Proceedings of the Twentieth International AAAI Conference on Web and Social Media (ICWSM 2026)

SemioMeme: A Symbolic–Subsymbolic Knowledge Graph Dataset for Multimodal Meme Analysis

Victoria Sherratt, Suzanne Elayan, Nina Dethlefs

Loughborough University
{v.sherratt, s.elayan2, n.dethlefs}@lboro.ac.uk

Abstract

Internet memes present a challenge for computational analysis as their meaning derives from cultural context external to their observable features; however, visually similar memes carry distinct cultural meanings, whilst semantically related memes may share no perceptual similarity resulting in a decoupling of format and meaning. To support analysis requiring both, we present SemioMeme, a knowledge graph providing symbolic representations of meme concepts with their cultural connections alongside subsymbolic vision and text embeddings connected via a dedicated property between both. This supports hybrid queries that can surface cultural associations accruing through graph proximity, often invisible to similarity search or explicit labelling alone. The resource, including source data and code, is made openly available and covers 16,707 meme concepts, 507K meme instances with multimodal embeddings, and 7.2M RDF triples.

Datasets — https://doi.org/10.5281/zenodo.17826799

1 Introduction

Internet memes are forms of online communication widely used to express opinions, humour and political commentary on social media platforms (Sherratt 2022). Memes generate meaning through the interaction of their visual and text modalities, interpreted alongside cultural knowledge, context or lore external to the meme itself. This reliance on implicit context from their shared histories makes them significant for understanding online discourse, yet difficult to analyse computationally as information necessary for interpretation is not fully contained within its observable features (Iloh 2021; Kiela et al. 2020; Polli and Sindoni 2024; Sherratt 2022; Nguyen and Ng 2024).

Memes are extensively documented through archival websites and visible on social media platforms, providing sources for researchers to collate data to study memes. However, this information largely exists in fragmented formats; machine-readable resources such as OCR-extracted text, images or embeddings usually exist as task-specific datasets, capturing memes at a certain point in time to address specific objectives such as harmful meme detection, and are therefore too thematically narrow to represent the diversity of meme culture (Afridi et al. 2020; Hermida and Santos 2023; Sharma et al. 2022; Routhu and Baruah 2025). Few task-agnostic resources capture the broader cultural position of internet memes to model relationships between meme concepts or their connection to the people, events and online communities which are central to their interpretation.

Encyclopedic resources like KnowYourMeme.com provide rich documentation of meme origins and cultural context, but this information exists primarily as unstructured text, embedded metadata or loosely organised categorical tags; the cultural knowledge encoded in these descriptions remains implicit in prose rather than explicitly modelled as queryable relationships.

Knowledge graphs have previously been proposed as a solution to structuring this encoded cultural knowledge (Sherratt 2022). However, existing resources that model meme data in knowledge graph formats or leverage archival sources like KnowYourMeme.com remain either limited in scale and availability, or characterise memes as template-based image-text objects rather than objects that operate within a broader online cultural ecosystem (Ekron, Milo, and Youngmann 2017; Tommasini, Illievski, and Wijesiriwardene 2023).

We therefore introduce SemioMeme, a multi-component knowledge graph and dataset resource for analysing internet memes. SemioMeme is designed on the principle that meme meaning and appearance are decoupled; content-based similarity captures visual and textual patterns through learned relationships, whilst semantic similarity reflects conceptual relationships, including associations that arise through cultural proximity rather than visual resemblance or direct involvement. Neither representation alone suffices for comprehensive meme analysis.

SemioMeme instead provides both semantic and perceptual representations through a symbolic RDF knowledge graph modelling meme relationships through metadata or unstructured text, and subsymbolic dense vector embeddings of meme images and text indexed for similarity search, connected through an explicit bridge layer mapping symbolic URIs and vector positions. This separation enables analyses that require both representations, such as tracing how cultural associations accumulate around entities, identifying where meme format and meaning diverge, or constraining similarity searches by semantic criteria.

	Description	Format	Location
Source Data			
KYM meta data	Entry pages including descriptive text and meta data from KnowYourMeme.com (16,707)	CSV	Zenodo
Raw images	Confirmed and Submission meme images downloaded (757,331)	Images	On Request
Image URLs	URLs for Confirmed meme images used in embedding construction (507,127)	CSV	Zenodo
OCR text	Text extracted using Google Vision from meme images (419,482)	CSV	Zenodo
Knowledge Graph			
Meta Layer	Semantic relationships modelled from KYM meta data	TTL	Zenodo
Corpus Layer	Meta Layer with instances from collected memes and bridge map to FAISS indices	TTL	Zenodo
Bridge Map	Standalone file linking instances and meta data by URI FAISS indices	JSON	Zenodo
Instance Graph	Graph of only meme instances and bridge map to FAISS indices	TTL	Zenodo
Ontology	Formal schema from SemioMeme Meta and Corpus layers	OWL/TTL	Zenodo
Embeddings and Indices			
Vision Embedding	SigLIP features of meme instances	NPY	Zenodo
Text Embeddings	SentenceTransformer features of meme instances	NPY	Zenodo
Vision Indices	Searchable FAISS indices built from fine-tuned SigLIP model	FAISS	Zenodo
Text Indices	Searchable FAISS indices built from fine-tuned SentenceTransformer model	FAISS	Zenodo
Retrieval Models			
SigLIP	Fine-tuned SigLIP model for retrieval of similar memes	PyTorch	Zenodo
SentenceTransformer	Fine-tuned SentenceTransformer model for retrieval of similar memes	PyTorch	Zenodo
Source Code			
KYM Webscraper	Complete pipeline for collecting KYM entries and meme instances (knowledge graph source data)	Python	GitHub
SemioMeme - Meta	Complete pipeline for building the knowledge graph	Python	GitHub
SemioMeme - Retrieval	Complete training script for fine-tuning models and generating embeddings	Python	GitHub
SemioMeme - Corpus	Complete pipeline for generating bridge mappings and corpus layer from embeddings	Python	GitHub

Table 1: Overview of available resources.

Alongside the knowledge graph, we release the complete source data, extraction pipelines, embeddings and fine-tuned retrieval models to reproduce or extend this resource.

In Section 2 we outline similar research with specific focus on available resources for meme analysis. We then detail the collection of the source data in Section 3. In Section 4 we detail the construction of the RDF Meta Layer, and in Section 5 building the RDF node of meme instances for the Corpus Layer and bridge map to FAISS indices. In Section 6 we demonstrate the unique capabilities of the three layers and resource, and discuss further uses and future work in Sections 7 and 8.

1.1 Complete Resource

We present three modular resources comprising the core of SemioMeme alongside source data, images, OCR-extracted text, source code and documentation to reconstruct the graph and embeddings, outlined in Table 1. SemioMeme is available under a CC BY 4.0 license with a persistent identifier and metadata for FAIR compliance, and employs standard Semantic Web Technologies with established vocabularies (Wilkinson et al. 2016). SemioMeme's three core components are:

Meta Layer (Symbolic): knowledge graph of 664,043 RDF triples (subject-predicate-object statements) covering 16,707 meme concepts based on KnowYourMeme.com (KYM.com) entries. The Meta Layer models conceptual meme types or categories and people, subcultures, events and sites associated with internet culture, connecting entities through available metadata on KYM, extracted relationships, or modelling series connections.

Corpus Layer (Bridge Map): an extended graph containing individual meme instances categorised under each KYM entry in the Meta Layer, extending the RDF graph to 7,460,955 triples. The Corpus Layer includes OCR-extracted text from meme instances, entities extracted from text and a dedicated property (the bridge map) linking instances to their corresponding vision or text indices.

Embedding Layer (Subsymbolic) FAISS-indexed dense vector representations of meme instances (vision and text embeddings), enabling subsymbolic similarity search linked to symbolic corpus nodes via bridge map in the Corpus Layer.

In Appendix A.1, we also show what information is available for a single meme instance from this resource.

