The News Atom: design decisions and possible rules

By Sannuta Raghu

The News Atom

In early February 2025, my team and I were working on how to create dynamically-generated timelines from our 11-year archive at Scroll. Our editors had requested the ability to pull contextual facts from deeply reported older stories, and a timeline seemed like a good place to start. This quest sent me down the rabbit hole of what the organising principles of facts might be and what signals we use as journalists to communicate them to our users.

In parallel, as a fellow at the Reuters Institute, I was studying provenance in journalism and how to negotiate a way out of the Faustian bargain.

These two streams of thought could be distilled into four guiding questions:

1. How can a story preserve its traceability and epistemic signals once it leaves its original container?
2. How can a story travel seamlessly across systems without semantic loss?
3. How can specific facts be pinpointed and recalled from within a vast mass of news content?
4. How can facts be extracted, recombined and adapted without losing meaning or rights clarity?

Answering these questions required more than an interface or a better CMS feature. They pointed to a structural gap: journalism has no fragment-level metadata standard that can preserve meaning, provenance and rights when content moves beyond its original form.

Existing frameworks operate primarily at the container level, and do not describe an individual fact within it. A solution would require designing not just a format but a metadata blueprint that could serve as the fundamental storage and transport unit of journalistic knowledge. This idea became the News Atom.

The News Atom is built on four design goals, each addressing how journalism is stored, transported and interpreted:

Verifiability	Can it be trusted?
Interoperability	Can it be understood across systems?
Retrievability	Can it be found?
Reusability	Can it be reused meaningfully?

Grounding the News Atom

A sentence is the smallest self-contained unit of meaning in a news story. It usually contains an idea or concept that can be examined, retrieved, verified and recombined. Sentences are also marked by terminal punctuation marks, making them straightforward for both journalists and machines to identify. For this reason, the News Atom takes the sentence as the fundamental structural unit of epistemic value. (Sentences may carry more than one "proposition" or a basic unit of meaning in discourse.[1] The News Atom schema accounts for complexity, as explained in the field-by-field description of the metadata blueprint.)

How the News Atom connects to the larger inventory of a news organisation

Sentences are grounded in events. Teun A. van Dijk, who has provided foundational research on news discourse, classifies events under a "situation", which journalists transform into news.[2] And sociolinguist Allan Bell said, "Events must contain actors and action."[3]

For the News Atom, I have chosen this definition so that it can be codified more deterministically. Each "event" will be identified by an actor, action, object and location and labelled accordingly. Events are extracted at the container-level (text article, podcast, video). Each event can be linked to related events in a repository or event-bank. The event-bank can become a component of the larger archive.

[1] van Dijk, T.A (1988). Discourse and Communication. Structures of News in the Press. De Gruyter.
[2] *Ibid.*
[3] Bell, A. (2005). The Language Of Time: A Reader. Oxford.

The News Atom schema, v1.0

The News Atom is a metadata blueprint made up of the 15 fields, which covers identity and governance, epistemic signals, provenance and relational context. Its canonical format is JSON (JavaScript Object Notation), a lightweight, text-based format used to store and exchange data.

Its structure makes it easy for journalists to inspect and understand, and for machines to parse. JSON is web-native, widely supported across programming languages and integrates neatly with APIs, databases, and semantic web frameworks like schema.org.

This schema is version-controlled to allow for future evolution without breaking compatibility, and is designed for both live content pipelines and long-term archival.

While JSON is the canonical format, it can also be serialised into formats like NDJSON for streaming (where each entry is in a newline, and this enables line-by-line parsing), and Parquet for large-scale data analytics, without losing fidelity.

The full JSON schema of the News Atom v1.0 is attached in Appendix 2. Below is the top-level structure of the News Atom, without its nested objects:

```JSON
{
  "atom_id",
  "version",
  "atom_status",
  "supersedes_atom_id",
  "knowledge_frame",
  "statement",
  "semantic_frame",
  "primary_expression",
  "media_anchor",
  "event_frame",
  "topic_id",
  "language",
  "review_process",
  "origin",
  "license"
}
```

This compact view serves as the index for the following field-by-field description in detail:

atom_id

The *atom_id* provides a globally unique identifier for every single News Atom. It creates an immutable address for each sentence-level piece of journalistic information, similar to how URLs identify web pages. This allows specific facts to be referenced, verified, and attributed – even when separated from the original article context.

```
JSON
"atom_id": {
  "type": "string",
  "pattern": "^[A-Z]{3}[0-9]{4,}$",
  "description": "Unique identifier: 3-letter organisation code + 4+ digit
sequence"
}
```

atom_id has a 3-letter organisation code to prevent ID collisions between publishers. The numeric sequence is an incrementing system that is human readable, and provides unlimited scalability as content libraries grow. Example:

```
JSON
{
  "atom_id": "SCR0001"
}
```

version

Each News Atom's *version* field tracks the schema version used to create an atom, ensuring compatibility as the metadata blueprint evolves over time. This prevents parsing errors and ensures correct interpretation when different versions of the schema co-exist in the same ecosystem.

```
JSON
"version": {
  "type": "string",
  "pattern": "^v[0-9]+\\.[0-9]+$",
  "description": "Schema version number in semantic versioning format",
```

```
}
```

The News Atom follows industry standard versioning practices, which indicate minor and major changes to the schema. Example:

```JSON
{
  "version": "v1.0"
}
```

atom_status and *supersedes_atom_id*

Sentences in news stories are updated, corrected or retracted on many occasions. This field accounts for those changes made at the container-level and documents it. *atom_status* and *supersedes_atom_id* work together to manage the complete lifecycle of an atom, and ensure transparency and traceability.

```JSON
"atom_status": {
  "type": "string",
  "enum": ["active", "superseded", "retracted", "draft"],
  "description": "Current lifecycle status of the atom",
  "default": "active"
},
"supersedes_atom_id": {
  "type": "string",
  "pattern": "^[A-Z]{3}[0-9]{4,}$",
  "description": "ID of the previous atom this one replaces; omit if
original"
}
```

In *atom_status*, an “active” status represents verified information OK for use, “superseded” atoms have been replaced but remain accessible for audit trails, ‘retracted’ atoms contain inaccurate information, and ‘draft’ is marked when atoms are under editorial review. All atoms are marked as active, superseded or retracted as required. ‘Draft’ is manually added when review is not immediate or needs deliberation.

When an atom supersedes another, the *supersedes_atom_id* creates a direct reference, automatically triggering the previous atom's status to 'superseded'. This means a new atom is created with every update. Example:

```JSON
{
  "atom_status": "active",
  "supersedes_atom_id": "SCR0001"
}
```

knowledge_frame

knowledge_frame is the heart of the News Atom. It classifies the main epistemic role and journalistic function of each sentence within a news story. It captures both the high-level purpose and the specific editorial techniques used to structure information. Each sentence, which becomes an atom, is first classified into an *observed_fact* or *sensemaking*. This separates the 'what happened' from 'what it means'. After this, the sentence is classified into one of nine *knowledge_types*:

- *action*,
- *reaction*,
- *consequence*,
- *context*,
- *evaluation*,
- *expectation*,
- *previous_episode*,
- *history* and
- *narrative*.

Where more precision is identified, the sentence is further divided into conditional sub-types across:

- *reaction*,
- *consequence*,
- *context*,
- *evaluation* and
- *expectation*.

Every sentence has one primary *knowledge_type* based on its journalistic function.

knowledge_frame also captures in-story attribution (as intended by Bell, in his News Text structure) as a whole sentence: “According to the police…”, “Reuters reported…”, “The MET department confirmed…”

