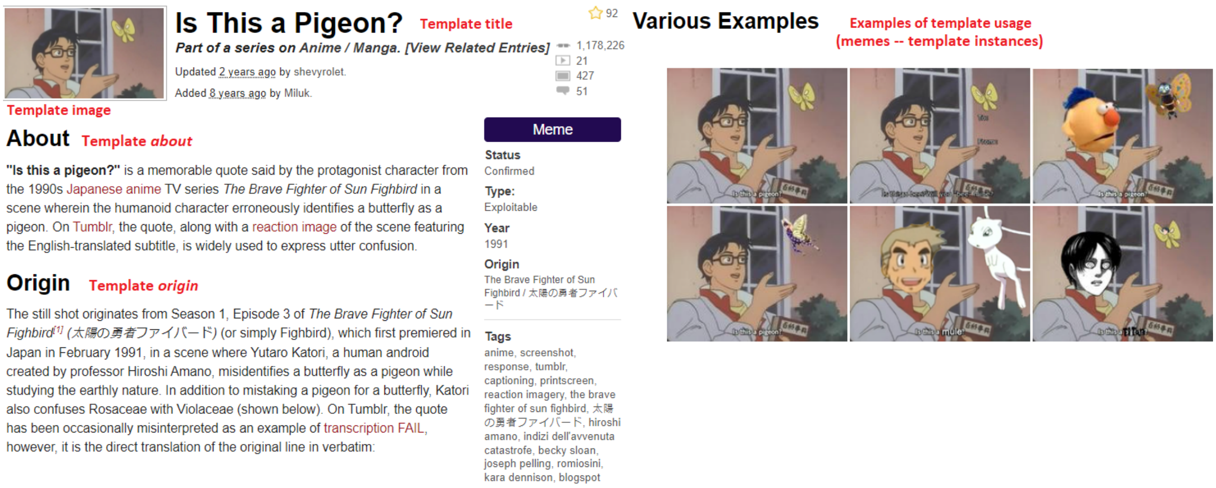

Figure 2: Example entry from KYM where we have added the text in red to label the template title, base template image, and examples of template instances, which are all relevant to our analysis.

ond images on the left.[4] This template conveys the idea that the subject of the man is misinterpreting the object of the butterfly due to his own world-view or limited knowledge. The exact meaning can be tuned via overlaid text or even overlaid images as chosen by the poster. However, these altered images are still considered instances of the same template. To interpret memes, one must not only recognize the entities in the meme, but also the template the meme uses, if any.

We distinguish between templatic memes and other meme types. Templatic memes reference a meme template, which is commonly reused material (text, images, audio, etc), to create a novel instance still grounded in the meme template's semantics. Non-templatic memes can be (visual) puns, jokes, or emphasis that do not reference a meme template and can be understood by people who do not have knowledge of specific meme templates or even memes (see Appendix A.2 for examples).

Know Your Meme (KYM), the Internet Meme Database, is a valuable resource for information related to memes, and specifically, 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 in KYM provide the base template and additional information about it, such as its meaning, origin, various examples, etc. By reviewing the entry of an unfamiliar template and seeing examples of its usage, users can learn how to interpret and to use the template themselves to create novel instances for their specific communication needs.

Memes have been of growing interest to the machine learning (ML) community (Aggarwal et al., 2023) because, not only do they pose a significant learning problem, but they can also be used to spread harmful content (Pramanick et al., 2021a), such as misinformation, propaganda, and hate speech in a convincing manner. Memes are usually used to communicate 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. The best-performing approaches involve fine-tuning large multimodal models on the image and the text, but we are still far from a reliable system capable of accurately flagging harmful memes.

Here, we create and release the Know Your Meme Knowledge Base (KYMKD), a general-purpose database rich with images and information about meme templates scraped from KYM. We hypothesize that knowledge about templatic memes and the KYMKB provides context that is missing from 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 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 outper-

[4] https://imgflip.com/memegenerator/Is-This-A-Pigeon

forms or is competitive with fine-tuning pretrained models, while also being far more computationally efficient. To further show the strength of using templatic memes as contextual grounding for meme understanding, we conduct detailed classification experiments and exploratory data analysis on seven meme classification tasks. We find that by leveraging the semantics of meme templates, we are able to rival fine-tuned methods by making naïve guesses about the label of a meme based on over-fitting to the most frequent class for a given template in the training split of a dataset. We propose that future work should carefully consider templates when constructing datasets, which is made possible thanks to the KYMKB. 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, majority-based classifier that is competitive with or stronger than more expensive methods.

3. We perform an extensive study of TLC and the KYMKB in the context of seven meme analysis tasks to demonstrate its potential.