2 Related Work and Resources

Existing work on structured meme representations either uses external knowledge to improve specific tasks like classification and harmful content detection, or constructs dedicated knowledge graph resources. Graph-based methods have used scene graphs, graph neural networks, and external knowledge bases such as ConceptNet to improve meme classification and harmful content detection (Kougia et al. 2023; Shang et al. 2021; Lee et al. 2021; Garg et al. 2025; Sharma et al. 2023; Grasso et al. 2024). Whilst these contributions demonstrate the value of using external knowledge like knowledge graphs to improve meme understanding, any supporting resources released are task-specific.

Task-agnostic representations in knowledge graph formats include SimMeme, an ontology-driven search engine with manual annotations, and the Internet Meme Knowledge Graph (IMKG), which models template-based memes with external knowledge integration (Ekron, Milo, and Youngmann 2017; Tommasini, Illievski, and Wijesiriwardene 2023). These systems address different research objectives: SimMeme pioneered ontology-based retrieval at limited scale; IMKG enables large-scale analysis of template-based memes through extracted entities. SemioMeme differs in three key respects (see Table 2 for detailed comparison):

Metric	SemioMeme (2025)	IMKG (2023)	SimMeme (2019)
Graph Scale			
Total nodes	771,085	4,850,636	Not reported
KYM nodes	771,085	167,662	-
Wikidata nodes	3,295 (links)	85,917 (subgraph)	-
ImgFlip nodes	-	4,698,912	-
Total triples	7,460,955	Not reported	Not reported
Total edges	2,219,396	16,549,810	Not reported
KYM edges	1,488,530	914,941	-
Wikidata edges	3,295 (sameAs)	504,781 (subgraph)	-
ImgFlip edges	-	15,129,606	-
Content Coverage			
Meme categories/types	16,707	12,585	Not reported
Individual memes	507,127	1,338,617	10,000
From KYM	507,127	12,585	-
From ImgFlip	-	1,326,032	-
From Wikidata	-	242	-
Entity types modelled	19 types	3 core types	2 core types
Multimodal Data			
Vision and Text instances	318,668 (62.7%)	Majority	Majority
Vision-only instances	188,461 (37.1%)	Not reported	Not reported
Text-only instances	1,302 (0.3%)	Not reported	Not reported
OCR text coverage	419,482 (82.7%)	Not reported	Not reported
Vector index	FAISS (vision + text)	-	-
Multimodal retrieval	Vector similarity	Entity-based queries	Tag + structure
Knowledge Extraction			
Entities extracted	459,382	3,780,975	Manual annotation
From captions	422,071	3,344,941	-
From images	-	388,579	-
From descriptions	37,311	47,455	-
Extraction method	REBEL	DBpedia + Google Vision	Manual
Triples extracted	7,460,955	Not reported	Not reported
From captions	6,816,912	Not reported	-
From images	-	388,579	-
From descriptions	664,043	Not reported	-
Architecture & Access			
Storage format	RDF + FAISS indices	RDF + Property Graph	Graph database
Layer separation	3 explicit layers	Integrated	Integrated
Query interface	SPARQL + FAISS	SPARQL or Cypher	Custom search
Public availability	Full (data + code)	Full (data + code)	Demo only

Notation: - = None/not applicable; Not reported = Information not available in published literature

Table 2: Comparison of Internet Meme Knowledge Graphs

Architectural Approach: Whilst IMKG's integrated architecture supports cross-modal queries and SimMeme's flat ontology enables straightforward retrieval, SemioMeme implements a three-layer architecture that explicitly separates symbolic and subsymbolic representations. Separating conceptual knowledge from instance-level content whilst maintaining an explicit map supports cleaner analytical queries around high-level concepts, more accurate similarity search of instances, and a flexible hybrid combination of both without collapsing representational layers.

Data Focus: SemioMeme prioritises comprehensive modelling of KnowYourMeme's editorially curated content, capturing structured metadata which includes origin narratives, temporal properties, and cross-cultural references. External enrichment is targeted rather than importing external subgraphs, whilst remaining extensible for researchers requiring broader integration. Curated encyclopedic data offers verified provenance that user-generated platforms typically lack; whilst IMKG achieves greater total instance volume through ImgFlip integration, SemioMeme provides substantially deeper coverage of KnowYourMeme content and leverages its rich metadata to connect meme concepts.

Cultural Ecosystem: SemioMeme models memes, people, events, subcultures and sites. Combined with vector indices, this enables hybrid queries chaining semantic filtering with perceptual similarity, such as finding memes visually similar to a template that reference a specific political event, or identifying subcultures sharing visual styles. Crucially, this structure often reveals cultural associations through proximity rather than direct involvement, surfacing connections that neither approach captures alone. We explore this in Section 6.

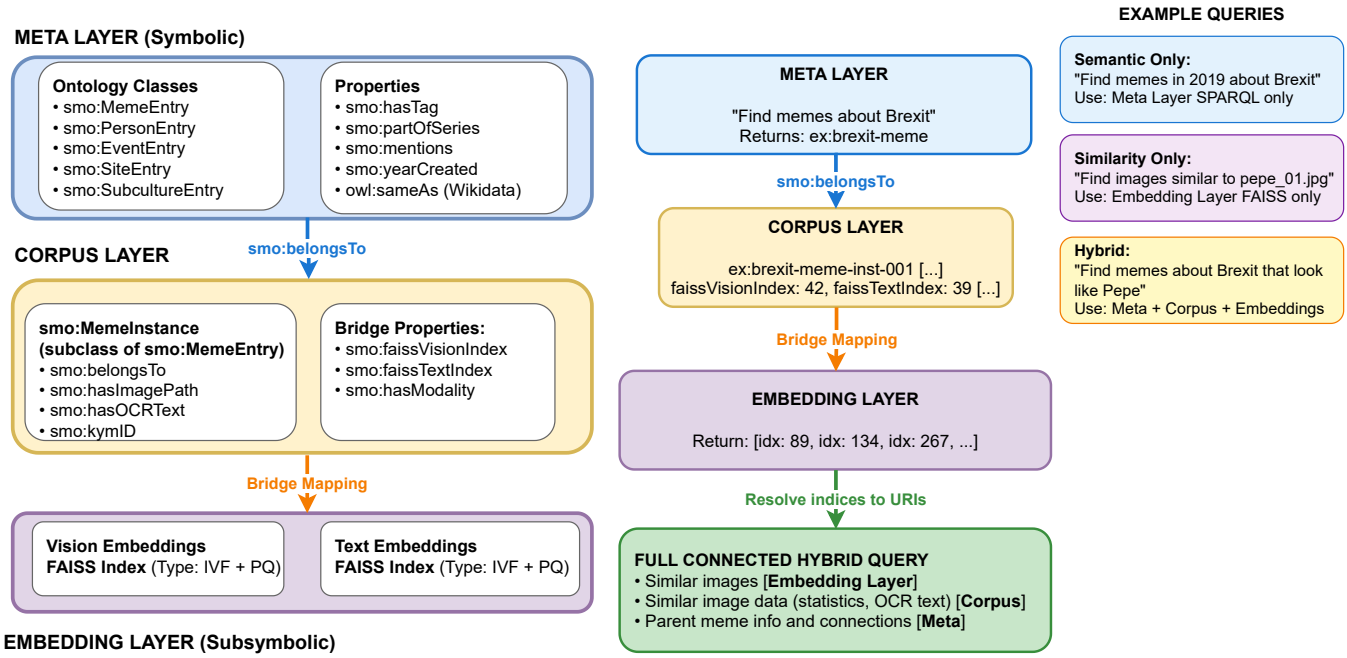

Figure 1: SemioMeme Three-Layer design approach and example query paths.

3 Data Collection

The primary data source for the knowledge graph is the publicly accessible KnowYourMeme.com (KYM), a collaborative, wiki-style website documenting internet phenomena, viral content, digital culture and internet memes. 'Entry pages' document phenomena or groups of memes - that is, memes which have similar histories, context, visual styles or catch phrases. These entry pages constitute the primary source data of the Meta Layer. Each 'entry' also contains example memes of the entry in image galleries; these meme instances constitute the Corpus Layer of the graph.