It also provides a boolean field to pay special attention to direct quotes within a sentence. Example:

```
JSON
{
  "knowledge_frame": {
    "type": "object",
    "description": "Epistemic and typological classification for this
atom.",
    "required": ["information_type", "knowledge_type", "direct_quote"],
    "additionalProperties": false,
    "properties": {
      "information_type": {
        "type": "string",
        "enum": ["observed_fact", "sensemaking"],
        "description": "Binary epistemic flag to distinguish between what
happened and what it means."
      },
      "knowledge_type": {
        "type": "string",
        "enum": [
          "action",
          "reaction",
          "consequence",
          "context",
          "previous_episode",
          "history",
          "narrative",
          "evaluation",
          "expectation"
        ],
        "description": "Primary journalistic category."
      },
      "subtype": {
        "type": "string",
        "description": "Optional refinement; valid only for certain
knowledge types."
      },
      "source": {
        "oneOf": [
          { "type": "string", "minLength": 1 },
          { "type": "array", "minItems": 1, "items": { "type": "string",
"minLength": 1 } }
```

```
        ],
        "description": "Full in-sentence attribution phrase(s) as printed -
captures both substantive and reporting attribution."
      },
      "direct_quote": {
        "type": "boolean",
        "description": "True if the sentence contains a direct quotation."
      }
    },
    "allOf": [
      {
        "if": { "properties": { "knowledge_type": { "const": "reaction" } }
},
        "then": {
          "required": ["subtype"],
          "properties": {
            "subtype": { "enum": ["claim", "allegation",
"position_statement", "denial", "appeal"] }
          }
        }
      },
      {
        "if": { "properties": { "knowledge_type": { "const": "consequence" }
} },
        "then": {
          "required": ["subtype"],
          "properties": {
            "subtype": { "enum": ["trend", "statistical_outcome",
"immediate_outcome"] }
          }
        }
      },
      {
        "if": { "properties": { "knowledge_type": { "const": "context" } }
},
        "then": {
          "required": ["subtype"],
          "properties": {
            "subtype": { "enum": ["analysis", "definition", "comparison",
"methodology"] }
          }
        }
      },
      {
        "if": { "properties": { "knowledge_type": { "const": "evaluation" }
} },
        "then": {
          "required": ["subtype"],
```

```
          "properties": {
            "subtype": { "enum": ["proposal", "risk_assessment",
"responsibility"] }
          }
        }
      },
      {
        "if": { "properties": { "knowledge_type": { "const": "expectation" }
} },
        "then": {
          "required": ["subtype"],
          "properties": {
            "subtype": { "enum": ["forecast", "prediction", "schedule",
"scenario", "speculation"] }
          }
        }
      },
      {
        "if": { "properties": { "knowledge_type": { "enum": ["action",
"previous_episode", "history", "narrative"] } } },
        "then": { "not": { "required": ["subtype"] } }
      },
      {
        "if": { "properties": { "direct_quote": { "const": true } } },
        "then": { "required": ["source"] }
      }
    ]
  }
}
```

(The logic and definitions used in creating *knowledge_type*, along with identified edge cases are explained in detail in the next section.) Example:

JSON

```
{
  "knowledge_frame": {
    "information_type": "ObservedFact",
    "knowledge_type": "Reaction",
    "knowledge_subtype": "Denial",
    "source": "According to the police,",
    "direct_quote": false
  }
}
```

statement

statement structures the grammatical and semantic content of each sentence in a news story (which is written in natural language) into machine-readable components while preserving original meaning and context. It decomposes every sentence into subject-object-predicate and anchors it in temporal and spatial context, if available.

JSON

```
"statement": {
  "type": "object",
  "description": "Structured representation of the sentence's grammatical
and semantic content",
  "required": ["subject", "predicate", "object", "original_text"],
  "additionalProperties": false,
  "properties": {
    "subject": {
      "oneOf": [
        { "type": "string" },
        { "type": "array", "items": { "type": "string" } }
      ],
      "description": "Who or what is performing the action"
    },
    "predicate": {
      "oneOf": [
        { "type": "string" },
        { "type": "array", "items": { "type": "string" } }
      ],
      "description": "The action, state or relationship being described"
    },
    "object": {
      "oneOf": [
        { "type": "string" },
        { "type": "array", "items": { "type": "string" } }
      ],
      "description": "What the action is being performed on or toward"
    },
    "date": {
      "type": "string",
      "format": "date",
      "description": "When the action occurred (only if specified in the
sentence)"
    },
    "location": {
      "type": "string",
      "description": "Where the action occurred (only if specified in the
sentence)"
    },
```

```
    "original_text": {
      "type": "string",
      "description": "The exact sentence as it appears in the source"
    }
  }
}
```

statement captures:

- *subject*: who or what is performing the action.
- *predicate*: the action, state or relationship being described.
- *object*: what the action is being performed on or toward.
- *date*: if available, when this statement was made or when the described action occurred (different than publication date)
- *location*: if available, where this statement was made or where the described action occurred.
- *original_text*: the complete original sentence, as it appears in the source.

When a sentence contains more than one "proposition" or a basic unit of meaning in discourse, the News Atom is designed to capture each one. Example:

> *A 32-year-old man died and 95 others were injured during dahi handi celebrations in Mumbai on Saturday, The Hindu reported*, from a published article on Scroll.

This complex sentence has two subjects, two predicates and one object (and is captured as an array).[4]

```
JSON
{
  "statement": {
    "subject": ["A 32-year-old man", "95 others"],
    "predicate": ["died", "were injured"],
    "object": ["during dahi handi celebrations", "during dahi handi
celebrations"],
    "date": "2025-08-16",
```

[4] OpenText. (2025). JSON Array Structure.

```
    "location": "Mumbai",
    "original_text": "A 32-year-old man died and 95 others were injured
during dahi handi celebrations in Mumbai on Saturday, The Hindu reported."
  }
}
```

statement is also designed to capture the exact temporal and spatial context, when available. Take this sentence:

> *External Affairs Minister S Jaishankar on Thursday said he was "very perplexed" by the United States imposing punitive tariffs on India for purchasing Russian oil, reported The Hindu,* from an article published on Scroll.

This sentence doesn't carry a specific location (even though it is captured and connected at the event-level and article-level), which is why it is skipped. But it does carry temporal context – Thursday. The article was published on Friday, 22 August 2025. But the atom is able to capture "21 August 2025" because this sentence carries the word "Thursday" – ensuring the occurrence of the event (not the publication date of the sentence) is captured.

```
Java
{
  "subject": "External Affairs Minister S Jaishankar",
  "predicate": "said he was \"very perplexed\" by",
  "object": "the United States imposing punitive tariffs on India for
purchasing Russian oil",
  "date": "2025-08-21",
  "original_text": "External Affairs Minister S Jaishankar on Thursday said
he was \"very perplexed\" by the United States imposing punitive tariffs on
India for purchasing Russian oil, reported The Hindu."
}
```

Also, attribution (as present in the example above) is skipped when it is supplementary, and captured in *knowledge_frame.source* (as seen in the previous section). In the above sentence "… reported *The Hindu*" is supplementary attribution, and is captured in *knowledge_frame.source*. Take this sentence:

```
The details about the remaining judges will be
published on its website when their statement of
assets are received, the court said, from a
published article on Scroll.
```

When attribution is the main action (as in the example below, "… the court said."), it is captured:

```
JSON
{
  "statement": {
    "subject": "the court",
    "predicate": "said",
    "object": "The details about the remaining judges will be published on
its website when their statement of assets are received",
    "original_text": "The details about the remaining judges will be
published on its website when their statement of assets are received, the
court said."
  }
}
```

semantic_frame

semantic_frame links journalistic content to structured knowledge bases and standardised event frameworks increasing the efficiency of atoms and its machine-interpretability in the information ecosystem. It identifies and categorises named entities (people, places, organisations and concepts) and links to authoritative knowledge bases, Wikidata and GeoNames.[5, 6] It also uses the FrameNet's standardised semantic frames and roles to structure reported events.[7]

When entities or frames can't be captured under these conventions, custom descriptions are triggered.

```
JSON
"semantic_frame": {
  "type": "object",
  "description": "Links to structured knowledge and flexible event
frameworks",
  "required": ["entities", "semantic_grounding"],
```

[5] Wikidata. (2025). Retrieved from https://tinyurl.com/2dcttvnp
[6] GeoNames. (2025). Retrieved from https://www.geonames.org/
[7] FrameNet. (2025). About FrameNet. Retrieved from https://shorturl.at/NwA0I

```
  "additionalProperties": false,
  "properties": {
    "entities": {
      "type": "array",
      "minItems": 1,
      "items": {
        "type": "object",
        "required": ["name", "type"],
        "additionalProperties": false,
        "properties": {
          "name": { "type": "string" },
          "type": {
            "type": "string",
            "enum": ["person", "organization", "location", "event",
"concept", "date", "quantity"]
          },
          "wikidata_id": {
            "type": "string",
            "pattern": "^Q[0-9]+$",
            "description": "Optional Wikidata identifier (Q-number)"
          },
          "geonames_id": {
            "type": "string",
            "pattern": "^[0-9]+$",
            "description": "Optional GeoNames identifier for locations"
          }
        }
      }
    },
    "semantic_grounding": {
      "type": "array",
      "minItems": 1,
      "items": {
        "type": "object",
        "required": ["frame_type", "frame_name", "roles"],
        "additionalProperties": false,
        "properties": {
          "frame_type": {
            "type": "string",
            "enum": ["framenet", "custom"]
          },
          "frame_name": { "type": "string" },
          "roles": {
            "type": "object",
            "additionalProperties": { "type": "string" }
          }
        }
      }
```

```
    }
  }
}
```

Take the sentence:

> *A 32-year-old man died and 95 others were injured during dahi handi celebrations in Mumbai on Saturday, The Hindu reported,* from a published article on Scroll. [8]

Here, the entities are not named and not grounded in Wikidata. So:

```JSON
{
 "semantic_frame": {
   "entities": [
     {
       "name": "32-year-old man",
       "type": "person"
     },
     {
       "name": "95 others",
       "type": "person"
     },
     {
       "name": "Mumbai",
       "type": "location",
       "wikidata_id": "Q1156",
       "geonames_id": "1275339"
     },
     {
       "name": "dahi handi",
       "type": "event",
       "wikidata_id": "Q28164099"
     },
     {
       "name": "The Hindu",
       "type": "organization",
       "wikidata_id": "Q926175"
```

[8] Scroll.in. (2025). Mumbai: Two dead, 95 injured in dahi handi celebrations. Retrieved from https://shorturl.at/3Zef1

```
      }
    ],
    "semantic_grounding": [
      {
        "frame_type": "framenet",
        "frame_name": "Death",
        "roles": {
          "protagonist": "32-year-old man",
          "place": "Mumbai",
          "time": "Saturday",
          "containing_event": "dahi handi celebrations"
        }
      },
      {
        "frame_type": "framenet",
        "frame_name": "Experience_bodily_harm",
        "roles": {
          "experiencer": "95 others",
          "place": "Mumbai",
          "time": "Saturday",
          "containing_event": "dahi handi celebrations"
        }
      },
      {
        "frame_type": "framenet",
        "frame_name": "Statement",
        "roles": {
          "speaker": "The Hindu",
          "message": "death and injuries during dahi handi celebrations",
          "topic": "Mumbai incident"
        }
      }
    ]
  }
}
```


Comparatively, in this example:

> *President Donald Trump on Friday said that technology company Apple could face a 25% tariff on iPhones sold in the United States if they were not manufactured in the country*, from a published article on Scroll.

