Web and Social Network Analysis Project

16/06/2025

Ashley Andrea Squarcio 512909
Pietro Saveri 524921
Matteo Salami 513974

Memescape of Science: A Multi-Modal Network, Content, and Visual Analysis

Objective

In an age where misinformation travels faster than ever, memes, often dismissed as mere digital amusement, emerge as a surprisingly potent vessel for both knowledge dissemination and distortion. They straddle the boundary between humor and persuasion, leveraging brevity and shareability to convey ideas with remarkable efficiency. By focusing on the subset of memes grounded in science, we aim to uncover the nuanced dynamics by which scientific concepts are communicated, and occasionally misconstrued, through this visual-verbal hybrid. In particular, our project revolves around three central questions: modes of scientific framing, community engagement and diffusion, and potential correlations between template and ideology.

As for the first, we explore whether science-themed memes tend toward seriousness, cynicism, conspiracy, or absurdity. While humor is intrinsic to the meme format, our focus is on how levity can mask, or reveal, underlying narratives about scientific authority, skepticism, or wonder.

The second aspect tackles meme proliferation via networks of users whose interests and beliefs shape what garners traction. We investigate who shares and interacts with science memes: are there any identifiable groups, and do their engagement patterns differ? Mapping these communities will illuminate the social vectors through which scientific ideas (and misideas) spread.

At last, we conjecture certain meme templates may carry their own ideological baggage. We ask whether specific formats systematically co-occur with particular framings or topics. Unraveling these template-content alignments offers insight into the visual shorthand that underpins how audiences interpret and propagate each meme.

Together, these questions offer a holistic view of the science-meme lifecycle: from initial creation and framing, through community-driven diffusion, to the subtle interplay between form and message. By zeroing in on scientific content, we keep our analysis targeted, rigorous, and directly relevant to debates about public understanding of science.

Data

To capture the diversity of science memes, we turned to Reddit, where niche communities flourish and data collection is both robust and reproducible. Using **PRAW** (the Python Reddit API Wrapper), we initially searched broadly for “science memes” across Reddit, which naturally foregrounded r/sciencememes. Recognizing the value of both deep-domain and general-interest contexts, we expanded our scope to include domain-specific subreddits (e.g., r/chemistrymemes, r/physicsmemes), broadly themed meme communities (r/memes, r/dankmemes, r/politicalcompassmemes), and even conspiracy-leaning spaces (r/conspiracy, r/conspiracymemes, r/bestconspiracymemes). This multi-faceted approach ensures that our dataset reflects the full scope of discourse, from earnest scientific humor to conspiratorial takes.

Reddit's API limitations posed a challenge: single-listing queries (e.g., "hot" or "new") yield only a fraction of available posts. To address this, we exploited multiple "listing sorts" (new, hot, top, rising) for science-specific subreddits and multiple "search sorts" (new, hot, top, relevance, comments, rising) for general meme communities. By intersecting these queries and de-duplicating overlaps, we maximized coverage without manual moderator access to Pushshift. To further ensure topical relevance, general subreddits were pre-filtered via Boolean keyword searches ("science," "scientist," "vax," or "climate"), thereby excluding off-topic memes while preserving diverse scientific perspectives.

Our final corpus comprises 6,411 static-image posts, each annotated with post ID, subreddit, author, title, self-text, URL, timestamp, score, comment count, and flair. Although posts from dedicated science subreddits naturally outnumber those from broader or conspiracy forums, this imbalance mirrors genuine community sizes and keyword-filtering effects rather than researcher bias. Importantly, our downstream analyses emphasize **relative distributions** and **network structures**, rather than raw counts, so emergent patterns reflect authentic user behavior, not subreddit quotas.

Recognizing that audience reaction is as critical as original posts, we also harvested **all nested comments** (with no depth limit) for each meme, which demanded careful pacing to respect server load. Together, posts and comments form a rich, multi-layered dataset, ready for the network, content, and template analyses that follow.

REPORT SECTION 1: Social Content Analysis

Main idea

After creating the df_memes we can finally start retrieving the necessary text to perform the different analysis to answer our question.
The title and selftext of the memes are not enough to really capture their essence, as many of today's memes focus more on the image itself and the text inside it. This only means one thing: we need to perform OCR on memes.

How-to?

Our first approach was to use one of the most famous tool available: *PyTesseract*, as this usually works well, it is fast and easy to setup. After preprocessing the image into greyscale for better results, we tested it on different Memes from our df and discovered that PyTesseract lacks strength when dealing with different fonts and complicate texts.
To fix this error we changed approach and tried PaddleORC, slightly less known and more difficult to setup, using this tool we manage to process also difficult fonts, thanks to the different parameters offered by the library we could also apply a more robust image preprocessing.

PaddleORC can be a good solution as it works quite well. But memes are not always clear and easy to read, sometimes there are hand-written sentences, phone screenshots with lots of useless text not relevant to the meme itself and this caused PaddleORC to also fall into some problems.
Is possibile to find some output example in the notebook: Content_Analysis.ipynb

After searching and reading some papers (https://blog.roboflow.com/best-ocr-models-text-recognition/) to understand which are the best ways to perform ORC nowadays we ended up in a easy and not very surprising discovery: AI.