2 Related Work

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

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.

FigMemes (Liu et al., 2022) scraped images from a politically incorrect and infamously toxic board on 4chan, /pol/,[5] 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.

The above works all fine-tuned pretrained multimodal models on their respective datasets, but do not use additional context in order to increase meme understanding. In general, this is a trend in meme-related ML research. One exception to this trend is MEMEX (Sharma et al., 2023). It formulates a new dataset and task by creating meme-explanation (a document) pairs and asks the question of whether a given explanation is accurate for a given meme. They use Wikipedia[6] and Quora[7] to assemble explanation documents and create a novel 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 about named entities, not about memes.

Most related to our work is Tommasini et al. (2023), software for making 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, it does not include images, makes no attempt to leverage the graph in a downstream task, nor is it clear how it could be applied due to a lack of documentation, explanations, and demonstrations.

Our work differs from the above in a number of aspects. We are the first work to specifically exploit meme templates and distinguish between templatic memes and non-templatic memes in AI literature. Second, our KYMKB is much larger: it is composed of more than 54,000 images, while MAMI, the largest dataset mentioned above, is composed of 11,000. Moreover, it is not labeled for a specific task, but contains detailed, general information about templatic memes, such as the title, meaning, and origin. Furthermore, while our method does perform inference with a multimodal model, CLIP (Radford et al., 2021), for encoding, we perform no expensive fine-tuning or prompting. We also do not rely on graphs to connect or to ground memes in potentially erroneous contexts. Instead, our TLC approach uses a distance-based lookup to find the most likely template and chooses the most frequent label associated with a template for a novel meme, making our approach computationally efficient. We study the value of our knowledge base and method by comparing them to other works, especially recent works or papers related to memes

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

[6] https://www.wikipedia.org/

[7] https://www.quora.com/

spreading harmful content. Finally, the KYMKB provides meme-specific context, rather than querying potentially unrelated knowledge sources.

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 entries 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 for an example entry). After an entry has been created, it is reviewed and eventually approved by the community. This entry can then be updated as the meme'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 semantics. 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.[8]

KYM is a valuable resource for meme-related knowledge, which has been under-utilized by the AI community. To address this, we create the KYMKB, a collection of meme templates, examples, and detailed information about the meme's usage. In order to ensure the quality of the meme entries, we crawl entries from KYM that have been confirmed by the community, scrapping 5,220 base templates and 49,531 examples, for a total of 54,751 images. See Appendix A.3 for details.

The KYMKB is organized for ease of use, to connect relevant meme templates to related information and examples, and to maintain all URLs used in our crawling process for reproducibility and future work. Figure 6 shows the structure of our knowledge base, where all textual data, such as the *about* section, is linked to a template and recorded in a JSON file along with the local location. The template is then in a parent directory, the subdirectory of which contains the examples. On average, each template has 9.49 examples associated with it.

4 Template-Meme Analysis

We hypothesize that retrieval-based methods should allow us to match base template to memes in the wild, giving us to access information about a novel meme by considering the text connected to the base template, such as the *about* section, in KYMKB. To confirm this, we fit a nearest neighbor lookup on encoded template images in our knowledge base, as this is an intuitive and commonly used vector-similarity measure (Buitinck et al., 2013). We then query it on seven existing meme classification tasks (see Table 2). Specifically, we query the 500 closest neighbors and manually inspect the similarities. In the main text, we investigate FigMemes, as we consider it a difficult dataset, but additional analysis can be found in Appendix A.4. We use CLIP as our encoder as it is a commonly-used pretrained model for vision and language learning problems and memes (Pramanick et al., 2021b).

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, 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 image that, to the best of our knowledge, is not a template instance. In the 7th column, we see the anime template of *You Get Used To It*[9] matched to a picture that appears to be a still from the anime.[10] We believe the FigMemes image is not an instance of the aforementioned template, however, 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,[11] a template with many different versions originally used in a manner similar to emoticons, but which has since become 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

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

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

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

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

merging into a single meme. Mocking SpongeBob[12] is a popular template poster 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 similarity measures and multiple neighbors retrieves enough information in the form of the *about* sections to interpret this meme as the alt-right angrily expressing derision, consistent with /pol/ (Hine et al., 2017), the domain from which FigMemes was created.

5 Template-Label Counter

We hypothesize that since KYM is composed of popular meme templates, meme datasets are nothing more than customized instances of the templates we have collected in the KYMKB. We should therefore be able to compare a novel meme to KYMKB templates, select the most similar template, and obtain a meme-specific context.

To test our hypothesis, we consider a meme's label, for example harmful vs. neutral, in a dataset to be a reflection of that meme's semantics. By matching KYMKB templates to memes in the training split of a dataset, we can then assign that label to any other instance of that template, i.e., a novel meme in the test split of that dataset (see Figure 4).