Here the entities are grounded in Wikidata and Geonames, so:

JSON


```
{
  "semantic_frame": {
    "entities": [
      {
        "name": "Donald Trump",
        "type": "person",
        "wikidata_id": "Q22686"
      },
      {
        "name": "Apple",
        "type": "organization",
        "wikidata_id": "Q312"
      },
      {
        "name": "United States",
        "type": "location",
        "wikidata_id": "Q30",
        "geonames_id": "6252001"
      },
      {
        "name": "iPhone",
        "type": "concept",
        "wikidata_id": "Q2766"
      }
    ],
    "semantic_grounding": [
      {
        "frame_type": "framenet",
        "frame_name": "Statement",
        "roles": {
          "speaker": "Donald Trump",
          "message": "Apple could face 25% tariff on iPhones if not
manufactured domestically",
          "time": "Friday",
          "topic": "trade policy"
        }
      },
      {
        "frame_type": "framenet",
        "frame_name": "Imposing_obligation",
        "roles": {
          "agent": "United States government",
          "duty": "25% tariff",
          "responsible_party": "Apple",
          "condition": "if iPhones not manufactured in country"
        }
      }
    ]
```

```
  }
}
```

Take another example:

> *This is because many such high-rises, which can have 40 or more floors, do not allow deliverymen like Sayyed to use the main lifts, which are reserved for residents,* from [a published article in Scroll.]

In this sentence from a long-form news feature article, all the entities are not grounded in Wikidata or Geonames, but captured.

```json
{
 "semantic_frame": {
   "entities": [
     {
       "name": "high-rises",
       "type": "concept"
     },
     {
       "name": "deliverymen",
       "type": "person"
     },
     {
       "name": "Sayyed",
       "type": "person"
     },
     {
       "name": "main lifts",
       "type": "concept"
     },
     {
       "name": "residents",
       "type": "person"
     }
   ],
   "semantic_grounding": [
     {
       "frame_type": "framenet",
```

```
        "frame_name": "Deny_or_grant_permission",
        "roles": {
          "authority": "many such high-rises",
          "protagonist": "deliverymen like Sayyed",
          "action": "use the main lifts",
          "circumstances": "lifts reserved for residents"
        }
      },
      {
        "frame_type": "framenet",
        "frame_name": "Reserving",
        "roles": {
          "reserver": "high-rises",
          "reserved": "main lifts",
          "booker": "residents"
        }
      }
    ]
  }
}
```

Both *statement* and *semantic_grounding* are borrowed from David Caswell's work in structured journalism.[9]

primary_expression and *media_anchor*

Text is a dominant form of communicating journalism, but it isn't the only form. Journalism is also rendered in audio and video forms. Is the News Atom only for text, and not for audio and video? The fields of *primary_expression* and *media_anchor* help answer this question.

A text article can be converted into machine-readable atoms easily. But for a podcast or a video to be atomised, it must first be transcribed and converted to a text format, then each sentence needs to be time-stamped and then individual atoms can be extracted.

Primary_expression captures if a news story is a text article, podcast or video. If it is a text article, a *media_anchor* is not required. But if it is a podcast or a video, *media_anchor* carries the timestamped transcript, which can be atomised as text.

[9] Youtube/David Caswell. (2015). Structured Stories Demo.

JSON

```
"primary_expression": {
  "type": "object",
  "description": "How journalism was originally conceived and structured,
based on The Directory of Liquid Content taxonomy",
  "required": ["content_type", "content_format", "title"],
  "additionalProperties": false,
  "properties": {
    "content_type": {
      "type": "string",
      "pattern": "^CT[0-9]+$",
      "description": "Directory of Liquid Content code for content type
(e.g., CT1, CT2)"
    },
    "content_format": {
      "type": "string",
      "pattern": "^CF[0-9]+$",
      "description": "Directory of Liquid Content code for content format
(e.g., CF1, CF2)"
    },
    "title": {
      "type": "string",
      "description": "The headline or title of the original work"
    }
  }
},
"media_anchor": {
  "type": "object",
  "description": "Technical bridge from multimedia to text atomization (only
for non-text sources)",
  "required": ["modality", "file_url", "transcript_text", "timestamp_start",
"timestamp_end"],
  "additionalProperties": false,
  "properties": {
    "modality": {
      "type": "string",
      "enum": ["audio", "video"],
      "description": "Whether source is audio or video"
    },
    "file_url": {
      "type": "string",
      "format": "uri",
      "description": "Direct link to the multimedia file (can be from
YouTube, Spotify, or any repository)"
    },
    "transcript_text": {
      "type": "string",
      "description": "The transcribed text that became this atom"
    },
```

```
    "timestamp_start": {
      "type": "string",
      "pattern": "^\\d{2}:\\d{2}:\\d{2}\\.\\d{3}$",
      "description": "When this sentence begins in the audio/video"
    },
    "timestamp_end": {
      "type": "string",
      "pattern": "^\\d{2}:\\d{2}:\\d{2}\\.\\d{3}$",
      "description": "When this sentence ends in the audio/video"
    }
  }
}
```

In this example, the *primary_expression* is a text article and doesn't need a *media_anchor*.

content_type and *content_format* are classifiers based on The Directory of Liquid Content, created by the author to map the structural layer of journalism.[10]

```
JSON
{
  "primary_expression": {
    "content_type": "CT1",
    "content_format": "CF2",
    "title": "Make iPhones in US or face 25% tariff: Donald Trump tells
Apple"
  }
}
```

Whereas in this case, from a podcast:

```
JSON
{
  "primary_expression": {
    "content_type": "CT3",
    "content_format": "CF7",
    "title": "Rush Hour, Episode 42"
  },
  "media_anchor": {
    "modality": "audio",
    "file_url": "https://www.youtube.com/watch?v=xyzabc",
```

[10] The Directory of Liquid Content. (2025).

```
    "transcript_text": "The president said that Apple could face a
twenty-five percent tariff on iPhones.",
    "timestamp_start": "00:12:34.500",
    "timestamp_end": "00:12:41.200"
  }
}
```

The *file_URL* can capture the location of the audio or video file from any URL it is stored in: for example, Youtube, Spotify, or a repository.

event_frame

The foundational information block of news and journalism is events. Most iterations of journalism come from "what happened". This is why sentence-based atoms (structural) are mapped to events (information) in the News Atom. *event_frame* captures this mapping. *Event_frames* are meant to be extracted when an LLM parses a "primary expression" (text article, podcast or video).

JSON

```
"event_frame": {
  "type": "array",
  "minItems": 1,
  "description": "Canonical event(s) this atom refers to.",
  "items": {
    "type": "object",
    "required": ["event_id", "event_label"],
    "properties": {
      "event_id": {
        "type": "string",
        "pattern": "^[A-Z]{3}[0-9]{4,}$",
        "description": "Event ID: org-local, opaque, stable (e.g.,
SCR1001)."
      },
      "event_label": {
        "type": "string",
        "description": "Human-readable event name: [DATE] [PRIMARY_ACTOR]
[ACTION_CODE] [OBJECT] [@LOCATION]."
      }
    }
  }
}
```

event_frame consists of an organisation-level *event_id* (similar in construction to *atom_id*) and *event_label*, which identifies each event by Date, Actor, Action, Object and Location (as per Allan Bell's definition of events). This makes events easily readable and sortable by journalists, as well as machines. Example:

```
JSON
{
  "event_frame": [
    {
      "event_id": "SCR1082",
      "event_label": "2025-05-23 Trump threatens Apple tariff @US"
    }
  ]
}
```

How, and if to make *event_frame* cross-organisational is a question for the next iteration of the News Atom. Currently, *event_frame* is designed to manage story coherence within an organisation, rather than a universal event tracking system.

topic_ids

The *topic_ids* field provides standardised thematic classification based on IPTC's *Media Topic NewsCodes*.[11] This enables consistent categorisation across different organisations and information systems.

```
JSON
"topic_ids": {
  "type": "array",
  "minItems": 1,
  "items": {
    "type": "string",
    "pattern": "^medtop:[0-9]{8}$"
  },
  "description": "IPTC MediaTopic codes for thematic classification"
}
```

When *topic_ids* are connected to *event_id* and *article_id*, they enable story arc reconstruction across multiple articles and audio and video outputs. Example:

[11] IPTC. (2025). IPTC Media Topic NewsCodes as of 2025-08-13 (language: en-GB).