Multimodal Language Models are now capable to perform the best OCRs, since they can handle images, think and return structured output we had to choose them.
We need a model that is cheap and fast. After a deep dive into "Best OCR Models for Text Recognition in Images" the answer was clear: **Gemini 1.5 flash.**

We really wanted to put it at test and presented really difficult memes, the result were astonishing.
Is possibile to see some output example in the notebook: Content_Analysis.ipynb

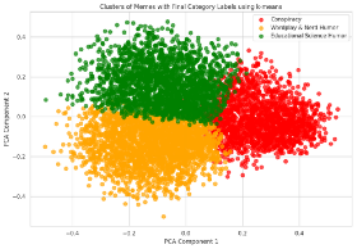
Obviously when dealing with LLMs the prompt is undoubtedly the most important aspect, we told him to return only the text important to the meme, not useless text is needed, in this way if a meme contains lots of text we only select what is important. Is possible to find the prompt in the section Approach: 2 GEMINI in the notebook: Content_Analysis.ipynb

Clustering and Label assignment

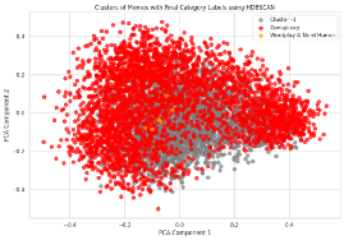
Now that we have our df ready with the title, selftest and OCR text we can concatenate everything and start to analysis it.
Our first approach was to divide the memes into clusters and see how this different clusters potrays science.

We used the all-roberta-large-v1 model to create the embeddings for our clustering analysis. We chose this model not only because it is one of the best state-of-the-art sentence embedding models available, but also because it is particularly well-suited for our data for several reasons: Contextual Understanding, Robustness Across Domains, Sentence-Level Embeddings, Pretrained on Large, Diverse Datasets.

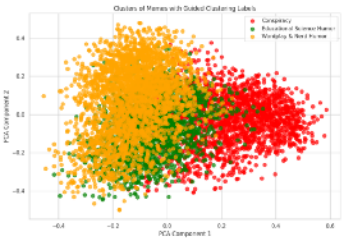
We tried a 4 different clustering algorithms to see which was better, to identify distinct groups of memes.



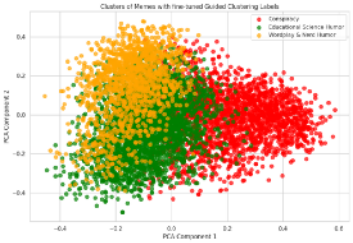
Approach 1: K-Means clustering with silhouette analysis
This is a pretty straightforward approach, we used a set of potential k, by applying the silhouette score, we where able to find the best k = 3.
This approach was a good start but the silhouette scores are very low, this could imply that the division is not well done.



Approach 2: HDBSCAN
Using HDBSCAN everything is automatic, it finds the best number of clusters by itself.
This approach does not work, since the text are difficult to classify, HDBSCAN assigns a lot of them as noise, then craters only 2 clusters that are very unbalanced. To ensure an interesting and robust analysis we need not only more classes but also more balanced.



Approach 3: Guided (Semi-supervised) clustering
In this approach, we used a semi-supervised clustering strategy to guide meme categorisation based on a small set of manually defined seed examples for each category. To embed both the seed texts and the memes, we used the s**entence-transformers/all-mpnet-base-v2** model, chosen for its strong performance in capturing semantic similarity across short, informal texts like memes. We computed the embeddings for the seed texts and averaged them within each category to obtain a set of category centroids. Meme texts were then embedded using the same model, and each meme was assigned to the category with the highest cosine similarity to its centroid.
This approach worked quite well, and we could have gone with that one, but we really wanted to try another method just to see if the performance was even better.



Approach 4: Fine-tuning `all-mpnet-base-v2`
To improve meme clustering quality, we fine-tuned the **sentence-transformers/all-mpnet-base-v2** model using a synthetic dataset generated via Gemini. This dataset consisted of 6,741 sentence pairs labeled for similarity: 3,141 positive examples (belonging to the same conceptual cluster) and 3,600 negative ones (from different clusters). Post-training, we uploaded the model to Hugging Face (here) and used it to embed new meme texts. Scores against predefined cluster centroids were computed using cosine similarity, as in Approach 3.
For example, the text: “5G was made from the goverment to kill us!!”
Received scores of:
Conspiracy: 0.639
Wordplay & Nerd Humor: 0.184
Educational Science Humor: 0.244

Sentiment Analysis:

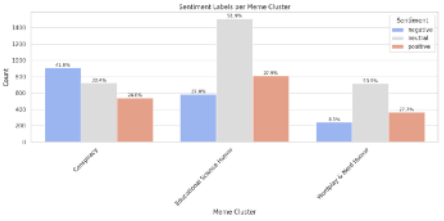
To perform sentiment analysis on the meme text, we tested two approaches covered during the lectures: Afinn and VADER.
For both models, we used minimal preprocessing, as our goal was to keep the original text structure as intact as possible, especially considering the informal and humorous tone typical of memes.

- **Afinn** does **not support emojis** or more complex syntactic structures. When encountering an emoji, Afinn simply skips it. This means we do not need to remove emojis from the text; instead, we moved emojis to the end of the sentence during preprocessing. This small adjustment ensures they don’t break the token structure while still preserving them for potential future analysis or for VADER.

- **VADER** (Valence Aware Dictionary and sEntiment Reasoner), on the other hand, is specifically designed to handle social media text, including emojis, punctuation, complex humorism, and slang.