Injecting meme knowledge The first step in the TLC pipeline is to encode all the meme templates and optionally the examples. Considering the success we had in Section 4, we again opt for encoding via CLIP and nearest neighbor indexing as a measure of similarity. We can formalize this as a ranking task where we first set up a reference to our templates, $ref = CLIP(X_{KYMKB})$.

Injecting dataset knowledge The next step is to learn the idiosyncrasies of a dataset, such as the labeling scheme. We encode the training data, $query_{train} = CLIP(X_{train})$, and we query our neighbor index, selecting the closest template and recording the label for each training instance. TLC reduces each index to 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.

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

Testing meme and dataset knowledge The final step is first to encode test data with CLIP, $query_{test} = CLIP(X_{test})$, and then to use our nearest neighbor lookup. We then assign the most frequent label for an index/template obtained from the training data to the testing instance, $\hat{y} = rank(ref, query_{test})$. In cases where we find a template not seen during training, we backoff to the most frequent label in the training data.

Hyperparameters When using TLC, we have the option to ignore the meme itself and instead match the *about* section of templates to the OCR text of a novel meme. Alternatively, we can choose whether we consider the base template or templates and examples for encoding meme knowledge. We can also consider multiple neighbors and select the most common template or label among them. Different encoders, for example, different versions of CLIP, 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 novel meme embeddings. We experimented with concatenating the CLIP embeddings of both modalities, fusing the two via the Hadamard product, or following Yu et al. (2023), we can normalize both vectors and then use the average of the two modalities as the final input vector. We can also use a type of late fusion, where text and image representations vote separately and we aggregate. After the hyperparameters are set, TLC is deterministic (see Appendix A.6 for details).

6 Classification Experiments

We test the various versions of TLC against the results reported for seven meme tasks: FigMemes, MultiOff, MEMEX, Memotion 3 Task A and B, and MAMI Task A and B. See Table 2 for a summary of the datasets and tasks. Regarding the size of each dataset, we provide the same number as reported in the relevant work. However, this is not always the number used in our analysis. In the case of Memotion 3, the labels for the test split, 1,500 memes, are not publicly available, and thus we used the training and the validation split in our analysis. For MEMEX, the validation split, 200 memes, is not public at the time of writing. We therefore consider the training and the test split.

Baselines and setup We compare TLC performance against the scores reported in related work for seven different tasks. Our baseline model is the

Figure 3: The first row contains KYMKB templates, and the second shows their nearest neighbor in FigMemes.