> *The Supreme Court on Monday made public the details of the assets owned by 21 out of its 33 judges including Chief Justice Sanjiv Khanna, published on Scroll.*[12]

The *topic_IDs*, according to IPTC's Media Topic NewsCodes are:
Medtop:20001287 Supreme and High Court
Medtop:20000110 Judge
Medtop:20000093 Corruption (or anti-corruption)
Medtop:20000621 Policy

Therefore:

```JSON
{
  "topic_ids": ["medtop:20001287", "medtop:20000110", "medtop:20000093",
"medtop:20000621"],
}
```

language

This field is a standard metadata field, which specifies the language of an atom. The News Atom has currently been tested only for English, but theoretically it can be expanded to include other languages. This field enables that inclusion. It uses the two-letter ISO 639-1 language codes for universal compatibility.

```JSON
"language": {
  "type": "string",
  "pattern": "^[a-z]{2}$",
  "minLength": 2,
  "maxLength": 2,
  "description": "Two-letter ISO 639-1 language code",
  "examples": ["en", "hi", "es", "fr", "de", "zh", "ar"]
}
```

The language of this atom is English, and is captured as "en".

[12] Scroll.in. (2025). Supreme Court publishes assets of 21 of 33 judges.

```
JSON
{
  "language": "en"
}
```

review_process

The *review_process* introduces a human element to the atomisation. In the future, agentic steps could be added to the review process but in this version of the News Atom, *review_process* captures which model was used to annotate the atom, and when. The journalist-reviewer then corrects or updates a field, if required.

```
JSON
"review_process": {
  "type": "object",
  "required": ["automated_annotation", "human_review"],
  "additionalProperties": false,
  "properties": {
    "automated_annotation": {
      "type": "object",
      "required": ["annotated_by", "timestamp"],
      "properties": {
        "annotated_by": {
          "type": "string",
          "description": "The LLM model that performed the initial
annotation"
        },
        "timestamp": {
          "type": "string",
          "format": "date-time",
          "description": "When the automated annotation was completed"
        }
      }
    },
    "human_review": {
      "type": "object",
      "required": ["status"],
      "properties": {
        "status": {
          "type": "string",
          "enum": ["reviewed", "pending", "not_required"],
          "description": "Current review status"
        },
        "reviewer_id": {
          "type": "string",
```

```
          "description": "Identifier of the human reviewer"
        },
        "changes_made": {
          "type": "array",
          "items": {
            "type": "string",
            "enum": [
              "corrected_knowledge_frame",
              "corrected_statement",
              "corrected_semantic_frame",
              "updated_knowledge_frame",
              "updated_statement",
              "updated_semantic_frame",
              "no_changes_needed"
            ],
            "description": "List of corrections or updates made during
review"
          }
        },
        "timestamp": {
          "type": "string",
          "format": "date-time",
          "description": "When the human review was completed"
        }
      }
    }
  },
  "allOf": [
    {
      "if": { "properties": { "human_review": { "properties": { "status": {
"const": "reviewed" } } } } },
      "then": { "properties": { "human_review": { "required":
["reviewer_id", "timestamp"] } } }
    }
  ]
}
```

Here is an example of an atom annotated by Open AI's GPT-4o model, and reviewed by "editor_123". They have corrected *knowledge_frame* and updated *semantic_frame*.

JSON

```
{
 "review_process": {
   "automated_annotation": {
     "annotated_by": "LLM-GPT4o",
```

```
      "timestamp": "2025-05-06T14:30:00Z"
    },
    "human_review": {
      "status": "reviewed",
      "reviewer_id": "editor_123",
      "changes_made": ["corrected_knowledge_frame",
"updated_semantic_frame"],
      "timestamp": "2025-05-06T15:45:00Z"
    }
  }
}
```

origin

origin provides essential publication metadata by establishing who published the story, when it was published and where it can be found. This field ensures that every sentence-level information maintains clear provenance, and supports verification and attribution.

JSON

```
"origin": {
  "type": "object",
  "description": "Publication metadata establishing accountability and
attribution",
  "required": ["organization", "journalist", "url", "created_at"],
  "additionalProperties": false,
  "properties": {
    "organization": {
      "type": "string",
      "description": "Publishing organization name"
    },
    "journalist": {
      "type": "string",
      "description": "Author or reporter byline"
    },
    "url": {
      "type": "string",
      "format": "uri",
      "description": "Canonical URL of the source article"
    },
    "created_at": {
      "type": "string",
      "format": "date-time",
      "description": "Publication timestamp of the original article"
    },
```

```
    "source_article_id": {
      "type": "string",
      "description": "Stable identifier of the article in the publisher's
CMS"
    }
  }
}
```

For example:

> *Make iPhones in US or face 25% tariff: Donald Trump tells Apple*, a published article on Scroll.[13]

Becomes:

```
JSON
{
 "origin": {
   "organization": "Scroll.in",
   "journalist": "Scroll Staff",
   "url":
"https://scroll.in/latest/1082730/make-iphones-in-us-or-face-25-tariff-donal
d-trump-tells-apple",
   "created_at": "2025-05-23T19:07:00Z",
   "source_article_id": "1082730"
 }
}
```

license

The *license* field allows embedding a terms-of-use URL at the atom level. This is a flexible field, which can be tailored based on the needs of a news organisation. It can also be filtered by *knowledge_frame.information_type* to allow for syndication, AI grounding, and other licensing utilities.

```
JSON
"license": {
```

[13] Scroll.in. (2025). Make iPhones in US or face 25% tariff: Donald Trump tells Apple.

```
  "type": "object",
  "required": ["type", "terms_url"],
  "additionalProperties": false,
  "properties": {
    "type": {
      "type": "string",
      "enum": ["all_rights_reserved", "cc_by", "cc_by_nc",
"syndicated_feed"],
      "description": "Standardized license type for this content"
    },
    "terms_url": {
      "type": "string",
      "format": "uri",
      "description": "URL to complete licensing terms and conditions"
    }
  }
}
```

JSON

```
{
  "license": {
    "type": "all_rights_reserved",
    "terms_url": "https://scroll.in/terms"
  }
}
```

A reminder now that I said at the outset that the News Atom is built on four design goals, each addressing how journalism is stored, transported and interpreted:

Verifiability	Can it be trusted?
Interoperability	Can it be understood across systems?
Retrievability	Can it be found?
Reusability	Can it be reused meaningfully?

How does the schema described stack up against these goals?

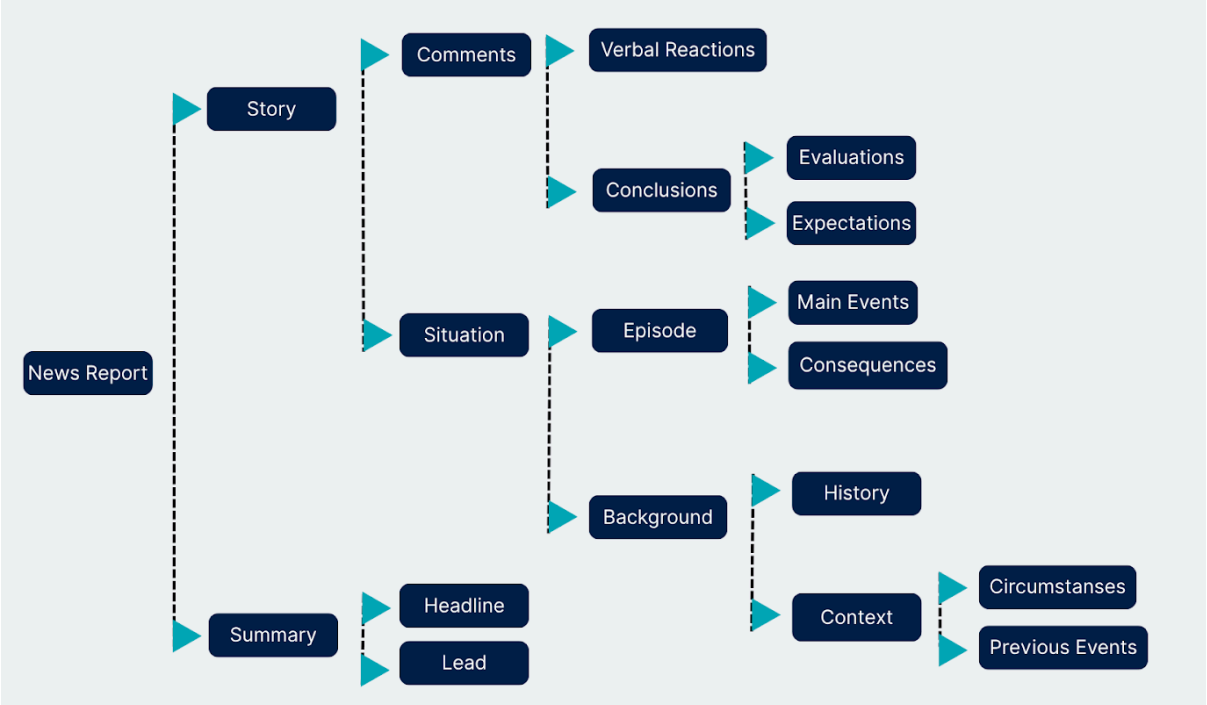
Evaluation of the News Atom schema against design goals