KYM records editorial status as Confirmed (editorially approved by moderators) or Submission (unreviewed). Submission entries often lack sufficient documentation but remain connected to other entries through metadata. For the purposes of our resource, all entries (Confirmed and Submission) are included in the Meta Layer; the Corpus Layer contains only Confirmed memes. Confirmed entries exhibit higher reliability in their metadata and relationships, while Submissions provide broader coverage at the cost of potential noise particularly at instance-level. We outline our decision on the impact on data quality of Submission memes further in Section 5; however, we provide indices and embeddings for Submission memes are part of the resource.

The collection pipeline consists of modular Python scripts executed sequentially:

Index Collection: Retrieves paginated listings of entries and splits these into categories (memes, people, subcultures, events and sites) and collects their status editorial status (Confirmed or Submission).

Update Detection: Compares the current webpage listing to the previously saved index to flag new pages that should be collected.

Entry Collection: Performs extraction of individual pages based on the index and metadata, outlined in Appendix A.2 Figure 7: main image URL, titles, series link, engagement metrics, entry type, metadata (origin, year, region), community tags, descriptive text and image gallery links containing meme instances. A rate limit of 31 seconds is applied to avoid server burden and entries are saved in batches to save progress.

Image Download: Collects the image URLs on each gallery page (Number 9 on Figure 7 in Appendix A.2) of an entry, then downloads the media content.

We use BeautifulSoup for HTML parsing and save a local copy of each page to enable offline re-collection. Failed entries are retried from cached HTML files or skipped if unavailable. With the applied rate limit, full collection takes approximately 5 days. Downloaded images include .jpg, .png, and .gif formats but exclude video formats. Unique identifiers are generated when entries are collected, and images are saved using the same unique identifier to ensure they can be tracked, merged and analysed later.

For the purposes of this knowledge graph, we collected KYM data in February 2023 initially, which included all content from December 2008 (KYM's first entry) up until this date. To confirm reproducibility and robustness of the method, in July 2025 we re-collected newly Confirmed memes and meme images to add these to the knowledge graph. This recollection also enabled us to identify that 3,917 memes entries which were initially flagged as Submission and were promoted to Confirmed; thus, the resource

provided in this paper is based on memes collected and their editorial status as of July 2025 from KYM.

In total, this collection consists of 507,127 Confirmed meme images and 250,179 Submission memes downloaded (757,331 total), 13,161 meme entries, 345 site entries, 1,625 events entries, 973 person entries and 603 cultures/subculture entries and their associated content.

3.1 Text Extraction

We extract the text of individual memes collected using Google Vision's API service[1]. Although there is a cost associated with using this service, we provide all extracted data as part of the available resource for other researchers to utilise. In Appendix A.3, we examine extraction consistency through cross-OCR comparison. Extracted OCR text is used in constructing the indices and extracting entities; at this stage we preprocess the text data to omit text which is of insufficient length (less than two words) or contains non-UTF-8 characters. The design of the graph, separating curated knowledge and user-generated data, isolates OCR-derived content within the Corpus Layer. As detailed in Section 5, OCR-extracted text and entities can be filtered by excluding this property, enabling users to apply additional post-processing where required.

4 Meta Layer Construction

The Meta Layer is a formal RDF knowledge graph representation of each entry page extracted from KYM. Extracted data from KYM is used to model the intertextuality of internet memes by connecting entries through their shared tags, series attributes, temporal data, origins and via entities extracted from unstructured text. The Meta Layer can be used as a stand-alone graph or combined with instance level data and indices for hybrid querying.

4.1 Data Processing

We begin by building a URL-to-title mapping for all collated KYM entries. Entries reference each other inconsistently, for example series links use URLs, tags use entry names, and origin references vary in format. The mapping normalises these references, ensuring that connections like `smo:PartofSeries` resolve correctly regardless of how the source data encodes them. A similar resolution process is applied to extracted entities (Section 4.3), ensuring multiple mentions of the same concept link to a single node.

To create valid RDF URIs from meme titles, we generate a 'URI Safe' version of each title column. This transformation (1) converts to lowercase for consistency, (2) replaces spaces and hyphens with underscores, (3) removes characters which are not alphanumeric, (4) collapses consecutive underscores, and (5) trims trailing underscores. Multi-valued properties (tags, meme types) are parsed from comma-separated lists into individual RDF relationships. Duplicate entries are removed based on unique identifiers.

[1]https://cloud.google.com/vision

4.2 Ontology Development

SemioMeme's ontology (SMO) defines two root classes: `smo:KYMEntry` for documented internet phenomena and `smo:KYMMetadata` for meta data and classification structures. `smo:KYMEntry` extends both `schema:CreativeWork` and `foaf:Document` connecting Schema.org's web publishing ecosystem to the FOAF vocabulary for social web applications. The ontology also defines `smo:MemeInstance` as a subclass of `smo:MemeEntry`, used in the Corpus Layer (Section 5) to represent individual meme instances. Under `smo:KYMEntry`, we define 6 primary entity types reflecting KYM content categories:

`smo:MemeEntry`: documentation about memes

`smo:PersonEntry`: individuals relevant to internet culture

`smo:EventEntry`: events that influenced meme creation

`smo:SiteEntry`: websites or platforms where memes originate or spread

`smo:SubcultureEntry`: online communities that produce or consume specific meme types

`smo:Entity`: fallback class for REBEL-extracted entities not matching above categories (171,934 instances).

Under `smo:KYMMetadata`, five subclasses capture KYM's classification structure: `smo:Tag` (folksonomy tags), `smo:Series` (thematic groupings), `smo:MemeType` (format classification, e.g., 'image macro'), `smo:Status` (editorial status) and `smo:Badge` (identifying 'Not Safe for Work' content). The ontology defines 34 properties organised into 8 object properties (relationships between entities) and 26 datatype properties (attaching literal values such as temporal data, content and engagement metrics, meta descriptions, or properties for provenance tracking of extracted data).

4.3 Entity Extraction and Linking

Beyond incorporating structured metadata from KYM, we extract additional entities and relationships from the text data on entry pages using REBEL, a transformer-based extraction model (Huguet Cabot and Navigli 2021). For all descriptive text on a KYM entry page, REBEL identifies entity mentions and semantic relationships to produce triples such as ⟨Elon Musk, creator, Tesla⟩. Extracted relations undergo canonicalisation to reduce semantic redundancy; relations appearing fewer than 5 times are mapped to more frequent forms using string similarity where possible (e.g., "founded by" → "founder"). Extracted entities are also linked to their parent KYM entry via `smo:mentions`. All REBEL-extracted predicates are marked with `smo:extractedBy ''REBEL''` provenance triples, enabling filtered queries to exclude extracted content.

Extracted entities are also resolved against existing KYM entries based on the defined hierarchy. For example, where an extracted named entity matches an existing KYM entry

(e.g., 'Elon Musk' matches a PersonEntry page), the entity is linked to the `smo:PersonEntry` rather than creating a duplicate node. Entities without existing matches are typed as `smo:Entity` to distinguish curated documentation from extracted entities.

Finally, we integrate the broader Linked Open Data ecosystem through Wikidata linking; for KYMEntry entities, the system queries Wikidata using label matching, alias resolution, and name variation handling. Successful matches establish `owl:sameAs` links (3,925 entries enriched), and type-specific enrichment adds supplementary properties: PersonEntry entities receive `foaf:name`, EventEntry entities receive `schema:eventName`, and so on. We attach only the Wikidata QID rather than importing connected subgraphs to enrich only curated KYMEntry entities to avoid polluting the graph with extensive external linkage and maintain focus on KYM's curated collection. The source code optionally enables enrichment of REBEL-extracted entities where preferred. Manual verification of 200 randomly sampled `owl:sameAs` links showed 98% unambiguously correct, with 2% linking to valid but broader Wikidata entities. In Appendix A.4 we detail the sequence-level confidence scores of REBEL triple extraction.