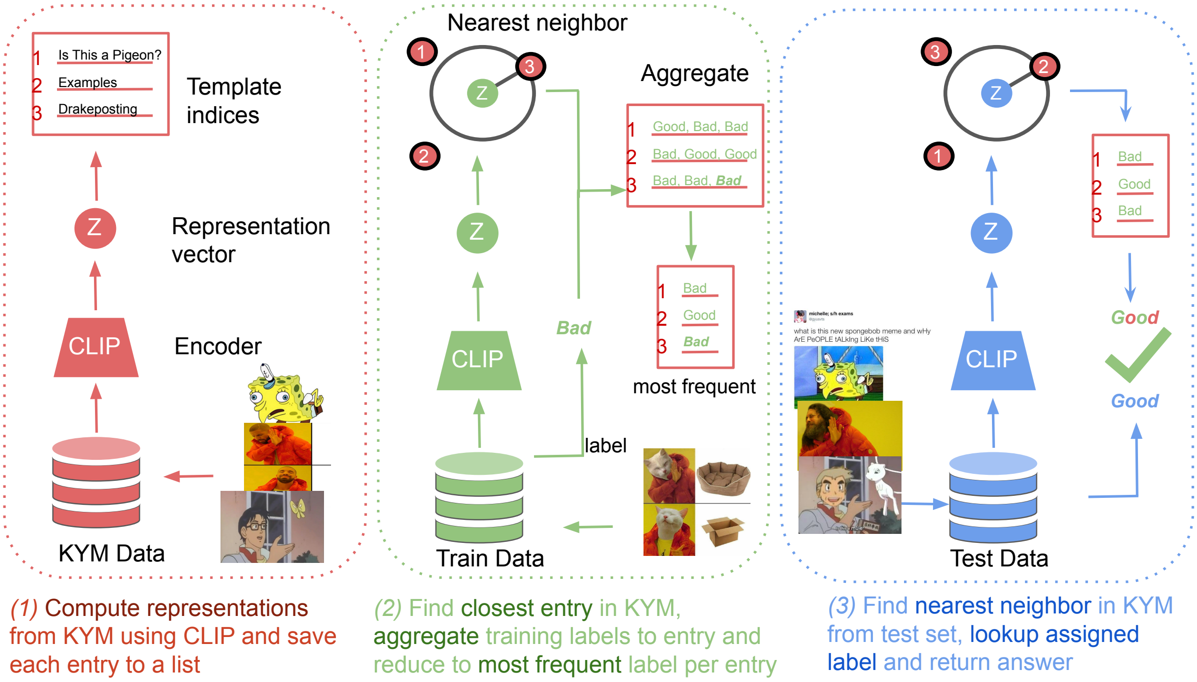

Figure 4: TLC works by first encoding meme knowledge from the KYMKB using CLIP and computing nearest neighbors. We then encode and perform an index lookup on the training data, recording each template's label and keeping only the most frequent class. Finally, we encode the test data, we query the index, and we assign the closest template's label to each test instance.

majority class from the training split for each task. In case the test split is unavailable, we evaluate on the validation split, while in cases where a training, validation, and test splits are available, we add the validation data to the training data. Our evaluation measures are computed with scikit-learn.

6.1 Results and Discussion

TLC outperforms fine-tuned methods Table 1 shows our results. We display the best-performing version of TLC, comparing embedding text versus encoding templates versus templates and examples. We further provide the best result from previous work, where a pretrained model was fine-tuned on OCR text, the meme itself, or a multimodal representation of the two. We note that TLC beats a majority class classifier, but in some cases, counting can work quite well. In the case of Memotion 3 (B), a majority class classifier is competitive with a multimodal, fine-tuned model.

More, more![13] TLC's performance consistently improves as we consider more modalities. Encoding the *about* section of a template and 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 bet-

[13] https://knowyourmeme.com/memes/kylo-rens-more-more

Method	MultiOff	Memotion 3 (A)	Memotion 3 (B)	FigMemes	MEMEX	MAMI (A)	MAMI (B)
Majority	37.92	21.5	72.59	5.72	40.7	33.33	18.2
Best previous: text only	*54.0*	NA	NA	34.06	76.4	NA	NA
Best previous: vision only	24.0	NA	NA	*47.69*	69.8	NA	NA
Best previous: vision+text	50.0	*33.28*	*74.74*	46.69	*81.2*	*83.4*	*73.1*
TLC_{Text}	51.83	35.4	77.6	21.14	46.25	61.86	35.93
$\text{TLC}_{Templates}$	**61.89**	**37.77**	79.89	**29.8**	44.99	69.24	39.99
$\text{TLC}_{Templates+Instances}$	58.58	37.04	**80.49**	28.97	**48.14**	**70.0**	**40.21**

Table 1: Classification results for the best performing version of TLC (**in bold**) compared against the best performing method from related work (*in italics*). *Instances* refers to template examples from the KYMKB. See Appendix A.8 for additional results and TLC hyperparameter configurations.

ter representation of the exact meaning by using additional, meme-specific information from OCR.

I Love Democracy[14] Voting with multiple neighbors improves the performance only in cases where we consider both templates and examples. We searched over one to five neighbors in the KYMKB to try to estimate the template of a novel meme and we found three or five neighbors to result in the strongest classification signal for $\text{TLC}_{Templates+Instances}$. This is consistent with our exploratory data analysis, where querying multiple templates in the KYMKB gave us enough information to interpret a new meme that amalgamated two templates (see Section 4 and Appendix A.8 for details). When we consider templates only or text only, this naturally results in instances of multiple distinct templates and a noisy label.

Examples? Well yes, but actually no[15] The base template is sufficient to encode meme knowledge and is more efficient than also embedding examples. In the case of MultiOff, we see a boost of more than two points when we only consider templates and we ignore the examples. In most other cases, $\text{TLC}_{Templates}$ is within one point if not higher than $\text{TLC}_{Templates+Instances}$. This creates a strong model grounded in meme knowledge by encoding only one-tenth of the available images, supporting our claim that meme datasets can be instances of the templates collected in the KYMKB.

Counting templates: GG EZ[16] 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. By harnessing the power of templatic memes, we get multilinguality without even trying. For FigMemes, we note that the TLC is competitive with or stronger than the text and the vision baselines, respectively, as reported in their work (see Appendix A.6). The performance varies greatly across methods and modalities, emphasizing the difficulty of the task.