We applied both models to each meme’s concatenated text and compared the resulting sentiment scores. VADER generally produced more nuanced and expressive results, especially when dealing with emojis, informal expressions, or punctuation emphasis. For instance, memes with humorous text followed by laughing emojis were more accurately interpreted by VADER, whereas Afinn tended to assign more neutral or flat scores due to its limited lexicon and inability to process such context.

Afinn

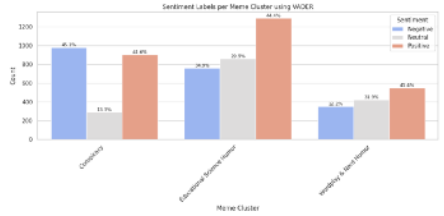


Visualisation
We used bar-charts to visualise the sentiments for each label. This allowed us to identify whether specific clusters tended to be more positive, negative, or neutral.

The sentiment distribution across meme clusters, as labeled by the Afinn lexicon, reveals patterns that are somewhat flattened and less expressive. For instance, in the Wordplay & Nerd Humor cluster, typically rich in irony, exaggeration, and emoji use, a majority of memes were classified as neutral (53%), with comparatively low proportions of both negative (8%) and positive (27%) sentiment. Similarly, Educational Science Humor was also dominated by neutral labels (51%), despite often containing witty or uplifting content.

Interestingly, the Conspiracy cluster shows the highest share of negative sentiment (42%), which aligns with expectations, but the relatively even distribution across sentiment types suggests limited sensitivity to tone subtleties. This could indicate an underrepresentation of emotionally charged or sarcastic nuances that are often present in meme culture but hard to detect with strictly lexicon-based approaches.

VADER



An important observation we made was that VADER returned significantly fewer "neutral" results compared to Afinn.

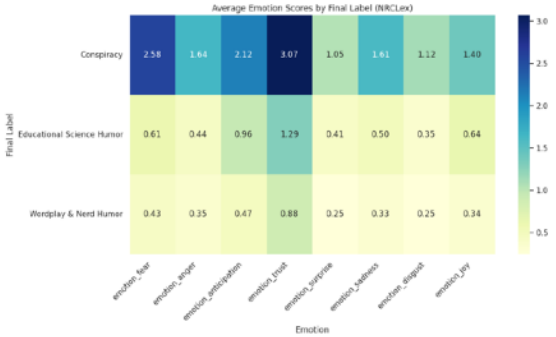
As in this plot we can see how sentiment varies significantly across the different meme clusters, revealing clear emotional patterns in how scientific topics are framed.

Conspiracy memes are largely negative in tone (45%), reflecting skepticism or mockery toward science, with minimal neutral content (13%), indicating a preference for emotionally charged expression. In contrast, Educational Science Humor memes show a more balanced sentiment, leaning slightly positive (44%) but still containing some criticism or satire. Wordplay & Nerd Humor memes are the most positive (41%) and least negative (12%), with the highest neutral share (31%), suggesting a lighthearted, emotionally moderate approach to science.

Overall, this cluster-level analysis confirms that VADER effectively captures emotional nuances in meme text, helping distinguish between hostile, supportive, and neutral tones across meme types.

Emotion Recognition:

To perform emotion recognition we first used a more classical approach: NRCLex. Starting from the fact that this library does not process emojis and does not understand demojize text we had to perform a more robust preprocessing. Removing the emojis and using an aggressive leammatization to increase the probability of match with the NRCLex lexicon. We did not remove the stop words to not loose important emotion alerts like “not”.



Despite the preprocessing efforts, NRCLex still presented notable limitations. Its lexicon-based approach means it relies heavily on exact or near-exact word matches, making it less effective in handling informal language, creative phrasing, or context-dependent expressions common traits in meme texts. This can lead to underestimation or misclassification of emotions, especially when the emotional tone is implied through sarcasm or non-standard grammar.

The average emotion scores extracted with NRCLex show some distinctions across meme clusters. The Conspiracy cluster stands out with significantly higher scores across almost all emotions, particularly trust, fear, and anticipation. This suggests that these memes often contain emotionally loaded language, possibly reflecting alarmist, persuasive, or speculative tones typical of conspiratorial content.

In contrast, both Educational Science Humor and Wordplay & Nerd Humor exhibit much lower overall emotion scores. Their emotion profiles are relatively flat, with modest peaks in trust and anticipation, hinting at more informative or playful rather than emotionally charged content.

Interestingly, despite being humorous in nature, the Wordplay & Nerd Humor cluster receives low joy scores, likely due to NRCLex's limited ability to detect affect in pun-based or ironic expressions.

Transformer-Based Emotion Classification
To address the limitations of lexicon-based approaches like NRCLex, we adopted a more robust solution using a transformer-based emotion classification model. Specifically, we used the **bhadresh-savani/ distilbert-base-uncased-emotion** model. This choice was motivated by the model's strong performance on emotion detection tasks, its fine-tuning on the GoEmotions dataset a diverse and richly annotated dataset of English Reddit comments and its ability to detect nuanced emotions beyond simple sentiment categories.