5 Corpus Layer

The Corpus Layer represents individual meme instances collected from KYM image galleries, linking each instance to its parent entry in the Meta Layer and to positions in FAISS indices. Each meme instance becomes an RDF node with properties referencing both the symbolic Meta Layer and subsymbolic vector indices to enable hybrid querying, yet maintain clear distinction between conceptual knowledge and perceptual similarity. Construction occurs in two stages: FAISS indices are first built from image and text embeddings of individual memes, and RDF nodes are constructed from these embeddings to reference index positions. We describe each stage below.

5.1 Embedding Index Construction

We extract embeddings from meme images using SigLIP's vision encoder to produce 768-dimensional vectors for each instance (Zhai et al. 2023). Metadata is preserved throughout which tracks filenames, KYM class identifiers (the KYM ID linking to parent entries) and editorial status (Confirmed or Submission).

Optionally, retrieval accuracy can be improved by training a linear projection head using supervised contrastive learning and meme classes as the labels. Fine-tuning SigLIP improves recall@1 from 60% to 74% and recall@5 from 74% to 85% over 507,127 memes across 8,597 classes in the Confirmed meme entries. We provide both raw embeddings and fine-tuned embeddings to enable researchers to optionally develop alternative fine-tuning approaches at this stage.

We construct FAISS Inverted File indices with product quantisation and generate separate indices for Confirmed memes, Submission memes, and combined sets. Each index maintains metadata mappings linking FAISS positions to filenames, KYM identifiers, and popularity scores.

The text pipeline follows an identical structure; OCR-extracted text described previously in Section 3.1 is processed through a SentenceTransformer with all-mpnet-base-v2 as the base model to produce 768-dimensional embeddings (Reimers and Gurevych 2019). We also fine-tune the projection head, improving retrieval recall@1 from 47% to 57%. Full fine-tuning experiments for both modalities are available in Appendix A.5.

5.2 Corpus Layer Construction

Once FAISS indices are generated the Corpus Layer is constructed as an RDF graph. For our design of the Corpus Layer, we opt only to use instances of Confirmed Memes; as previously described, Submission memes have varying quality despite being connected to Confirmed meme categories, making them suitable for Meta Layer inclusion but detrimental to retrieval. Retrieval accuracy when including Confirmed and Submission memes in a combined indices significantly reduced retrieval recall compared to using only the Confirmed indices. We outline the impact of using combined editorial status indices in Appendix A.5.

Each meme with a FAISS position becomes a `smo:MemeInstance` node, a subclass of `smo:MemeEntry`. Instance URIs are also generated from the filename hash and KYM ID. Using the metadata generated at the embedding stage, instances are matched with embeddings with `smo:faissVisionIndex` and `smo:faissTextIndex`. Each instance includes `smo:belongsTo` linking to the parent Meta Layer entry by title URI and `smo:kymID` linking by the generated ID, `smo:hasModality` indicating what modalities the instance contains (vision, text or both), `smo:datasetType` indicating editorial status and `smo:popularity` ranking based on the number of memes within a category. As outlined previously in Section 1.1, we provide two graph outputs: a combined graph merging the Meta Layer entries with the Corpus instances, and an instance-only graph for researchers only requiring meme instances.

5.3 Instance Entity Enrichment

Beyond structural properties, we enrich meme instances with context extracted from text elements. For instances with OCR data, we add `smo:hasOCRText` properties containing the extracted text as string literals, enabling direct text queries without requiring index lookups. We also extract entities from OCR text using the same REBEL pipeline described in Section 4.3, including entity resolution, enabling entity queries at instance level without embeddings. All OCR-derived entities are marked with the same `smo:extractedBy 'REBEL'` provenance triples as with the Meta Layer to allow exclusion of extracted data through simple SPARQL filters based on provenance.

6 System Capabilities

We demonstrate core system capabilities through three query modes that highlight cultural research applications.

Figure 2: Elon Musk 'Person Entry' with top 20 connections by engagement.

6.1 Semantic Queries

We query the Meta Layer for the `smo:PersonEntry` 'Elon Musk', which has 76 direct connections via tags, series membership, and entity mentions extracted via REBEL. Figure 2 shows the top 20 connections filtered by engagement metrics and one depth of secondary connections.

Even this filtered view shows secondary connections to Site and Subcultures that indicate Musk acts as a bridge across platform-specific meme communities and disparate subcultures that rarely overlap. The connection types themselves indicate memes directly spawned from the target entity ('series' edges), while mentions and tags capture broader cultural associations connected to Elon Musk.

The Goncharov connection, a fictional Scorsese film invented on Tumblr, exemplifies the latter; Goncharov has no actual involvement with Musk, yet the graph highlights how individuals in internet culture become linked by accumulating associations through shared community participation, temporal co-occurrence, or thematic proximity even without explicit relationships. In this case, the connection is extracted from the free-text descriptions as a 'mentions' entity from REBEL. Such connections would not surface through visual similarity search as there is no visual resemblance between Elon Musk and Goncharov memes.

6.2 Similarity Queries

Using FAISS indices, similarity searches can be performed on meme text, images, or both modalities combined. Since FAISS returns index positions of meme instances stored in the Corpus Layer, optionally for each similarity search the parent entry information can be retrieved.

Figure 3: Similarity search query example.