Field	Verifiability	Retrievability	Reusability	Interoperability
atom_id	Yes	Yes	Yes	Yes
version	Yes			Yes
atom_status	Yes	Yes	Yes	
supersedes_atom_id	Yes	Yes	Yes	
knowledge_frame	Yes	Yes	Yes	
statement	Yes	Yes	Yes	Yes
semantic_frame	Yes	Yes	Yes	Yes
primary_expression	Yes		Yes	
media_anchor	Yes		Yes	
event_frame	Yes	Yes	Yes	Yes
topic_IDs		Yes	Yes	Yes
language		Yes	Yes	Yes
review_process	Yes			
origin	Yes			
license			Yes	

Deep-dive into *knowledge_frame*

As we have seen, journalism isn't just a pile of facts. There is order and structure to how facts are transformed into journalism. The logic of this order and structure has been studied for decades. In his 1988 book, *News as Discourse*, Teun A. van Dijk presented the "News Schemata" and formalised how news is organised.[14]

His work is foundational to understanding the implicit logic baked into a news story, and news discourse at large.



Graphic representation of Teun A. van Dijk's News Schemata, 1988.[15]

In 1991, Allan Bell provided a finer and a more comprehensive look at how news is structured and the cues it carries in his book, *The Language of News Media*.

[14] van Dijk, T.A (1988). Discourse and Communication. Structures of News in the Press.

[15] *Ibid.*

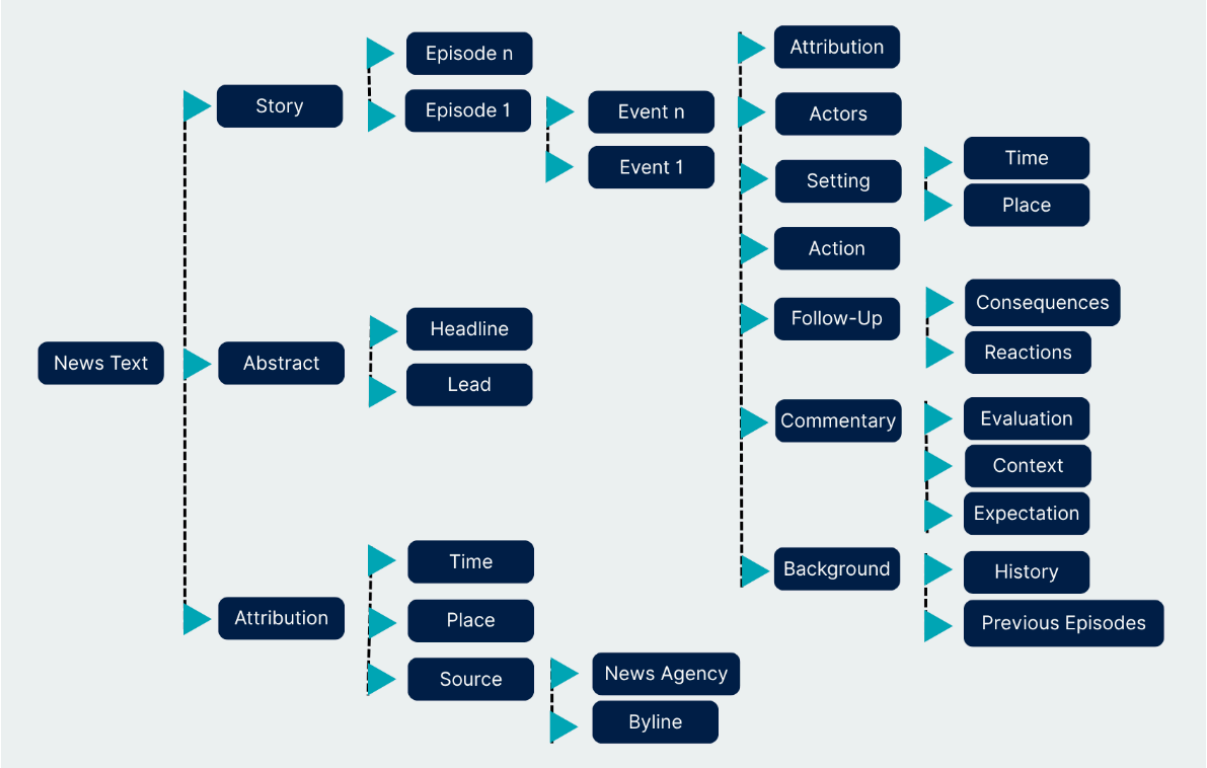
His structure documented “attribution” (both of-story and in-story) as a trust signal, and was grounded in events.

In addition, his structure, like van Dijk's, divided news text into Abstract and Story.

Abstract contained Headline and Lead, and Story was subdivided into episodes.

Episodes were further divided into events. Events contained (in-story): Attribution, Actors, Setting, Action, Follow-Up, Commentary, and Background.

The lowest-level structure captured “knowledge types”: concrete enough to be codified (and incorporated into metadata). These were: Time, Place, Consequences, Reactions, Evaluation, Context, Expectation, History, and Previous Episodes.



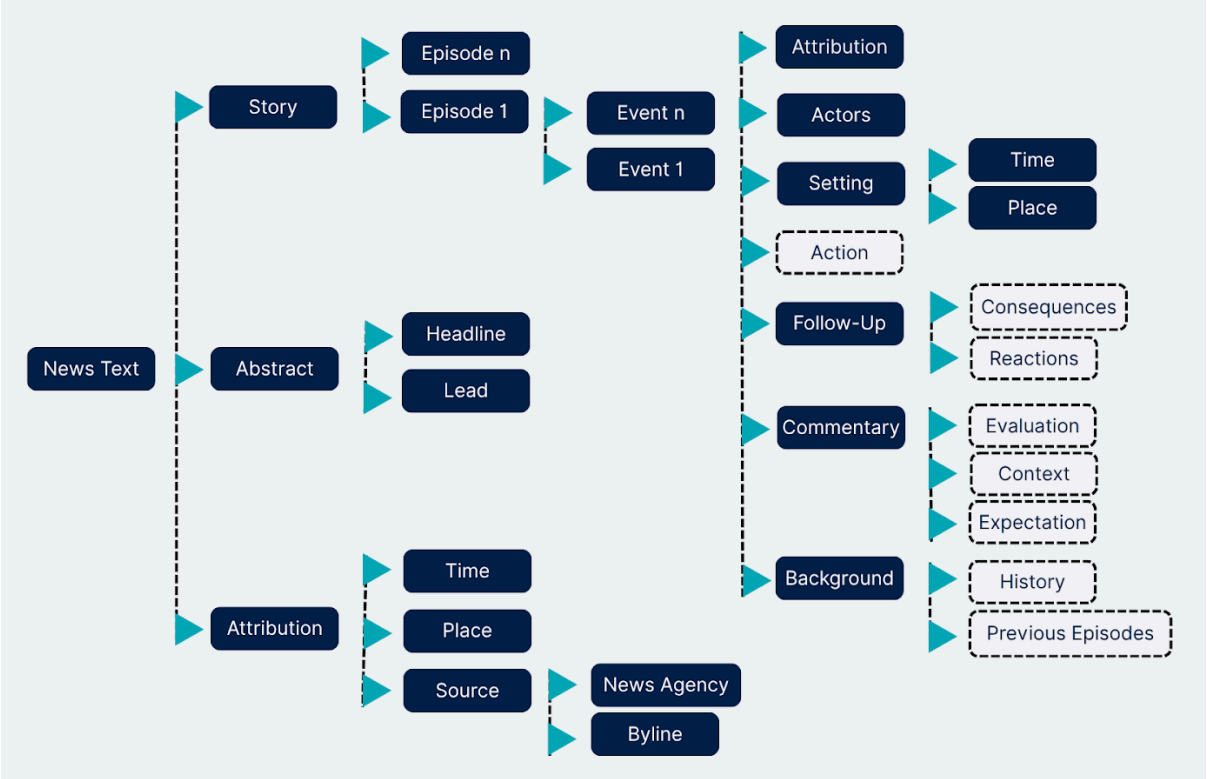
Graphic representation of Allan Bell’s News Text structure, 1991. [16]

In a 2019 paper titled, *Towards the Automatic Analysis of the Structure of News Stories*, Bell’s schema, and especially its lowest-level categories, was tested for machine-usability.[17]

[16] Bell, A. (2005). The Language Of Time: A Reader. Oxford.