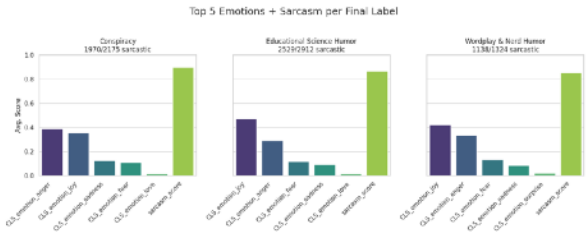
One major advantage of this model is its flexibility in handling real-world, unstructured text. It was trained on raw, user-generated data and can natively process emojis, informal phrasing, and punctuation-based expressiveness. As such, minimal preprocessing was required: we skipped lemmatization, stopword removal, and emoji filtering entirely.

However, since memes often convey meaning through humor, irony, and contradiction, we recognised the need to augment the emotion detection pipeline with sarcasm awareness. **Sarcasm** can heavily distort the literal meaning of text, leading to misleading emotion predictions if not properly accounted for. To address this, we incorporated a sarcasm detection model to refine the emotional interpretation of meme text.

For sarcastic detection we took the opportunity to revamp an old model we fine-tuned: **PietroSaveri/ Sarcastic_01** (here). This model is based on distilbert-base-cased and was trained on the News Headlines Dataset For Sarcasm Detection.
Unlike the emotion model, sarcasm detection required more classical NLP preprocessing, including: expanding contractions, removing punctuation, tokenization, stopword removal, lemmatization.

Combining Emotion and Sarcasm Models
To combine these models, we created a pipeline that processes each meme's text through both classifiers. Emotion scores are obtained using the emotion model, which outputs a confidence value for each emotion category.
In parallel, the meme text is also passed through the sarcasm detector. If the model assigns a high probability of sarcasm, we apply a sarcasm-aware adjustment to the emotion scores. Specifically, if the most likely emotion is neutral, we reduce its score and promote the second-best emotion by a small factor. This correction is based on the intuition that sarcastic expressions often mask real emotions behind a superficially neutral or ambiguous tone.

Results Analysis: Transformer-Based Emotion + Sarcasm Detection



This plot presents the average emotion scores along with the sarcasm scores across our 3 clusters. Compared to NRCLex's flatter and less differentiated outputs, these results offer a richer and more context-aware emotional landscape.

One of the most striking features of the plot is the consistently high sarcasm scores across all clusters.

This underlines how meme communication often leans on sarcasm and irony. By explicitly modeling sarcasm, the transformer approach avoids misinterpreting literal meanings and can adjust emotion

predictions accordingly. For instance, a sarcastic meme saying “Great job, flat-earthers!” might express anger or disbelief.

Conspiracy memes score highest in anger, joy, and sadness, reflecting their tendency to mix outrage with mocking or emotionally charged content. The blend of anger and joy might seem contradictory, but in the context of sarcasm, this makes sense.

Educational Science Humor shows elevated joy as the dominant emotion, followed by anger and fear. This mix reflects a tone that is both enthusiastic about science and frustrated by its hard nature.

Wordplay & Nerd Humor features high joy, moderate anger, and some fear/sadness but importantly, this cluster sees lower sarcasm scores than the others (though still high). This suggests more wholesome or light-hearted humor, which matches the tone of pun-based or logic-heavy jokes that don’t necessarily rely on ridicule.

Topic Analysis

To explore the dominant themes within each meme category, we applied topic modeling using BERTopic separately for each final label. This per-label approach allowed us to uncover nuanced subtopics specific to individual meme clusters. BERTopic generated interpretable topic clusters along with relevance scores, enabling us to examine the internal structure of each meme category.

These topics were visualised using interactive plots, which made it easy to explore the most frequent themes and their associated keywords per label.

Here we put a link to site that we built to better visualise them:
Topic analysis visualisation

REPORT SECTION 2: Social Network Analysis

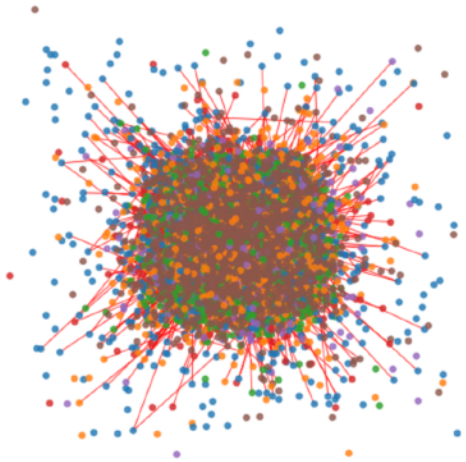
Network Analysis: Models

Centrality Measures

To quantify user influence and structural importance within the author–commenter graph, we computed three complementary centrality metrics, namely **degree centrality, betweenness centrality,** and **closeness centrality**. Following the order, the first counts the number of **direct connections** each node has. High degree indicates users (authors or commenters) who interact with many others. Betweenness centrality, on the other hand, measures the fraction of shortest paths that pass through a node. Users with high betweenness serve as **bridges** between different parts of the network, facilitating information flow. Finally, closeness centrality accounts for the inverse average shortest-path distance from a node to all others. High closeness indicates users who can **rapidly reach** (and be reached by) many others, reflecting efficient access to the broader network.