TLC? Sounds good, doesn't work[17] There are a number of reasons why TLC does poorly on certain tasks. First, many meme datasets are created via crawlers and not curated to remove non-memes, containing both memes and images. This can be empirically verified by looking into the datasets or reading the paper. For example, Figure 1 in the FigMemes paper shows an example of a visual metaphor/simile, but this is a picture, not a meme (see (f)). Similarly in Figures 7 and 8 in Appendix, we see that FigMemes contains comics or visual jokes/puns. Similarly, in Figure 1 of the MAMI paper, all example memes are not templatic memes and can be understood even by those without knowledge of memes or templates. TLC assumes that novel memes belong to a template, but our prediction has no meaning for a picture (which is not a meme). We note, however, that while TLC does not perform as well as results reported in Zhang and Wang (2022), our method is competitive with Hakimov et al. (2022) on Task A, who report a macro-averaged F1 of 73.1.

MEMEX asks a question that only MEMEX can answer. Rather than pairing a meme with a label, MEMEX creates a new task with a meme and

[14] https://knowyourmeme.com/memes/i-love-democracy
[15] https://knowyourmeme.com/memes/well-yes-but-actually-no
[16] https://knowyourmeme.com/memes/ez-ez-clap
[17] https://knowyourmeme.com/memes/sounds-good-doesnt-work

an explanation pair, where the label is whether or not an explanation is relevant for a meme. Their work recognizes that additional context is needed to understand memes, but the approach cannot be applied to memes not contained within their dataset as it relies on having an explanation. Their scraping process for gathering memes and explanations is not detailed, making it unclear how one would explain a novel meme. Moreover, the context they add may be about the idiosyncrasies of a single meme, but this is not a general resource that can be used to aid in broad meme understanding.

Wait it's more complex? Always has been[18] TLC's strength and simplicity point to a problem in the creation of meme datasets. By only taking the most common label for a given template, we assume a template can only convey a fixed message; in the case of a classification task, this means a template can only ever be harmful or not harmful, for example. 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 deliberately over-fitting to the majority class, TLC is naïve but competitive as compared to far more expensive methods. This demonstrates the power of meme templates, but by design TLC is incapable of interpreting novel templates and is instead exploiting the manner in which meme datasets are created.

SOTA meme classifier? I missed the part where that's my problem[19] It is not a goal of this work to create a state-of-the-art meme classifier, but rather a resource for furthering meme understanding and to add the missing piece of meme templates to the literature. If we were testing the generalizability of the methods, a template instance would not be present in both the training and the test split of a dataset. TLC exploits this and, in a sense, is cheating by taking advantage of leaked information; it has not learned to interpret memes, but is instead exploiting the template signal that has been neglected in the literature. Templates should be carefully considered when constructing meme datasets to determine whether a model is learning to understand memes or is relying on frequency alone. We have shown that by fixing the interpretation of a meme based on a naïve and tenuous relation to a meme template, we can rival expensive methods. However, TLC does not actually understand memes (see Appendix A.1 for a discussion on the nature of memes).

[18] https://knowyourmeme.com/memes/wait-its-all-ohio-always-has-been

[19] https://knowyourmeme.com/photos/2190982-bully-maguire

7 Conclusion and Future Work

We have created and released the KYMKB, currently composed of more than 54,000 images and 5,200 base templates with detailed information about each template. To demonstrate the power of templatic memes and their semantics, we have conducted detailed 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, such as the foundational meaning of templates. To demonstrate this, we proposed TLC, an inexpensive, majority-based classifier and found it competitive with far more expensive methods. Existing works simplify memes by assuming they are static images sometimes with text, and despite this, we are still not training methods capable of interpreting them. We have shown we can be competitive with more expensive methods by counting and fixing the semantics for a given template. We have therefore asserted that we must consider meme templates, which our work makes possible. We feel this is the first step on a long road toward intelligent systems that understand memes.

The distinction between different meme types is not always clear and arguably subjective. In future work, we will use the KYMKB to develop a taxonomy of memes in order to aid the development meme-aware systems. Finally, the ability of large languages models (LLM) to understand memes remains untested. We believe the KYMKB makes it possible to examine LLM meme understanding in a simple and systematic manner by accessing information found in our database.

8 Limitations

KYM is in our view the best resource for meme-related knowledge, but this does not mean 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. The 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.

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.

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.

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

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.

Yuhao Du, Muhammad Aamir Masood, and Kenneth Joseph. 2020. Understanding visual memes: An empirical analysis of text superimposed on memes shared on twitter. 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.

Sherzod Hakimov, Gullal Singh Cheema, and Ralph Ewerth. 2022. TIB-VA at SemEval-2022 task 5: A multimodal architecture for the detection and classification of misogynous memes. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 756–760, Seattle, United States. Association for Computational Linguistics.

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.

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*.

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.

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*.

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.

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.