[17] Proceedings of the Text2StoryIR’19 Workshop, Cologne, Germany. (2019). Towards the Automatic Analysis of the Structure of News Stories. Retrieved from https://shorturl.at/p18Ot

The categories of Action, Reaction, Consequence, Context, Evaluation, Expectation, Previous Episode and History were then used as labels for sentence-level automatic annotation.



Graphic representation of Allan Bell's News Text structure adapted for machine-usability.[18]

The News Atom's *knowledge_frame* is built on the foundation set by this research, as described above.

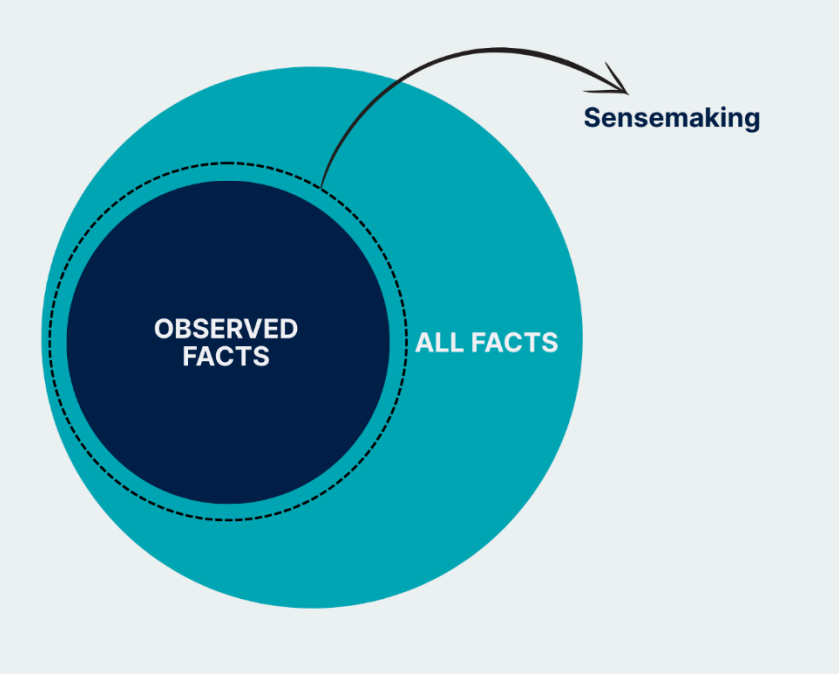
The design of *knowledge_frame*

A fact is defined as something that has an actual existence or is an actual occurrence.[19]

Academic research in journalism studies has reiterated in study after study that meaning-making starts when a choice is made on which facts to report on.[20, 21]

[18] *Ibid.*
[19] Merriam Webster Dictionary. (2025). Fact.
[20] van Dijk, T.A (1985). Discourse and Communication. Structures of News in the Press.
[21] Mast, J & Temmerman, M. (2021). Journalism Studies. What's (The) News? Reassessing "News Values" as a Concept and Methodology in the Digital Age.

After a fact (what happened?) has been reported, another layer of sensemaking (what it means) is added to it. Both van Dijk and Bell's research acknowledges this and calls it 'Comments' and 'Commentary' respectively.



Graphic representation of the metadata field of information_type

The *information_type* field incorporates this binary as *observed_fact* and *sensemaking*, sharpening the documentation of journalism's role as a record of events and a guide to understanding.

Because *information_type* describes the nuance of meaning-making it could be used to determine separate licensing and syndication strategies.

When reversioning content, *observed_fact* atoms could be recombined to create timelines, mindmaps, statistical summaries etc; and *sensemaking* atoms could be recombined to create contextual overlays like a comprehensive "why it matters" or a "big picture" or a "bottom line" module plugged into a news report.

The next field, *knowledge_type* is directly based on Bell's schema (see previous page) and the sentence-level automatic annotation research described above.

According to Bell's schema, every story has a "lead" or a main event that it describes. Every node downstream is pegged to this main event.

These downstream nodes are captured and labelled under *knowledge_type*.

The eight downstream nodes are:

action	What happened?
reaction	Who said what in response?
consequence	What followed from it?
context	What explains or situates it?
evaluation	What significance is assigned?
expectation	What is projected or speculated?
previous_episode	What led up to it in the near past?
history	What long-term past shaped it?

The News Atom also introduces a ninth node called "Narrative" to incorporate scene-setting and other storytelling devices used to set up facts in a story.

narrative	How it feels?

knowledge_type also has optional subtypes for five of the nodes described above. This additional granularity can have a practical effect on how atoms can be organised, discovered and reused at the archive-level.

(The subtypes were arrived at upon studying Scroll.in's The Latest articles from April 1, 2025 to April 30, 2025 and may not be an exhaustive list. This field can be scaled or muted as required). [22]

The subtypes are as follows:

Reaction	Consequence	Context	Evaluation	Expectation
claim	*trend*	*analysis*	*proposal*	*forecast*
allegation	*statistical_outcome*	*methodology*	*risk_assessment*	*prediction*
position_statement	*immediate_outcome*	*definition*	*responsibility*	*schedule*
denial		*comparison*		*scenario*
appeal				*speculation*

For the categories of *action*, *previous_episode*, *history* and *narrative*, subtypes have been avoided.

action is event-oriented, and further granularity will be captured in *statement* and *semantic_frame*.

[22] Scroll.in (2025). The Latest. Retrieved from https://shorturl.at/MLHub

previous_episode and *history* are already well defined by temporal distance, and *narrative* is included to capture non-action, non-consequence, non-reaction, non-evaluation and non-expectation parts of the story.

The next field in *knowledge_frame* is *source*.

```JSON
{
  "source": "the Ministry of External Affairs told Parliament"
}
```

source is designed to capture in-sentence attribution, as intended in Bell's schema. This field documents the exact phrase used to attribute the facts in a sentence.

Take this example:

> *The United States Agency for International Development did not provide $21 million in funding for voter turnout in India, the Ministry of External Affairs told Parliament on Thursday, countering claims made by President Donald Trump in February, from a published article in Scroll.*

source captures the exact phrase used for in-story attribution: "the Ministry of External Affairs told Parliament".

It is also designed to capture a nested and a complex attribution:

Take this example, with nested attribution:

> *Citing a July 2 communication, the ministry said the US embassy had clarified that USAID had not carried out any voter turnout-related activities in India,* from the same published article as above.

Becomes:

```JSON
{
```

```
    "source": "Citing a July 2 communication, the ministry said"
}
```

And the following example, which has complex attribution:

> *West Bengal Chief Minister Mamata Banerjee on Monday claimed that stability has returned to Murshidabad following the violence that took place in the district during protests against the Waqf Act in April, PTI reported, from a published article on Scroll.*

Becomes:

```
JSON
{
  "source": [
    "West Bengal Chief Minister Mamata Banerjee on Monday claimed",
    "PTI reported"
  ]
}
```

In-story attributions will allow answer machines, powered by LLMs, to be more accurate, as intended by the original news story.

The next field in *knowledge_frame* carries this thought forward, and specifically flags if a sentence is a *direct_quote*. Take this example:

> *"If that is not the case, a tariff of at least 25% must be paid by Apple to the US," he added, from a published article on Scroll.*

Because it is a direct quote, the *direct_quote* field will be true.

```
JSON
{
  "knowledge_frame": {
```

```
    "information_type": "observed_fact",
    "knowledge_type": "reaction",
    "subtype": "claim",
    "source": "he added",
    "direct_quote": true
  }
}
```

(In case you were wondering, the "who?" in "he added" will be inferred from all the atoms bunched together via *event_frame*.)

```
JSON
{
  "event_label": "2025-05-23 Donald Trump threatens_tariff Apple iPhones
@US"
}
```

Possible rules for *knowledge_frame*

knowledge_frame is the most interpretive part of the News Atom schema. When the above-described context and schema are provided, both GPT-5 and Gemini 2.5 Flash are able to annotate sentences as specified.[23] But there are obvious errors, which can be fixed with better definition:

Understanding the role of *action*

Every news story, according to Bell's schema, has a lead (which has a main event). This lead is codified as the first *action* in the News Atom. This is the anchor of a news story. *reaction*, *consequence*, *context*, *evaluation*, *expectation*, *previous_episode*, *history* and *narrative* are all pegged to this anchor. This *action* anchor is an *observed_fact*, even in opinion pieces, long-form features or other sensemaking formats.