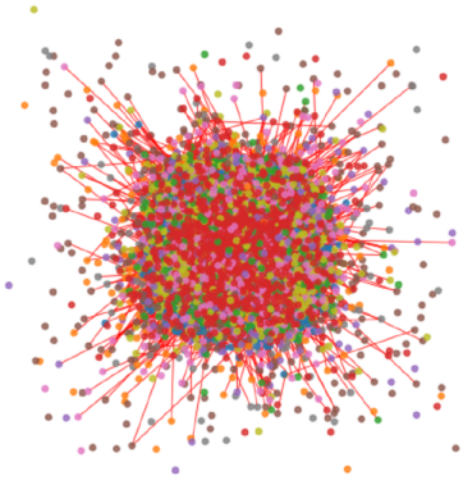
By combining these three measures, we capture **different facets of “influence”**: volume of interactions (degree), brokerage (betweenness), and network-wide reach (closeness). Extracting the top five users for each metric allows us to spotlight the key actors driving science-meme diffusion.

Community Detection



Louvain Communities (6): Inter-Community Edges View

To uncover the hidden structure of interactions around science memes, we built an **author–commenter graph** in which each node represents a Reddit user (either as a meme author or as a commenter) and each edge indicates at least one comment interaction on a specific post. To partition this graph into meaningful communities, we applied two complementary algorithms: Louvain and Fluid Communities. The **Louvain algorithm** efficiently uncovers densely interconnected groups by maximizing **modularity**, a measure of how much more tightly nodes link within communities than between them. In our context, Louvain pinpoints clusters of users who repeatedly engage around similar memes or themes, offering stable, hierarchical communities that reflect core audiences for different scientific framings.



FluidC Communities (9): Inter-Community Edges view

Conversely, **FluidC** propagates "fluid labels" across the network to form communities of roughly equal size, capturing more ephemeral or diffuse interaction patterns. This is especially useful in Reddit environments, where participation can be **uneven** and **fast-moving**. FluidC may reveal peripheral or cross-subreddit groups that Louvain's more cohesive clusters might overlook.

By running both algorithms, we ensure a robust view of the network: Louvain highlighting the core, tightly knit communities and FluidC bringing to light broader, more fluid interaction patterns.

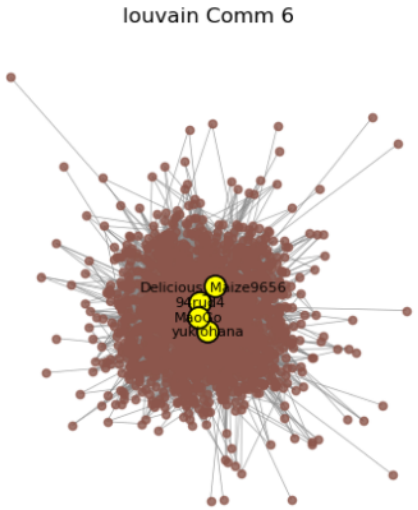
Finally, by exploiting the previously performed content analysis, we enriched each user node with three labels: **network community** (via Louvain), **content cluster** (e.g. Conspiracy, Educational Humor, Nerd Humor), and **emotion profile** (sentiment scores, sarcasm, dominant emotion). Aggregating these labels per community lets us see how **structural groups** correspond to **meme themes** and **emotional tone**.

Network Analysis: Results

Top Influencers

After ranking users by each centrality measure, we identified the top five nodes for degree, betweenness, and closeness. Strikingly, **most of these high-centrality users**, regardless of the metric, **reside in Community 6** (as defined by our Louvain partition).

This concentration of influential nodes in a single community underscores Community 6's pivotal role in both **content creation** and **cross-community information flow**, marking it as a strategic target for understanding, and potentially guiding, the spread of science memes.



4 out of the top 5 Closeness Centrality Nodes are located in Community 6

Other SNA metrics

After community detection, we measured three key network metrics, **assortativity**, **local** and **global clustering coefficients**, and **density**, to characterize the overall and in-community structure. All values point to a **content-centered**, broadcast-style interaction model.

Degree Assortativity A negative assortativity (-0.0923) implies that high-degree nodes (popular meme authors) tend to connect with low-degree nodes (one-time commenters), rather than forming "elite" clusters. This **hierarchical pattern** is typical of information diffusion networks, where content creators broadcast to a broad, otherwise unconnected audience.

Clustering Coefficients Both coefficients are very low (0.0140 for global, 0.0402 for local), indicating that neighbors of any given user are rarely connected to one another. In practical terms, comment threads do not form tightly interlinked triangles of interaction: users comment on posts but do not frequently comment on each other's contributions. This confirms that the network is dominated by **star-like**, post-centered structures rather than by dense discussion circles.

Density With fewer than two edges per thousand possible (density of 0.0017), the graph is extremely sparse. This level of sparsity is consistent with episodic engagement around individual posts: most pairs of users never interact, reinforcing the view of a **content-driven rather than socially driven** network.

When these same metrics are computed within each Louvain community, they remain similarly low, negative assortativity, minimal clustering, and low density, confirming that even internally, communities function as **audiences around shared memes**, not as cohesive social groups.

Taken together, these findings paint a coherent picture: science memes on Reddit propagate through a **broadcast mechanism**, with creators acting as hubs and audiences engaging once or sporadically, rather than through dense peer-to-peer discussions.