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*,[20] a template that references a popular YouTube video[21] where a player in World of Warcraft[22] 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.[23] 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 6.1, 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-

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

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

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

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

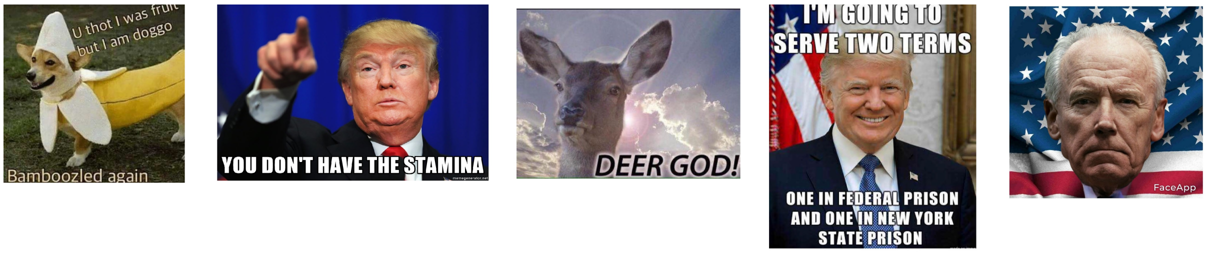

Figure 5: Examples of non-template images found in FigMemes.

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*[24] and language from the *Cheezburger*[25] templates. The second references Donald Trump,[26] who has become a meme in and of himself. This reference is specific to a debate between Hilary Clinton and Donald Trump,[27] but this reference is to emphasize the poster's intended meaning with humor. The fourth example is a meme that can be understood by recognizing that the picture is meant to disambiguate the pronoun *I* in the text. While some of the examples we consider do make references to templates, we believe that they are not instances of templates and can be interpreted without knowledge of a specific template.

A.3 Scraping details

We use the Wayback Machine[28] (WM) to adhere to KYM's terms of use.[29] 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).

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

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

[26] https://knowyourmeme.com/memes/people/donald-trump

[27] https://www.youtube.com/watch?v=3307jg50FjE

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

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

A.4 More template-meme analysis

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*,[30] 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 Wojak in the *I Support the Current Thing* meme template,[31] [32] a template that criticizes social me-

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

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

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

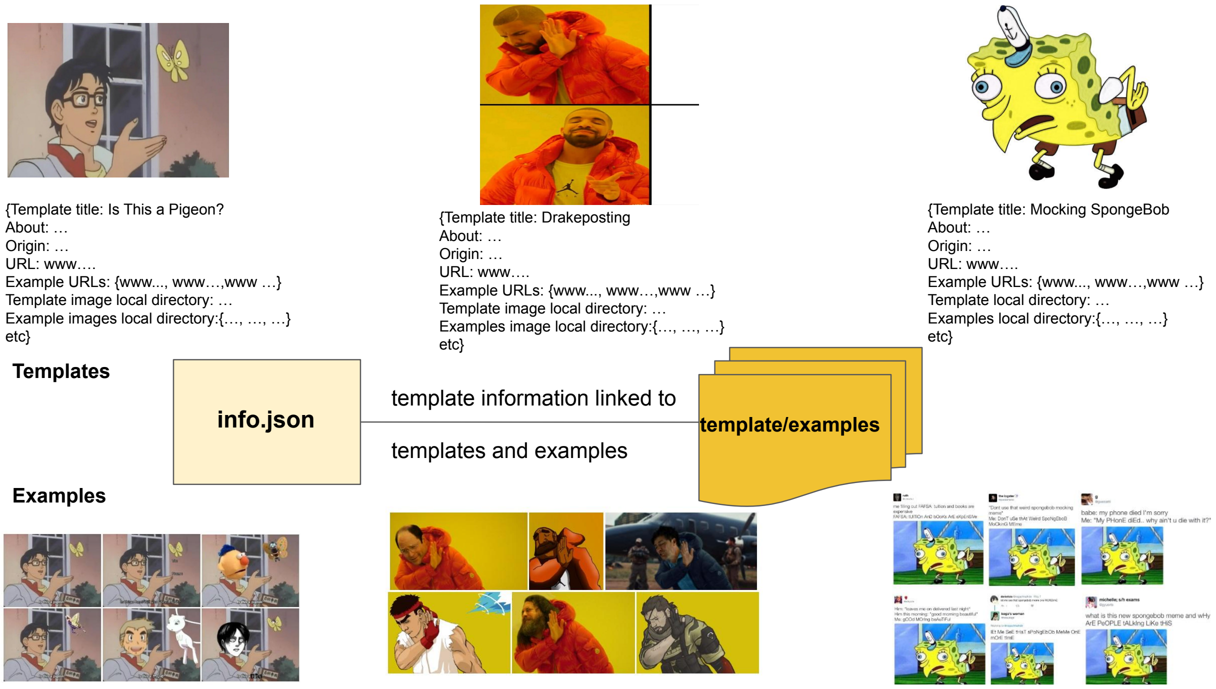

Figure 6: The KYMKB records all textual information about a meme in a .json file, including the text found on KYM, the URLs used in the scraping process, and local locations of all template and example images in the knowledge base.