Facts exist on a continuum (which means, each fact could simultaneously be a *reaction*, *consequence*, *previous_episode* or *history*). Anchoring the facts (contained in sentences) to the main event grounds them within the broader topic, and preserves their contextual meaning.

This example is categorised as *observed_fact* and *action* because it is the lead:

[23] As of August 23, 2025

> *President Donald Trump on Friday said that technology company Apple could face a 25% tariff on iPhones sold in the United States if they were not manufactured in the country,* [*from a published article on Scroll*](#).

Thus:

```
JSON
{
  "knowledge_frame": {
    "information_type": "observed_fact",
    "knowledge_type": "action",
    "source": "President Donald Trump on Friday said",
    "direct_quote": false
  }
}
```

Grammatically, and based on the rules of languages learned by LLMs (and without the rule explained above), this sentence would be classified as *reaction* and *claim* – which isn't incorrect per se, but within this context it would be.

In multiple tests, speech is almost always classified as a *reaction* in both GPT-5 and Gemini 2.5 Flash. But many news stories can be *reaction*-led (or *consequence*-led, *previous_episode*-led) and anchoring and labelling it as *action* reduces the ambiguity in interpreting the context of a story.

The role of *reaction*

reaction is a reaction to the *action* described above. It has five subtypes: *claim*, *allegation*, *position_statement*, *denial* and *appeal*.

Once the anchor event rule is applied, both GPT5 and Gemini 2.5 Flash are able to annotate a sentence containing a reaction, and further classify it by its subtypes.

reaction is an *observed_fact* and is given more granular meaning by the *source* field in *knowledge_frame*.

See how GPT-5 classified the sentence in this example (from the same article as above):

The US president had added that he had told Cook: "India can take care of themselves, they are doing very well."

Becomes:

```json
{
  "knowledge_frame": {
    "information_type": "observed_fact",
    "knowledge_type": "reaction",
    "subtype": "position_statement",
    "source": "The US president had added that he had told Cook",
    "direct_quote": true
  }
}
```

Gemini 2.5 Flash's annotation was identical:

```json
{
  "knowledge_frame": {
    "information_type": "observed_fact",
    "knowledge_type": "reaction",
    "subtype": "position_statement",
    "source": "The US president had added that he had told Cook",
    "direct_quote": true
  }
}
```

The role of *consequence*

Although *consequence* was intended to be an *observed_fact* according to Bell's schema, some instances in modern news writing could be classified as *sensemaking*. Take this example (from the same article as above):

The tariffs had led to concerns of a broader trade war that could disrupt the global economy and trigger recession.

The words "had led to concerns of a broader trade war that could disrupt" indicate that this sentence could be labelled as *consequence* and *trend*, and that this is *sensemaking*.

```JSON
{
  "knowledge_frame": {
    "information_type": "sensemaking",
    "knowledge_type": "consequence",
    "subtype": "trend",
    "source": null,
    "direct_quote": false
  }
}
```

The subtype *trend* could be either an *observed_fact* or *sensemaking*.

The subtypes *immediate_outcome* and *statistical_outcome* are *observed_fact*.

The role of *context*

context is a tricky field (and seems counterintuitive).

Unlike *consequence*, *context* was intended to be "Commentary" according to Bell's schema, or a *sensemaking* layer. But if we examine the subtypes of *context*, this may not be accurate.

Within *context*, *analysis* and *comparison* are *sensemaking* fields, which provide causal interpretation (why things happened) and significance reasoning (what things mean), and comparative positioning (how things relate across different contexts).

But *definition* and *methodology* are *observed_fact*.

Take this sentence:

> *A waqf is an endowment under Islamic law dedicated to a religious, educational or charitable cause, from a published article in Scroll.*

Becomes:

```JSON
{
  "knowledge_frame": {
    "information_type": "observed_fact",
```

```
    "knowledge_type": "context",
    "subtype": "definition",
    "source": null,
    "direct_quote": false
  }
}
```

Take another sentence:

> *Voters born before July 1, 1987, were required to show proof of their date and place of birth, while those born between July 1, 1987, and December 2, 2004, had to also submit documents establishing the date and place of birth of one of their parents, from a published article in Scroll.*

Becomes:

```
JSON
{
  "knowledge_frame": {
    "information_type": "observed_fact",
    "knowledge_type": "context",
    "subtype": "methodology",
    "source": null,
    "direct_quote": false
  }
}
```

The role of *evaluation*

evaluation and its subtypes (*proposal*, *risk_assessment* and *responsibility*) can be classified as both *observed_fact* or *sensemaking* based on the context of the sentence.

Evaluation can come from an actor within a news story. Take this example:

> *If what is coming out of the indictment is material information under the purview of the LODR regulations and if SEBI fi nds that the regulations were violated by the Adani group, then the Adani group and specific individuals could be penalised by SEBI," she said, from a published article in Scroll.*

```JSON
{
  "knowledge_frame": {
    "information_type": "observed_fact",
    "knowledge_type": "evaluation",
    "subtype": "responsibility",
    "source": "she said",
    "direct_quote": true
  }
}
```

And this sentence:

> *The Smart City Mission, in its present avatar, reflects a deeper malaise in Indian urbanism: a desire to appear modern without truly engaging with the social and ecological complexities of cities,* [*from a published article in Scroll.*]

```JSON
{
  "knowledge_frame": {
    "information_type": "sensemaking",
    "knowledge_type": "evaluation",
    "subtype": "responsibility",
    "source": null,
    "direct_quote": false
  }
}
```

evaluation, when attributed to an actor is an *observed_fact* but when it is interpretive and not attributed to an actor, it is *sensemaking*.

The role of *expectation*

Other than *schedule*, which is an *observed_fact* under *expectation*, all other subtypes (*forecast*, *prediction*, *speculation* and *scenario*) can be both *observed_fact* or *sensemaking*. Take this example:

The chief minister added that she would visit Dhuliyan town and provide compensation to persons whose houses and shops were damaged during the violence in April, from a published article on Scroll.

Becomes:

```
JSON
{
  "knowledge_frame": {
    "information_type": "observed_fact",
    "knowledge_type": "expectation",
    "subtype": "schedule",
    "source": "the chief minister added",
    "direct_quote": false
  }
}
```

observed_fact + *expectation* atoms could also be inferred based on the context of the all the atoms clustered under an event, as in this case:

The process of deciding on claims and objections and verifying eligibility documents is slated to be completed by September 25, from a published article on Scroll.

Becomes:

```
JSON
{
  "knowledge_frame": {
    "information_type": "observed_fact",
    "knowledge_type": "expectation",
    "subtype": "schedule",
    "source": null,
    "direct_quote": false
  }
}
```

Take another example:

> *In the case of Adani, where Adani Green Energy could have earned $2 billion (Rs. 16,800 crore) in profits, as per the indictment, fines could be as high as $4 billion (Rs. 33,600 crore), Katarki said, from a published article on Scroll.*

Becomes:

```json
{
  "knowledge_frame": {
    "information_type": "observed_fact",
    "knowledge_type": "expectation",
    "subtype": "forecast",
    "source": "Katarki said",
    "direct_quote": false
  }
}
```

(Note that in the above example, *direct_quote* is marked as false even though it is specifically attributed to an actor in the original sentence. This is because it is not within quotation marks. This is an edge case, and many such edge cases will appear as the News Atom is tested at scale)

Like *evaluation*, when *expectation* is attributed to a specific actor in a news story, it is an *observed_fact*. When it is interpretive and not attributed to an actor, it is *sensemaking*.

Take this example:

> *The tariffs had led to concerns of a broader trade war that could disrupt the global economy and trigger recession, from a published article on Scroll.*

Becomes:

```JSON
{
  "knowledge_frame": {
    "information_type": "sensemaking",
    "knowledge_type": "expectation",
    "subtype": "scenario",
    "source": null,
    "direct_quote": false
  }
}
```

The role of *previous_episode* and *history*

When looked at within a large cluster of events and themes, how do you temporally divide *previous_episode* and *history*?

If we were to take the literal meaning of *previous_episode*, everything before the immediate previous episode of an event would be *history*. But then how do you contextualise this for facts on a continuum?

To solve this, the News Atom will consider every event cluster – events part of an unfolding story with direct causal connections – as the organising principle for temporal classification. Atoms within the same event cluster are classified as *previous_episode* – because they represent an unfolding story with direct causal connections. Events outside of this cluster but still within the same thematic cluster become *history*. Take this example:

> *The case was filed in 2001, when Saxena was heading the Ahmedabad-based non-governmental organisation National Council for Civil Liberties, from a published article in Scroll.*

This article is about an actor (activist Medha Patkar in this case) who was arrested and released hours later in a 24-year-old defamation case. The case may be 24 years old but this update is a direct causal connection to it, making everything before this update within this event cluster a previous_episode.

```JSON
{
  "knowledge_frame": {
    "information_type": "observed_fact",
```

```
    "knowledge_type": "previous_episode",
    "subtype": null,
    "source": null,
    "direct_quote": false
  }
}
```

Take another example:

The Western Ghats Ecology Authority was proposed by the Western Ghats Expert Ecology Panel in its 2011 report – also known as the Gadgil Commission Report, or, simply, the Gadgil Report, from a published article on Scroll.