Leveraging content-network correlations, we were able to answer the core questions: *"who shares and engages with science-themed memes? Are there distinct meme-sharing communities that map to the way science is portrayed?"*

Distinct communities do, in fact, align with distinct portrayals of science. Specifically, we were able to discover that **communities 2 & 4** are very **conspiracy-heavy:** 85–92 % of their shared memes are conspiracy-related. They are also denoted by a general negative sentiment and high sarcasm, and have fear (plus anger/disgust) as their dominant emotion. Conversely, **communities 0, 3 & 5** are leaning towards **educational humor** for the most part (47–67 %, plus some Wordplay). Sentiment ranges from **neutral to positive**, accompanied by moderate sarcasm. The dominant emotion here is **trust**. Finally, **community 1** definitely stands out as the **nerd humor niche,** with neutral-to-positive sentiment and low sarcasm, and no particular dominant emotion.

final_label community	Conspiracy	Educational Science Humor	Wordplay & Nerd Humor
0	0.350852	0.474432	0.174716
1	0.034208	0.091488	0.874304
2	0.842779	0.098720	0.058501
3	0.050553	0.672986	0.276461
4	0.917603	0.056180	0.026217
5	0.114333	0.542670	0.342998

Community x Cluster Distribution

These patterns show that users naturally cluster not only by interaction but also by **thematic focus** and **emotional stance**, confirming that **science portrayal** in memes shapes, and is shaped by, the communities that share them.

REPORT SECTION 3: Template Recognition

The **goal** of this section of the project evolved to be the following: given an image representing a meme, **identifying the template** that the creator used.
The challenges of the task are multiple and various in nature. Going in order - from most evident to less evident - we must address the enormous volume of different meme templates that populate the internet. There is **great variety** in structure, color, characters and object involved and many more. This fact by itself rules out many more classical techniques that have many times been implemented in numerous classification tasks. Labelling all of the millions of possible templates—or even just the visible objects within them—would have been far too time-consuming and, quite simply, impractical.
To elude the issue we first engaged in a zero-shot approach using a retrained model. It is to be noted that the initial goal was to perform object recognition within memes.

The model in question is **CLIP** which was released by OpenAI in 2021. CLIP stands for Contrastive Language-Image Pre-Training and it is a Transformer based architecture trained, from scratch, on image and text pairs that is able to embed both images and text.

For the purpose of template recognition we first exploited the image embedding capabilities, each image was preprocessed using the preprocess method provided by the API and converted to RGB - as expected by CLIP. The next step was to get, for each image, and so for each meme, the **corresponding embedding**. The embedding consists of a **512 dimensional vector** that was carefully stored in a cache for later use, and also to avoid recomputing it.

CLIP's peculiarity is that it was trained to map both the image and text embeddings to the same latent space to permit the use of **cosine similarity** to compute its outputs.
This characteristic was utilised in our first draft of this section of the project in the following way: a list of labels was defined pre-processed and encoded. Examples of labels were "a lab-coat", "a confused expression" "a cartoon character" and again, many more. These labels' embeddings would then, using cosine similarity, compared with the preexistent image embeddings to determine which one yielded the highest similarity score.

This was the moment were a crucial realisation took place, there will never be **enough labels** and they will never be specific enough (because of the great variety mentioned above). An even less supervised approach was needed.

The goal developed with the approach and it finally landed on "template recognition". For each template its own specific denomination. Some examples are (…).
The second approach we explored involved two main stages, embedding through CLIP - which remained unvaried - and **clustering** via **HDBSCAN** which stands for "Hierarchical Density-Based Spatial Clustering of Applications with Noise" which groups data based on density which is in turn based on the **distance between data points**.

To fulfil this aim **dimensionality reduction** needed to be part of the pipeline. Both PCA and UMAP were used in order to discover which one would outperform the other and the parameter "number of components" was interchanged again for performance reasons.
Finally, after fitting the clusterer to the data, UMAP was used again to reduce the dimensions to 2, in order to be able to visualise the formed clusters in a plot.

The idea behind the clustering was to, once the clusters were formed , name each and every one of them manually. The hope was that, in the sample of memes we took in consideration - which were 6400 - the templates would **repeat** allowing the clustering algorithm to function properly.

Reality was, as one could guess, different. Here arose the **second problem** in the list; even when taking thousands of different samples, you will still end up with only few samples that use the same template and which therefore are equal in structure.
This is clearly a great issue when clustering was the ultimate goal.

Moreover HDBSCAN has the ability of classifying noise with a specific label "-1". This helped shed light on a further point at issue; Many data points were classified as **noise** because they don't actually use any template!. Examples are **text only memes**, or pictures made from a phone's camera and the list is potentially infinite in length.

An additional **change of approach** was inevitable at this point. Indeed we opted for something we have known and used for a long time and in a great variety of occasions; a classifier. The idea was to carry out a multi class classification where the classes would be the most used memes, so that we would be able to classify the vast majority of the whole dataset.
An annotation must be made, when beginning to build the classifier we knew that the chance of all the meme templates falling under the 16 arbitrarily defined classes was close to zero. That is why we came prepared with an alternative solutions for the meme template that would remain unnamed, but more on that later.
We chose to use the **resnet18** architecture, an 18 layer deep convolutional neural network that was trained to classify images into 1000 object categories. The action plan required the fine-tuning of the classification head of the model.

The first issue that needed addressing was **finding clean data** to train, or in our case fine-tune the chosen classifier.

To solve this issue we employed the Google search API (**SerpApi**) that would return the 100 first image results given a prompt. The prompt was also used to automatically label the images.
On these 1600 images we also performed data augmentation e.g. rotation and horizontal flip to help prevent the always lurking overfitting problem. We also, for performance comparison reasons, trained a second version of the model to which all the images would be fed after being converted to grayscale. The results showed overall equal performance between the two:

When tested on the training set:	**Accuracy**: 97.63% 0.976271186440678
When tested on the test set:	**Accuracy**: 77.48% 0.7747747747747747

Once the fine-tuning process came to an end we utilised the classifier to actually label the memes in the original dataset. We kept track of the confidence score because, as mentioned before, we knew not all memes would fall in those categories. This is why we implemented a threshold mechanism; all classification where the confidence score was below the threshold were classified as **noise**. Results were stored in a cache to avoid re-computations in future.