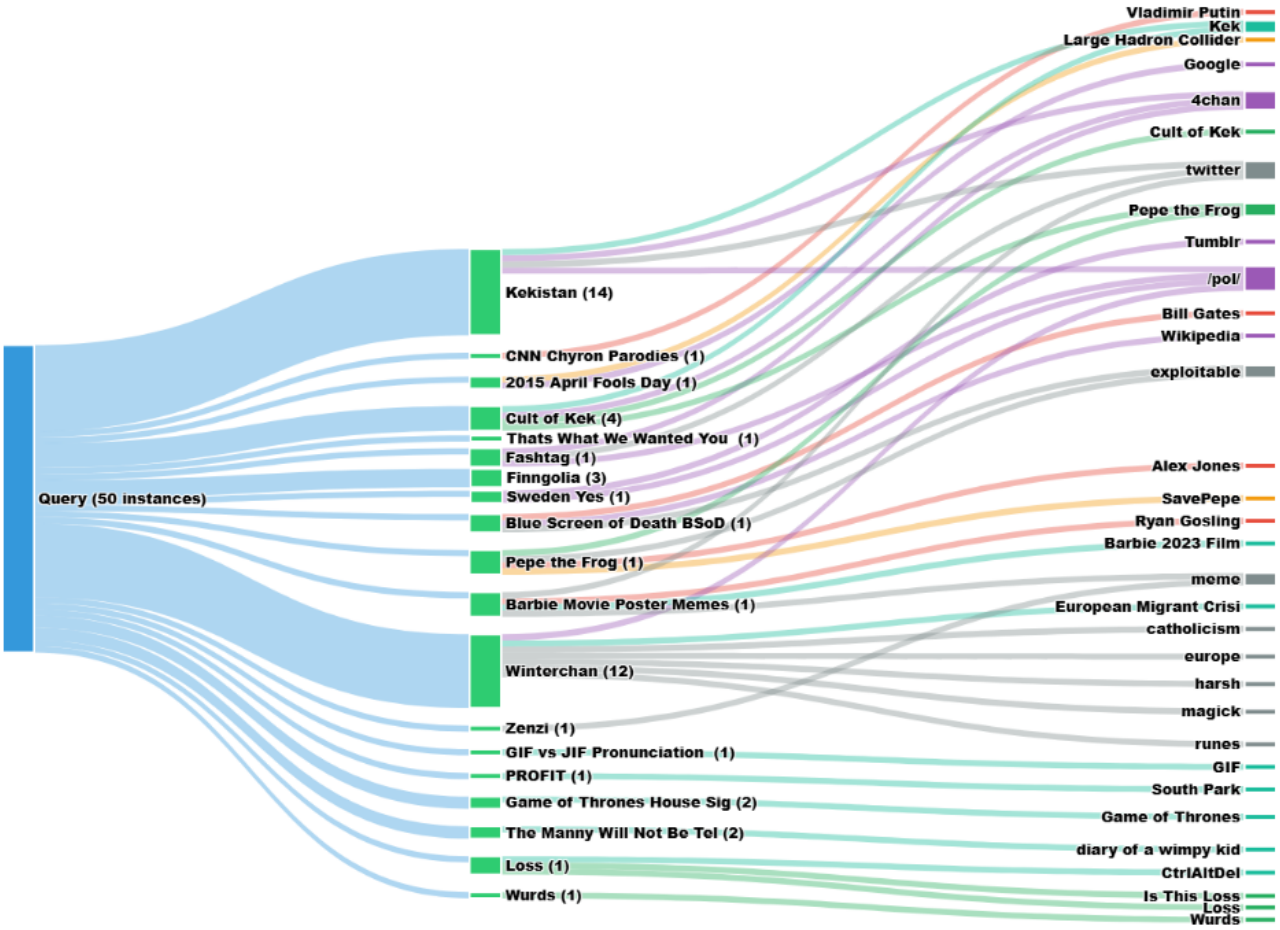

Figure 4: Semantic context from Meta Layer from visual similarity to Kekistan flag.

Figure 3 shows results for an unseen Pepe variant as the query image and the text 'feels good man'. Row 1 shows vision-only results, row 2 shows text-only results and row 3 shows multimodal fusion. Multimodal retrieval uses multiplicative Reciprocal Rank Fusion, scoring candidates as $s = \frac{1}{(k+r_v)(k+r_t)}$ where r_v and r_t are ranks in each modality, and candidates must rank well in both modalities to score highly.

The results demonstrate why separating perceptual and semantic representations is important; visual similarity returns results spanning two distinct entries ('Feels Bad Man Sad Frog'and 'Pepe the Frog') that share visual features but different cultural meanings. Text similarity returns visually disparate images within a single entry ('Feels Good Man') which share the catchphrase but nothing visual. Conversely, multimodal fusion crosses three entries to retrieve memes that share both visual style and textual content.

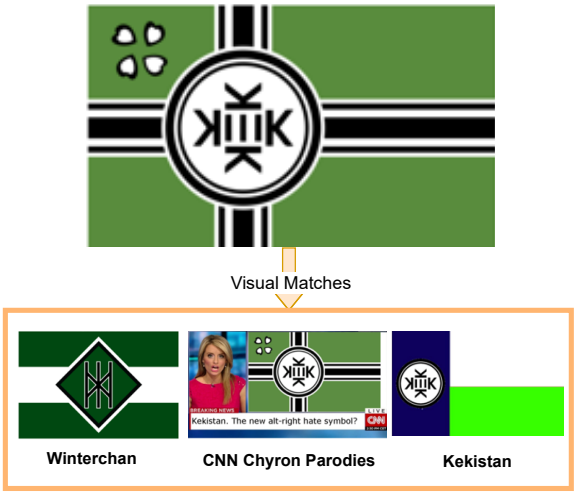

Figure 5: Kekistan Flag Visual Matches.

6.3 Hybrid Querying

Hybrid queries combine semantic filtering from the Meta Layer with similarity search in the Corpus Layer to either retrieve information based on restrictive criteria, or add conceptual and contextual information to similarity searches. We demonstrate the latter by searching for an unseen image of the 'Kekistan' flag, a satirical symbol associated with alt-right communities and deeply connected to 'Pepe the Frog'

during its height as a politically charged meme icon (Al-Rawi 2020; de Keulenaar 2023; Tuters 2019).

Figure 4 visualises the hybrid query semantic flow retrieved, with Figure 5 showing example matching queries from a similarity search. Visual similarity, the left column of Figure 4, retrieves 50 similar memes spanning entries grouped in the middle column (e.g., Kekistan: 14 visual matches, Winterchan: 12 visual matches).The right column shows semantic connections reachable from each entry, where 4chan, /pol/ and Pepe the Frog recur across multiple paths while outliers like Barbie Movie Posters reflect visual coincidence.

The most revealing result is the proximity of the meme entry Winterchan. Visual similarity places it with Kekistan as a query, but also amongst visual coincidences like Barbie Movie Posters, with no basis to distinguish culturally significant matches from this coincide. Examining Winterchan in isolation, its tags contain no overtly hateful or extremist terminology, and OCR text yields only sparse results which may warrant further investigation, but none of these signals alone would identify Winterchan as problematic.

However, graph traversal changes this by surfacing connections that contextualise visual matches. Winterchan shares a /pol/ connection with Kekistan, the meme entry for the queried image, with secondary hops linking both to 4chan and Russia. Winterchan carries a unique tag referencing the European Migrant Crisis, and OCR text includes recurring 'White Christmas' terminology. In combination, these signals warrant closer examination. SemioMeme does not classify Winterchan as problematic; rather, it surfaces the context necessary to interpret why a visual match may be culturally significant. Conversely, coincidental matches can be confirmed by sharing no cultural connections (only perceptual similarity) to known harmful content. Appendix A.6 reinforces proximity-based association discovery and demonstrates retrieved categories are graph-connected at 2.8× random baseline.

7 Discussion

The query modes demonstrated in Section 6 show three distinct analytical capabilities, where semantic queries reveal how entities accumulate cultural associations through proximity rather than direct involvement (which keyword matching cannot surface), or similarity search exposes the decoupling of format and meaning, where perceptually similar instances span distinct cultural categories. Hybrid querying combines both without collapsing semantic and perceptual representations, enabling practical applications such as content moderation where problematic content rarely self-identifies through explicit tagging.

This flexibility comes from the separation of symbolic and subsymbolic representations, which requires synchronisation between layers and adds complexity compared to flat approaches. However, this design enables each component to function independently, for example the Meta Layer as a stand-alone cultural knowledge graph, the Corpus Layer for instance-level similarity search, or both combined for hybrid queries that neither supports alone.

The Corpus Layer includes only Confirmed memes; retrieval accuracy degraded substantially when Submission memes were included, reflecting inconsistent metadata and editorial quality, however we provide Submission embeddings separately for researchers requiring broader coverage. Similarly, all REBEL-extracted entities carry provenance triples, enabling queries to exclude or weight extracted versus curated relationships. Potential OCR errors remain isolated to the Corpus Layer and can be filtered for downstream tasks requiring precision. As KYM is English-language, coverage of non-English memes is limited and SemioMeme inherits its source data's cultural and language bias at present.

Whilst processing for OCR text and REBEL occurs offline, it is currently limited for live coverage due to associated costs; SemioMeme is currently designed for scheduled updates outlined in the resource package rather than live updates.

Our example demonstrates SemioMeme provides interpretive context rather than automated detection, supporting the close reading that cultural analysis requires; however, SemioMeme equally supports automated approaches such as graph-based risk scoring or lexicon-filtered retrieval at scale. The resource is intended to complement ethnographic or platform-native analysis by supporting hypothesis generation and large-scale pattern discovery.

Beyond content moderation, SemioMeme supports longitudinal cultural analysis through KYM's 17-year temporal coverage, platform migration studies via origin metadata, and genealogical research into meme evolution through series connections. The embedding data and KYM class labels also provide training data for computational approaches and an initial retrieval benchmark.

8 Conclusion and Future Work

We presented SemioMeme, a multi-layer knowledge graph and dataset resource for internet meme analysis. The resource comprises (1) a Meta Layer RDF knowledge graph encoding cultural relationships between meme concepts, people, events, sites, and subcultures; (2) a Corpus Layer linking over 500,000 meme instances to semantic context; and (3) FAISS indices enabling visual and textual similarity search with explicit bridges to symbolic representations.