dia 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*.[33] 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.[34] 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 2). Here, for conciseness, we consider only templates and FigMemes, as we consider it a difficult dataset and it has the most labelsl; 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 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

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

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

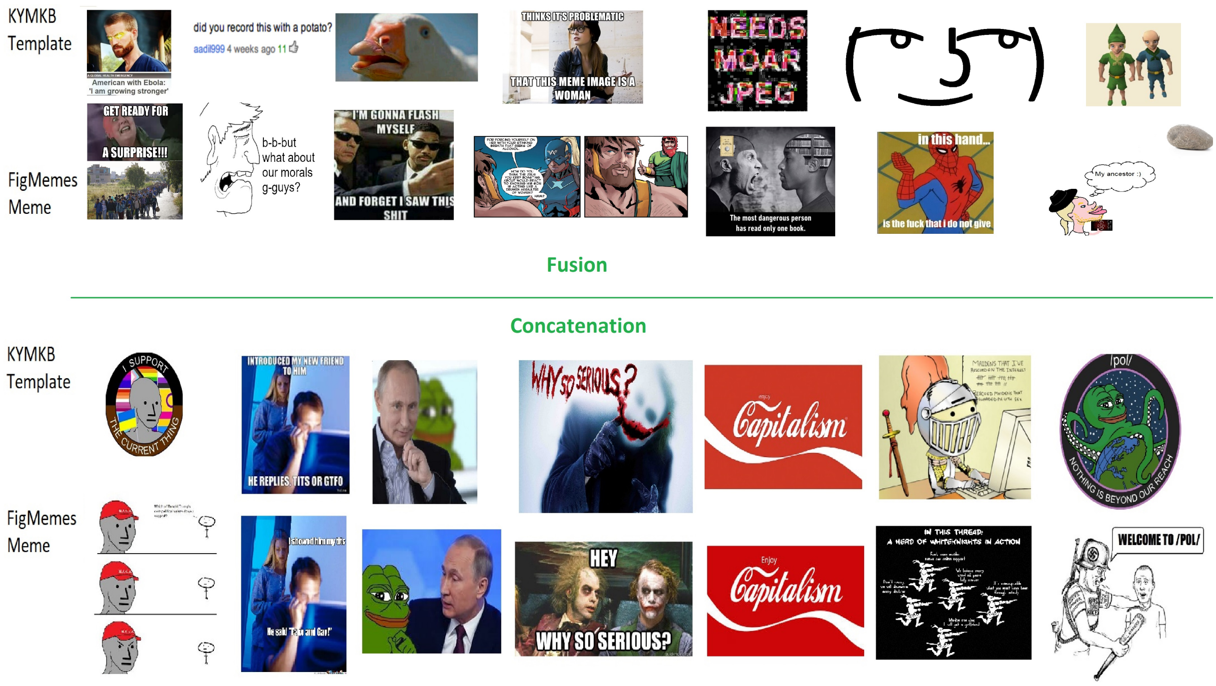

Figure 7: KYMKB templates matched via similarity search to FigMemes images.

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.[35] 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*,[36] which can involve posting an image of Rick Astley from the Never Going to Give You Up music video,[37] 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.[38] We would argue this is an intertextual meme instance referencing the Rickroll template.

Loss[39] is another famous template where an instance is an action, not an image. The template is a reference to the Ctrl+Alt+Del Comic[40] 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, 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