We established that the we expected numerous “noise” labels, and we were correct. Many of the memes used rare templates or even no templates at all, some were only text and others had extremely peculiar and unusual structures.

This led us to turn to our backup plan. New state-of-the-art **large language models** were our final solution. In particular we made use of “Gemini 1.5 flash” which, to cite the google written overview, “is a foundation model that performs well at a variety of multimodal tasks such as visual understanding, classification, summarisation, and creating content from image, audio and video.”

We gave it the task of naming every meme template it was fed so that, once it was finished, we could **replace** the noise labels with what the model specified.

A **pivotal** item of this newly introduced procedure was the structuring of a prompt that would be fed to the language model together with the image.
We landed on the following, which we believed addressed all of the challenges mentioned up to this point:

prompt = “""

You are a meme classification expert.

Given an image of a meme, identify the visual meme template it uses and return a short name that uniquely and consistently describes it.

The name should:
- Be 2–5 words
- Refer to what is visually iconic or culturally recognisable
- Match common internet naming conventions if known (e.g., “drake meme”, “two buttons”, “surprised pikachu”)
- If the meme doesn't use a known template, describe the main visual layout or character (e.g., “man with whiteboard”, “knight with arrow”, “woman with math overlay”)
- If the image is only text (e.g., a screenshot, a tweet, or handwriting), return: **”text only"**
- If there is no identifiable meme template or structure, return: **"no template”**

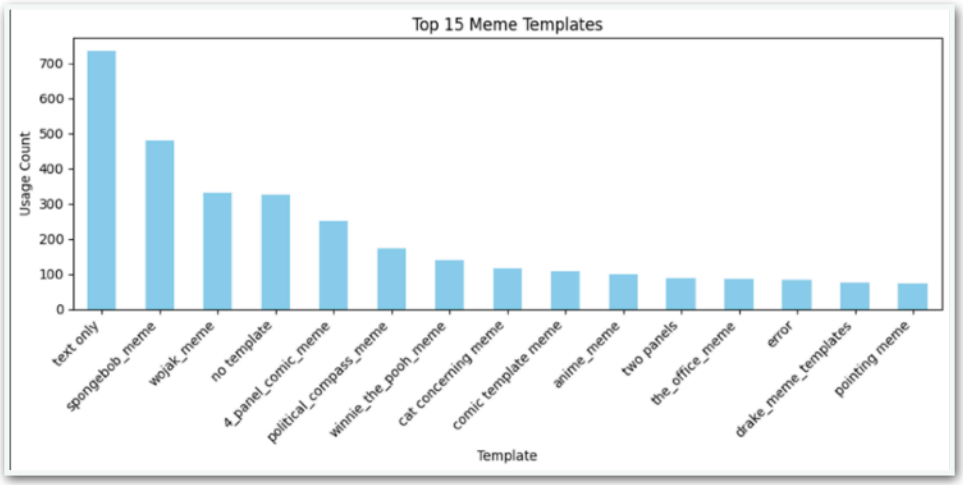
Return only the template name. Do not explain or comment.
Always use lowercase.

“""

The template names were, as they were being generated, stored in a dictionary as values while the filenames, i.e. the post_id, were the keys.

At last the **final data frame was built**. We put together the data that emerged from the section 1 in order to be able to perform some further analysis related to the established template names.

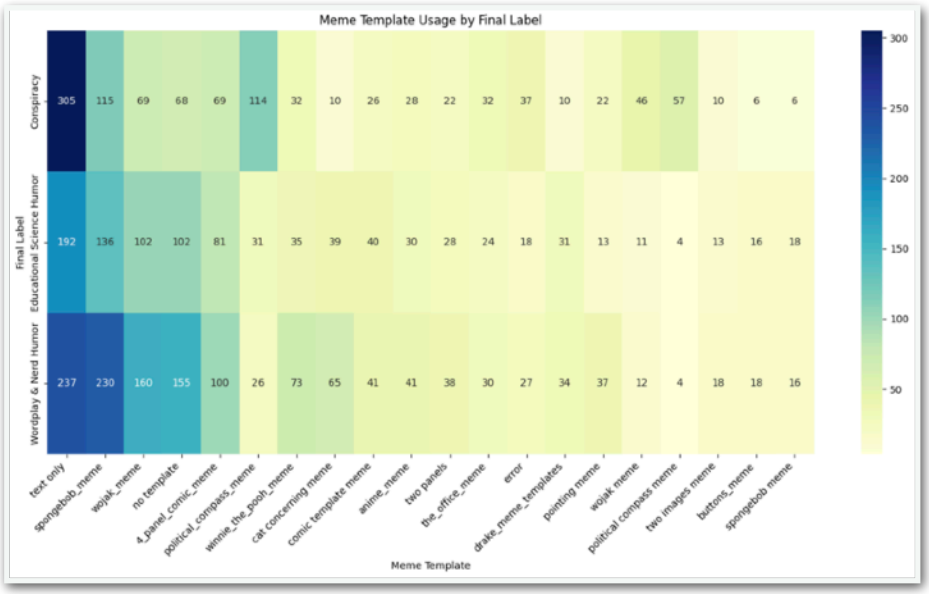
The different **analyses** we accomplished were the following:



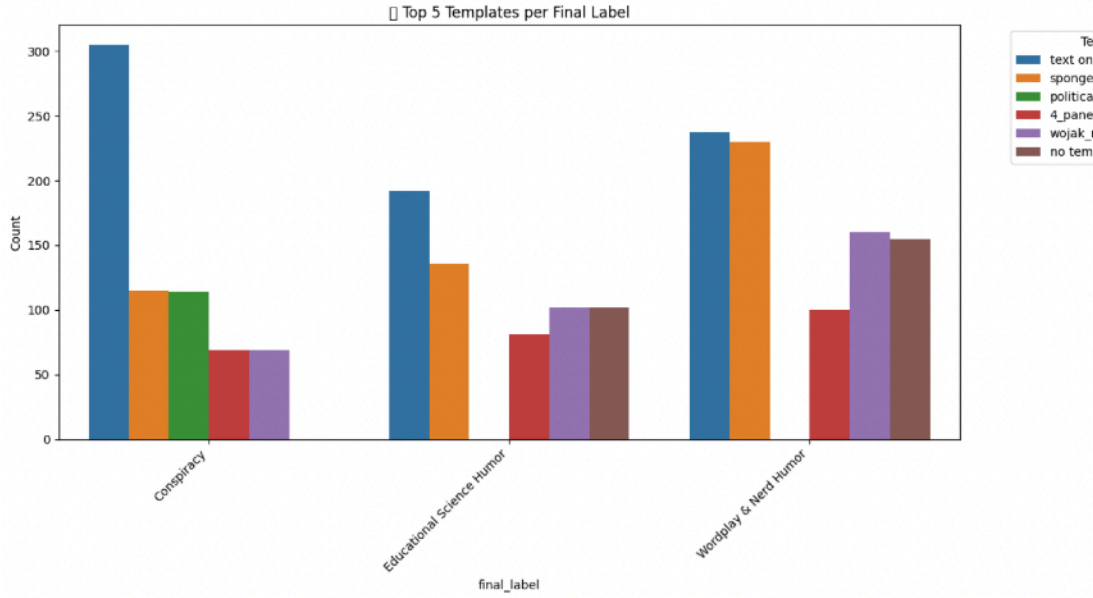
We started by simply analysing and plotting what were the most frequently used templates across the whole dataset.

The subsequent analyses is concerned with the relation between the template names and the labels coming from the first section of the project: “conspiracy”, “Wordplay & Nerd Humor” and “Educational science humor”.

We first investigated what was the template usage by label and visualised it with the help of a heat-map

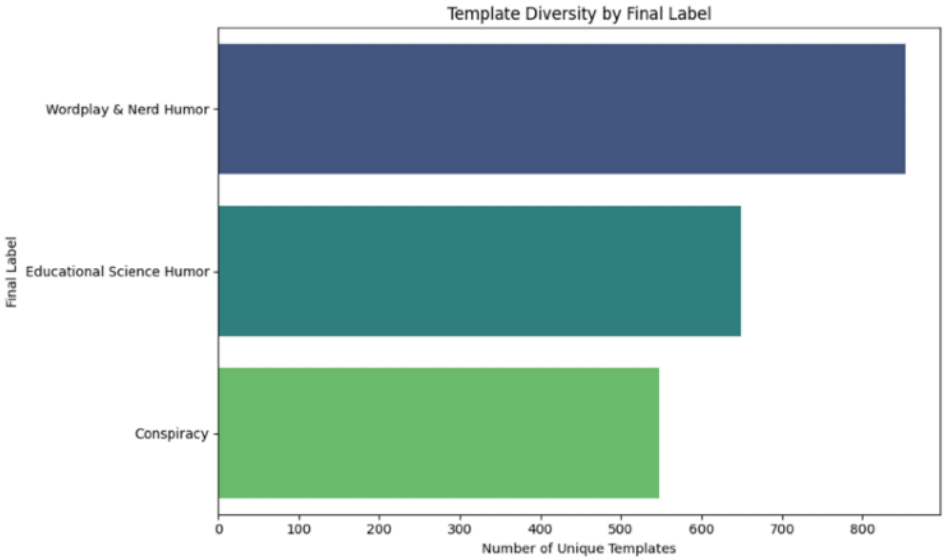


Then a **top templates per label** visualisation followed



The succeeding analysis was concerned with measuring how visually repetitive or diverse memes are in a category:

The results show a higher diversity in the non-conspiracy categories implying a higher amount of repetition for the conspiracy category - when constructing a conspiracy meme people choose from a more limited pool of templates.



A final and more statistically centred examination was centred on the Chi-Squared Test for Independence. The idea was to statistically tests whether certain templates are associated with certain labels more than expected by chance.

Chi2 = 4143.70, **p** = 0.0000

The p-value, showing statistical significance, demonstrates that the answer to that question is positive, there is a correlation between meme category and chosen template.
An example of this (and only one will be highlighted for report length reasons) is the political compass meme template (numbers indicate the number of times the template was used in the category): Conspiracy: 114; Wordplay & Nerd Humor: 26; Educational science humor: 31.
On the other hand there are also many templates that are random or uniform.