This architecture supports analyses such as tracing cultural associations that emerge through proximity rather than direct connection, identifying where meme format and meaning diverge, and combining semantic constraints with perceptual similarity search to make large-scale retrieval tractable for specific research questions. The complete resource, including source data, construction pipelines and fine-tuned retrieval models, is publicly available to support reproducibility and extension.

Future work includes integration with complementary resources like IMKG to expand corpus coverage, or incorporating additional sources beyond KYM such as harmful meme datasets matched via similarity retrieval. Additionally, SemioMeme could be extended to include visual entity detection alongside OCR entity extraction, to identify

occurring characters, symbols and objects within meme instances. Visual entity detection would link instances directly to Meta Layer context, enabling symbol propagation analysis. We omit this currently as accurate detection at scale remains cost-prohibitive but plan this as an extension as open methods mature.

References

Afridi, T. H.; Alam, A.; Khan, M. N.; Khan, J.; and Lee, Y.-K. 2020. A Multimodal Memes Classification: A Survey and Open Research Issues. ArXiv:2009.08395 [cs].

Al-Rawi, A. 2020. Kekistanis and the meme war on social media. *The Journal of Intelligence, Conflict, and Warfare*, 3(1): 12–32.

de Keulenaar, E. 2023. The affordances of extreme speech. *Big Data & Society*, 10(2): 20539517231206810.

Ekron, M.; Milo, T.; and Youngmann, B. 2017. SimMeme: Semantic-Based Meme Search. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2455–2458. Singapore Singapore: ACM. ISBN 978-1-4503-4918-5.

Garg, R.; Padhi, T.; Jain, H.; Kursuncu, U.; and Kumaraguru, P. 2025. Just KIDDIN' : Knowledge Infusion and Distillation for Detection of INdecent Memes. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Findings of the Association for Computational Linguistics: ACL 2025*, 23067–23086. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-256-5.

Grasso, B.; La Gatta, V.; Moscato, V.; and Sperlì, G. 2024. KERMIT: Knowledge-EmpoweRed Model In harmful meme deTection. *Information Fusion*, 106: 102269.

Hermida, P. C. d. Q.; and Santos, E. M. d. 2023. Detecting hate speech in memes: a review. *Artificial Intelligence Review*, 56(11): 12833–12851.

Huguet Cabot, P.-L.; and Navigli, R. 2021. REBEL: Relation Extraction By End-to-end Language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2370–2381. Punta Cana, Dominican Republic: Association for Computational Linguistics.

Iloh, C. 2021. Do it for the culture: The case for memes in qualitative research. *International Journal of Qualitative Methods*, 20: 16094069211025896.

Kiela, D.; Firooz, H.; Mohan, A.; Goswami, V.; Singh, A.; Ringshia, P.; and Testuggine, D. 2020. The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes. In *Advances in Neural Information Processing Systems*, volume 33, 2611–2624. Curran Associates, Inc.

Kougia, V.; Fetzel, S.; Kirchmair, T.; Çano, E.; Baharlou, S. M.; Sharifzadeh, S.; and Roth, B. 2023. MemeGraphs: Linking Memes tonbsp;Knowledge Graphs. In *Document Analysis and Recognition - ICDAR 2023: 17th International Conference, San José, CA, USA, August 21–26, 2023, Proceedings, Part I*, 534–551. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-031-41675-0.

Lee, R. K.-W.; Cao, R.; Fan, Z.; Jiang, J.; and Chong, W.-H. 2021. Disentangling hate in online memes. In *Proceedings of the 29th ACM international conference on multimedia*, 5138–5147.

Nguyen, K. P.; and Ng, V. 2024. Computational meme understanding: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 21251–21267.

Polli, C.; and Sindoni, M. G. 2024. Multimodal computation or interpretation? Automatic vs. critical understanding of text-image relations in racist memes in English. *Discourse, Context & Media*, 57: 100755.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Routhu, R. K.; and Baruah, U. 2025. Sentiment Analysis on Memes: A Review. *Expert Systems*, 42(11): e70133.

Shang, L.; Youn, C.; Zha, Y.; Zhang, Y.; and Wang, D. 2021. KnowMeme: A Knowledge-enriched Graph Neural Network Solution to Offensive Meme Detection. In *2021 IEEE 17th International Conference on eScience (eScience)*, 186–195. Innsbruck, Austria: IEEE. ISBN 978-1-66540-361-0.

Sharma, S.; Alam, F.; Akhtar, S.; S., M.; Dimitrov, D.; San Martino, G.; Firooz, H.; Halevy, A.; Silvestri, F.; Nakov, P.; and Chakraborty, T. 2022. Detecting and Understanding Harmful Memes: A Survey'. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence Survey Track*, volume 23-29, –22. Vienna.

Sharma, S.; Arora, U.; Akhtar, M. S.; Chakraborty, T.; et al. 2023. MEMEX: Detecting Explanatory Evidence for Memes via Knowledge-Enriched Contextualization. *arXiv preprint arXiv:2305.15913*.

Sherratt, V. 2022. Towards Contextually Sensitive Analysis of Memes: Meme Genealogy and Knowledge Base. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 5871–5872. Vienna, Austria: International Joint Conferences on Artificial Intelligence Organization. ISBN 978-1-956792-00-3.

Tommasini, R.; Illievski, F.; and Wijesiriwardene, T. 2023. IMKG: The Internet Meme Knowledge Graph. In Pesquita, C.; Jimenez-Ruiz, E.; McCusker, J.; Faria, D.; Dragoni, M.; Dimou, A.; Troncy, R.; and Hertling, S., eds., *The Semantic Web*, volume 13870, 354–371. Cham: Springer Nature Switzerland. ISBN 978-3-031-33454-2 978-3-031-33455-9. Series Title: Lecture Notes in Computer Science.

Tuters, M. 2019. LARPing & liberal tears: Irony, belief and idiocy in the deep vernacular web.

Wilkinson, M. D.; Dumontier, M.; Aalbersberg, I. J.; Appleton, G.; Axton, M.; Baak, A.; Blomberg, N.; Boiten, J.-W.; da Silva Santos, L. B.; Bourne, P. E.; et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data*, 3(1): 1–9.

Zhai, X.; Mustafa, B.; Kolesnikov, A.; and Beyer, L. 2023. Sigmoid Loss for Language Image Pre-Training. arXiv:2303.15343.