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

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

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

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

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

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

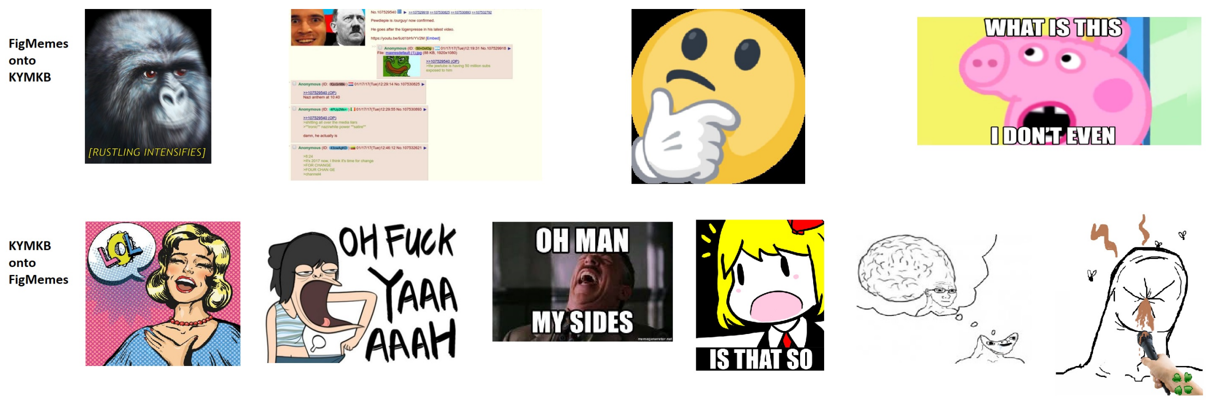

Figure 8: In the first row, we show the templates closest to seven KMeans centroids fit on the FigMemes, while in the second row, we show FigMeme images closest to seven centroids derived from KYMKB. We combine the text and the image representations by normalizing and averaging the two modalities. This results in multiple centroids close to the same meme/template.

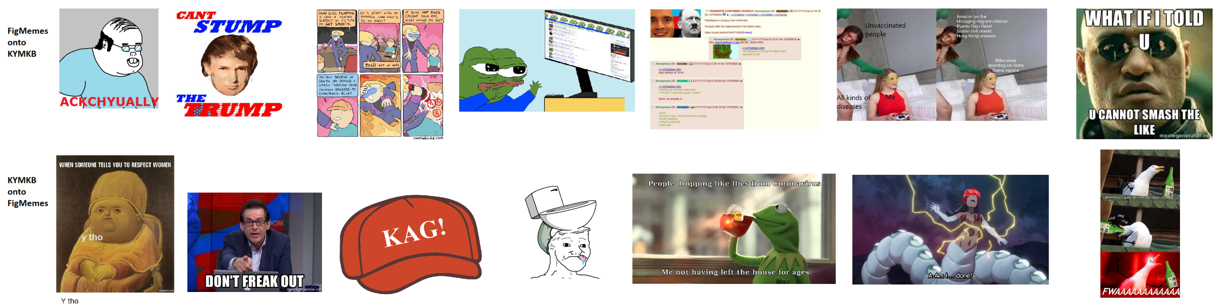

Figure 9: In the first row, we show the templates closest to seven KMeans centroids fit on the FigMemes, while in the second row, we show FigMeme images closest to seven centroids derived from the KYMKB. We only use the image modality, which results in seven distinct images.

were instances of the *Loss* template (see Figure 10).

Instances of the *Planking*[41] 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*.[42] 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.[43] 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.[44] 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 examples, this would mean often backing off to the most majority class in the dataset because we will find distinct templates. Alternatively, we could keep all labels for a given template and then reduce to its most frequent label, which we refer to as *label vote*. We consider all cases. We find that the tem-

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

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

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

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

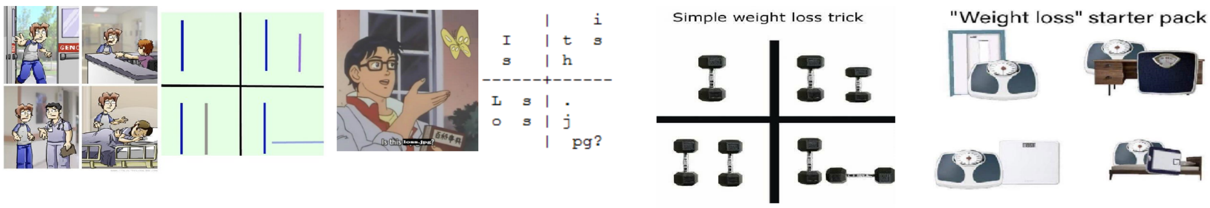

Figure 10: The first image is the original template of *Loss*, while the other three images are *Loss* instances, all of which are visual puns that cannot be understood without knowing the original template. The second image is another intertextual meme where *Loss* and *Is This a Pigeon?* have been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

plate 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 Dataset information

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 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).

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.

FigMemes is a multilable 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.

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.

Finally, MEMEX is a binary task, baseless (30%) vs. valid (70%), of whether or not a explanation

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 2: Summary of the datasets we use in our experiments.

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. At time of writing, their validation split is not public.

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 6.1. 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,[45] 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 TLC$_{About/OCR}$ is only present in cases where the modalities are not combined, because in the other cases text embeddings are combined with the template or the meme embeddings.

We find that ViT-L/14@336px usually results in the strongest performer, 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.

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

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 3: 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 4: 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 5: 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 6: 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.