Paper Checklist

1. For most authors...
   (a) Would answering this research question advance science without violating social contracts, such as violating privacy norms, perpetuating unfair profiling, exacerbating the socio-economic divide, or implying disrespect to societies or cultures? Yes, the resource enables academic study of internet culture using publicly available data.
   (b) Do your main claims in the abstract and introduction accurately reflect the paper's contributions and scope? Yes, we present SemioMeme as a dataset resource and clearly scope claims to what the architecture enables.
   (c) Do you clarify how the proposed methodological approach is appropriate for the claims made? Yes, Section 6 demonstrates capabilities directly supporting claims about hybrid querying and cultural ecosystem analysis.
   (d) Do you clarify what are possible artifacts in the data used, given population-specific distributions? Yes, we discuss KYM's editorial process (Confirmed vs Submission), English-language bias inherent in the source, and temporal coverage limitations.
   (e) Did you describe the limitations of your work? Yes, Section 7 discusses synchronisation complexity and scope limitations.
   (f) Did you discuss any potential negative societal impacts of your work? Yes, the Ethics Statement addresses offensive content and potential dual-use concerns.
   (g) Did you discuss any potential misuse of your work? Yes, we acknowledge content moderation research could theoretically inform evasion, though patterns are already public.
   (h) Did you describe steps taken to prevent or mitigate potential negative outcomes of the research, such as data and model documentation, data anonymization, responsible release, access control, and the reproducibility of findings? Yes, we provide NSFW filtering via badges, provenance triples enabling exclusion of extracted content, and comprehensive documentation.
   (i) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes.
2. Additionally, if your study involves hypotheses testing...
   (a) Did you clearly state the assumptions underlying all theoretical results? NA, this is a dataset paper without hypothesis testing.
   (b) Have you provided justifications for all theoretical results? NA
   (c) Did you discuss competing hypotheses or theories that might challenge or complement your theoretical results? NA
   (d) Have you considered alternative mechanisms or explanations that might account for the same outcomes observed in your study? NA
   (e) Did you address potential biases or limitations in your theoretical framework? NA
   (f) Have you related your theoretical results to the existing literature in social science? NA
   (g) Did you discuss the implications of your theoretical results for policy, practice, or further research in the social science domain? NA
3. Additionally, if you are including theoretical proofs...
   (a) Did you state the full set of assumptions of all theoretical results? NA
   (b) Did you include complete proofs of all theoretical results? NA
4. Additionally, if you ran machine learning experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? Yes, complete pipelines, trained models, and documentation are available at the provided GitHub and Zenodo links in the main paper body.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes, Section 5.1 describes fine-tuning procedure; full hyperparameters are available in the GitHub link provided in the main paper body. The Appendix further details training parameters.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? Yes, the Appendix further details training experiments on retrieval accuracy.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? Yes, described in the Appendix.
   (e) Do you justify how the proposed evaluation is sufficient and appropriate to the claims made? Yes, Section 6 demonstrates capabilities through representative queries; we do not claim state-of-the-art retrieval performance.
   (f) Do you discuss what is "the cost" of misclassification and fault (in)tolerance? NA, we release a resource rather than a deployed classification system.
5. Additionally, if you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? Yes, we cite REBEL, SigLip, SentenceBERT, and FAISS.
   (b) Did you mention the license of the assets? Yes, we mention the license of other tools and resources used to construct SemioMeme in the Section 7.1. KnowYourMeme as a public data source is briefly discussed separately in the ethics statement.
   (c) Did you include any new assets in the supplemental material or as a URL? Yes, complete dataset, models, and code are available from GitHub and Zenodo links in the main paper body.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? Yes, KYM content is user-submitted to a public wiki; PersonEntry nodes concern public figures documented in their capacity as internet culture participants.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? Yes, Ethics Statement addresses both; NSFW badges and provenance filtering enable responsible use.

(f) If you are curating or releasing new datasets, did you discuss how you intend to make your datasets FAIR? Yes, Section 1.1 introducing the resource details FAIR compliance including persistent identifiers, standard formats, and documentation.

(g) If you are curating or releasing new datasets, did you create a Datasheet for the Dataset? Yes, included in supplementary materials at Zenodo link in main paper body.

6. Additionally, if you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots? NA, no crowdsourcing or human subjects research.

(b) Did you describe any potential participant risks, with mentions of Institutional Review Board (IRB) approvals? NA

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? NA

(d) Did you discuss how data is stored, shared, and deidentified? NA

Ethics Statement

External resources used include the following licences: REBEL (CC BY-NC-SA 4.0), SigLIP and SentenceTransformers (Apache 2.0), FAISS (MIT).

SemioMeme derives from KnowYourMeme.com, a public archive commonly used in meme research. Our collection applies rate limiting and respects robots.txt directives. The dataset necessarily contains content that may be offensive, including memes associated with extremist movements, but include available filtering methods to exclude this content below.

PersonEntry nodes document public figures; no private individuals are identified. OCR-extracted text may inadvertently capture names; provenance triples enable exclusion where sensitivity requires. Hybrid querying could theoretically inform content moderation evasion, but these cultural associations are already implicit in public archives.

Image Distribution We release URLs rather than raw images due to (1) UK Online Safety Act 2023 liability considerations, (2) hosting constraints, and (3) deferring to KYM for authoritative hosting and takedown compliance. Raw images are available upon request from the corresponding author.

Offensive Content SemioMeme necessarily contains such material because excluding it would misrepresent meme culture and undermine the resource's utility for studying online discourse, including research on hate speech detection and content moderation. Filtering is enabled via `smo:Badge` (NSFW metadata), `smo:datasetType` (editorial status), and provenance triples (extracted vs curated content).

A Appendix

A.1 Representative Graph Entry

A single meme instance is represented across three interconnected components, which broadly includes the following in the available resource: the image itself (available on request from corresponding author), OCR-extracted text, image and text embeddings and all data modelled in the Corpus and Meta Layer RDF graphs. We detail the information per layer below and source data.

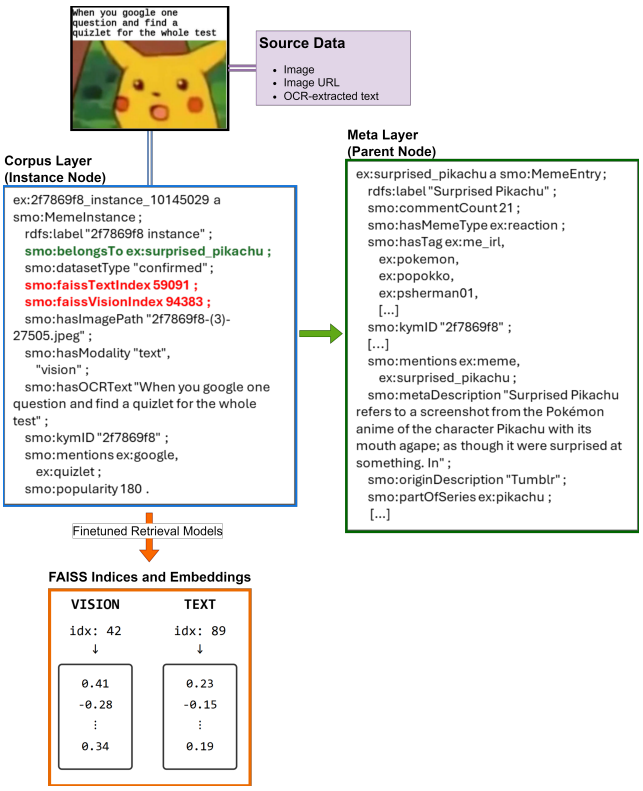

Figure 6: Provided resources for a single meme instance.

The **Corpus Layer** represents individual meme instances with OCR-extracted text, image file references, popularity scores, and bridge properties (`smo:faissTextIndex`, `smo:faissVisionIndex`) linking to FAISS index positions. Each instance connects to its parent concept via `smo:belongsTo`.

The **Meta Layer** represents documented meme concepts with editorial metadata, semantic classification (types, tags), textual descriptions (origin, spread), and relational proper-

ties (`smo:mentions`, `smo:partOfSeries`) linking to other entries.

The **Embedding Layer** stores SigLIP vision and SentenceTransformer text embeddings indexed via FAISS, accessible through the corpus layer bridge properties.

Finally, the **Source Data** includes the image upon request, OCR-extracted text, and image URL link to the filename stored in the graph for researchers to download. Figure 6 illustrates the complete resource structure for a single meme instance.

A.2 Example Entry Page and Collated Data

The primary source of the Meta Layer are the entry pages from KYM. Figure 7 illustrates an example entry from 'Surprised Pikachu'[2] and the collected data numbered. This includes a: main image URL (1), titles (2), series link (3), engagement metrics (4), entry type (5), metadata (6), community tags (7), descriptive text (8) and image gallery links containing meme instances (9). The Corpus Layer is then comprised of meme instances found under the image gallery link (number 9 on Figure 7). The source data for all entries is provided as part of the resource in a CSV file.

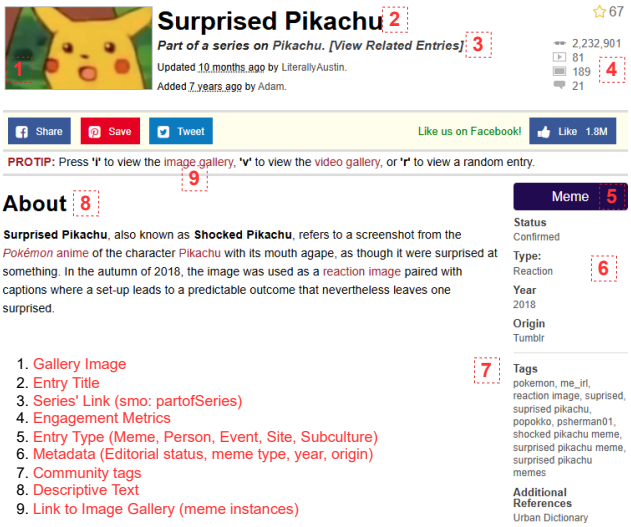

Figure 7: Example KYM entry page with collected data highlighted.

A.3 OCR Accuracy

To assess OCR extraction consistency, we compared Google Cloud Vision output against EasyOCR, an independent open-source OCR system, on a 10% random sample of meme images (n=50,826). Median character error rate between systems was 0.21, with median word-level Jaccard similarity of 0.64. This indicates moderate-to-good agreement between independent systems and consistent text extraction; discrepancies are expected given the typographic variability of meme images, including non-standard fonts often used in internet memes. We note this comparison validates consistency of the chosen extraction method (Google Vision) rather than accuracy against ground truth.

[2]https://knowyourmeme.com/memes/surprised-pikachu

A.4 REBEL Sequence-Level Confidence Scores

To assess REBEL extraction quality, we computed confidence scores from the model's token-level probabilities during generation. On a 10% sample of Meta Layer entries (n=1,695), median token-level confidence was 0.90 and median sequence-level confidence was 0.84, indicating reliable extraction from curated descriptions. For OCR text (n=60,608), confidence was lower (median token=0.82, median sequence=0.26), reflecting the fragmentary nature of meme text—short phrases, slang, and intentional misspellings rather than complete sentences.

A.5 Retrieval Fine-tuning

Visual embeddings are extracted from SigLIP (google/siglip-base-patch16-384) and textual embeddings from SentenceTransformer (all-mpnet-base-v2) on OCR-extracted text, both producing 768-dimensional representations.

Confirmed Memes	Metric	Raw	Fine-tuned	Improvement
Vision	r@1	60.14	73.67	13.53
	r@5	73.88	84.81	10.93
	r@10	77.61	87.19	9.58
	MMR	66.00	78.50	12.50
Text	r@1	46.55	56.97	10.42
	r@5	57.27	68.29	11.02
	r@10	60.65	71.32	10.67
	MMR	51.15	61.88	10.73
Submission Memes				
Vision	r@1	40.61	43.32	2.71
	r@5	71.79	77.47	5.68
	r@10	77.02	82.98	5.96
	MMR	53.89	57.90	4.01
Text	r@1	32.81	32.76	0.05
	r@5	57.49	57.02	-0.47
	r@10	62.07	61.68	-0.39
	MMR	43.39	43.17	-0.22
Combined Indices				
Vision	r@1	45.12	50.12	5.00
	r@5	69.48	74.72	5.24
	r@10	74.34	79.23	4.89
	MMR	55.47	60.58	5.11
Text	r@1	35.11	36.11	0.99
	r@5	54.04	54.84	0.80
	r@10	58.28	58.93	0.65
	MMR	43.19	44.17	0.98

Table 3: Retrieval result comparison by editorial status.

We employ metric learning on frozen embeddings with lightweight projection heads, trained on a single NVIDIA RTX 3090 Ti. On Confirmed memes, fine-tuning improves vision Recall@1 from 60.1% to 73.7% and text from 46.6% to 57.0% (Table 3).

Submission memes degrade retrieval performance even with raw embeddings; combining both subsets drops vision Recall@1 from 60.1% to 45.1%. This likely reflects lower samples per class, variable image quality, and label overlap with Confirmed entries. We therefore use Confirmed-only

Query	Entries Spanned	Entities (Top-1)	Entities (Top-5)	Density (Top-5)	Density (Top-20)	Top 5 Entries
Random Baseline	–	–	–	*17%*	*19%*	–
	1	23	23	–	–	Bad Luck Brian
	33	15	74	60%	9%	Winterchan; Kekistan; Loss; Fashtag; Check Your Privilege
	9	34	75	80%	89%	Pepe the Frog; Feels Bad Man; MonkaS; Twitch Emotes; Pepega
	17	10	52	50%	28%	SpongeBob Burning Paper; Stop Killing Alligators; Krusty Krab is Unfair; Gru's Plan; Buff SpongeBob
	14	16	49	30%	13%	Quiz Kid; The Crash Wasn't Your Fault; Hex Nut; Surreal Memes; Does Bruno Mars
feels good man	8	22	67	100%	86%	Feels Good Man; Feels Bad Man; Brushie Brushie; Approval Guy; Birthday Dog
this is fine	5	31	76	20%	20%	This Is Fine; Sonic Dreams Collection; It's Been 84 Years; Giannopoulos Portrait; Lonely Theresa May
make america great	10	72	135	50%	51%	Make America Great Again; Amerimutt; OK Symbol; DarkMAGA; Senator Armstrong
not great not terrible	32	10	62	10%	18%	Not Bad Obama Face; You're Gonna Have a Bad Time; Your Music's Bad; Purah; Art Gallery Puking
this does not spark joy	23	11	92	40%	31%	Does it spark joy; Trashcat Is Not Amused; Are You Not Entertained; Screaming Sun; Momo Challenge

Table 4: Hybrid query evaluation results. Entities counts unique semantic entities reachable at each cutoff. Density measures the proportion of category pairs sharing at least one entity.

fine-tuned models for all retrieval experiments and Corpus Layer construction.

A.6 Hybrid Query Evaluation

We evaluate how effectively graph structure distinguishes culturally meaningful similarity matches from visual coincidences. Perceptual similarity is a noisy signal for semantic relatedness; some visually similar memes share cultural connections while others match coincidentally, contributing to the format-meaning decoupling of internet memes. If the graph captures meaningful cultural structure, semantically related categories should share connections at rates above chance.

To test this, for each query (image or text), we retrieve the 50 nearest neighbours from the FAISS index and map each instance to its parent `MemeEntry` via the `smo:belongsTo` relation, aggregating by category and ranking by instance count. For each retrieved category, we extract all connected semantic entities from the knowledge graph: tags (`smo:hasTag`), meme type classifications (`smo:hasMemeType`), series membership (`smo:partOfSeries`), geographic regions (`smo:region`), and typed entity mentions (`smo:mentions`) from REBEL which includes people, events, sites, subcultures, and related memes.

We report the entries a retrieval query spans and total unique entities accessible at cutoffs of 1, 5, and 20 entries in Table 4. For coherence, we compute pairwise density as the proportion of retrieved category pairs sharing at least one semantic entity. We evaluate on 10 queries (5 vision, 5 text) and, to establish a baseline, we compute pairwise density for 100 meme entry samples selected at random. Random selection yields 17% density at top-5 and 19% at top-20.

Meme entries retrieved from a similarity search substantially outperform this baseline. At top-5, similarity-based retrieval achieves on average 49% pairwise density, indicating that retrieved categories share semantic connections nearly half the time compared to 17% by chance. This drops at top-20 which achieves on average 38% compared to only 19% at random. The highest coherence occurs for queries target-

ing memes with dense derivative networks and few entries spanned; for example ‘feels good man’ achieves 100% density while the Pepe image query reaches 80%. Conversely, generic text queries such as ‘not great not terrible’ achieves only 10% density, below the random baseline, indicating the embedding matched sentiment rather than meme concepts.

These results demonstrate that similarity search retrieves semantically related content at rates substantially above chance. That roughly half of matches remain coincidental validates the format-meaning decoupling central to this work, and demonstrates perceptual similarity alone cannot reliably identify cultural relationships. Low density results signal queries where similarity matched surface features rather than cultural meaning, whilst high density results indicate retrieval of coherent cultural clusters.