The article is an op-ed that contextualises a horrific landslide in Wayanad in the southern state of Kerala, India, which claimed 400 lives in 2024.

The landslide itself and its causal connections form one event cluster. But the Gadgil report, which this story is about, provides historical context to the landslide outside of its causal connections.

JSON

```
{
  "knowledge_frame": {
    "information_type": "observed_fact",
    "knowledge_type": "history",
    "subtype": null,
    "source": null,
    "direct_quote": false
  }
}
```

The role of *narrative*

narrative helps immerse a user by setting the scene, providing texture to a story and providing continuity in storytelling. Take this example:

When the monsoon arrived in Delhi last year, it brought welcome respite from the relentless heat, from a published article on Scroll.

```JSON
{
  "knowledge_frame": {
    "information_type": "sensemaking",
    "knowledge_type": "narrative",
    "subtype": null,
    "source": null,
    "direct_quote": false
  }
}
```

How two frontier models define the knowledge_type and its subtypes

> **What is a frontier model?**
>
> A frontier model is a large, cutting-edge AI system that sits at or near the current limits of capability for machine learning models. Examples include: GPT-5 from OpenAI, Claude 3 from Anthropic, Gemini from Google DeepMind. GPT-5 and Gemini both have access to the web, making them more suited to this testing.

Querying frontier models on their "understanding" of terms helps set a baseline benchmark for LLM-based annotation.

Each definition will need to be standardised both at the organisation-level as well as at the industry-level through collective iteration.

This benchmarking also estimates how much finetuning will be needed to describe each term better. The prompt used to generate these tables: "Based on the context and rules provided, define the following terms."

knowledge_type	GPT-5	Gemini 2.5 Flash
Action	The central event or occurrence that anchors the news story. This is the main "what happened?" fact, often expressed in the lead of a story. All other knowledge types are pegged to this anchor.	The primary event or core subject of a news story. This is the central point that everything else in the article relates to. It is always a factual observation, even if the article itself is an opinion piece.
Reaction	A statement, response, or position made by an actor in relation to the action. Captures who said what, how they responded, or what position they took.	A statement or response from an individual or group to the central action. Reactions are factual occurrences themselves, and are often attributed to a specific source.
Consequence	The direct or indirect outcomes that follow from the action. Can be immediate results, measurable effects, or longer-term implications.	What happens as a result of the main action. While some consequences can be straightforward facts (like a statistical outcome), others might

		be more interpretive, such as analysing a potential trend.
Context	Background, definitions, explanations, or methodological details that situate the action within a broader frame. Provides the scaffolding for understanding.	Information that helps explain or situate the main event. This includes providing background details, definitions, or methodology. Depending on its nature, context can be either a factual detail (like a definition) or a layer of interpretation (like an analysis or comparison).
Evaluation	Interpretations, judgments, or assessments that assign meaning, responsibility, or significance to the action or its consequences.	An assigned significance or judgment about the event. This can be a factual statement if it's a judgment from a specific person mentioned in the story (e.g., "she said the penalty would be severe"). It's considered an interpretive element if it's an unattributed comment within the article itself.
Expectation	Projections, forecasts, or anticipations of what may happen next. Covers both attributed plans and speculative scenarios.	Information that projects or speculates on future events. This can be a factual statement, like a scheduled date, or a more interpretive one, like a forecast or a scenario. As with evaluation, if the expectation is attributed to someone, it's a fact. Otherwise, it's a form of sensemaking.
Previous_Episode	Events from the recent past that are directly and causally connected to the current action or story. Belongs to the same unfolding event cluster.	An event from the recent past that is directly connected to the current story. It's a key part of the unfolding narrative and is considered a factual observation.
History	Longer-term or thematically related past events that provide historical background but are not directly part of the current unfolding event cluster.	A past event that provides broader, long-term context for the current story but isn't part of its direct causal chain. This is always a factual element.
Narrative	Scene-setting, descriptive, or atmospheric text that situates the reader emotionally or experientially within the story. Not a factual event but a storytelling device.	Descriptive language used to set a scene or add texture to a story. This element helps create a mood or atmosphere and is always considered a form of sensemaking rather than a factual statement.

Knowledge_Type	Sub_Type	GPT-5	Gemini 2.5 Flash
Reaction	Claim	An assertion made by a source, presented as true but not adjudicated here.	A statement asserts a particular proposition as being true without providing immediate evidence.
	Allegation	A statement that accuses an actor of wrongdoing, fault, or misconduct, presented without adjudication.	A statement that asserts a particular proposition as being true without providing immediate evidence.
	Denial	An explicit rejection or refutation of a stated claim.	A statement refutes or contradicts a specific claim,

			allegation, or assertion.
	Position Statement	A formal declaration of stance or policy by an actor.	An entity formally declares its official stance or viewpoint on an issue.
	Appeal	A request or call to action made by an actor, often urging others to intervene, respond, or support.	A statement made by a source that is a request for a specific action to be taken.
Consequence	Trend	A sustained directional pattern over multiple observations or time.	Information describes a general direction or pattern of change over a period.
	Statistical Outcome	A measured effect tied to the action (e.g., change in level, rate, or count).	The text presents a quantified, measured effect resulting directly from a specific action.
	Immediate Outcome	A discrete follow-on development that occurs directly after the action.	The text reports a direct, discrete event that immediately followed a preceding action.
Context	Analysis	An interpretive reasoning that draws patterns or mechanisms from multiple facts.	The text provides interpretive reasoning to understand events, data, or statements.
	Definition	A statement that specifies what a term, entity, or metric means in this context.	The text states the precise meaning of a specific term or concept.
	Comparison	A contrast that benchmarks then vs. now or here vs. elsewhere to provide scale.	The text juxtaposes two or more subjects across time, geography, or other criteria.
	Methodology	A description of how data or results were produced (sources, samples, methods, scope).	The text describes the specific process used to generate data or research findings.
Evaluation	Proposal	A prescriptive recommendation about what should be done.	The text recommends a specific course of action to be taken in the future.
	Risk Assessment	An evaluation of the likelihood and severity of potential harms or losses.	The text evaluates the probability and potential severity of a negative future outcome.
	Responsibility	An assignment of credit or blame to specific actors.	The text attributes credit or blame for an action or its specific outcome.
Expectation	Forecast	An evidence-based projection grounded in models or systematic indicators.	The text projects future outcomes based on a formal model or existing evidence.
	Prediction	A probabilistic or speculative call about what will happen without a formal model.	The text offers a speculative view about a future event without formal modelling.
	Schedule	A planned or officially announced time frame or date for an expected action.	The text outlines a sequence of planned events with their intended specific timings.
	Scenario	A conditional projection describing outcomes under stated “if/then” conditions.	The text describes a potential future situation contingent on a specific condition.
	Speculation	A conjectural statement about what might happen	It is a probabilistic or speculative statement about

	in the future, lacking supporting evidence or formal modelling.	what might happen in the future, but it is not based on a formal model or evidence.

Next steps and considerations

The News Atom is not an end point. Metadata blueprints like the News Atom need to be stress-tested rigorously: through pilots, and then working groups.

The immediate next step is to get consensus on the interpretive parts; build tiny, verifiable prototypes like a CMS plugin, a small backfilled dataset, or a test of provenance resilience. Each of these steps is about proving what computationally rich journalistic data can unlock.

From here, several questions arise:

- How might users benefit from atom-level metadata?
- What minimum set of fields would be most valuable in a first pilot - what should be left out?
- How might we collaboratively build annotation guidelines?
- How might we store atoms, and how much space will they occupy?
- What amount of journalist overview and automation is needed to atomise legacy archives?
- Could News Atoms be embedded directly into an article's HTML as a hidden script or link?
- How might CMSs automatically generate atom metadata without slowing down newsroom workflows?
- Can schema.org's NewsArticle be extended to carry atom-level metadata?
- What lightweight demonstration would most clearly showcase the value of atoms to sceptics?
- Could embeddings be used to link paraphrased text back to its atom?
- How might we collectively negotiate to preserve journalism's epistemic value in the larger digital information ecosystem?

Finding answers to these questions will not be as easy as designing a metadata blueprint. I hope to collaborate with many of you who have continued to read this far.

Appendix 1: Exif Metadata of an image

Sample image:[24]

File information

Metadata takes 553 KB (14.6%) of this image and includes location data.[25]

File Size	3,877,787 Bytes / 3786.9KB / 3.70MB
File Type	JPEG
File Type Extension	JPG

Image properties

Aspect Ratio	4 / 3
Orientation	6
X Resolution	72
Y Resolution	72

[24] Imagy.app. (2025). View Exif Data Online. Retrieved from https://shorturl.at/9CEeD
[25] Jimpl. (2025). Online EXIF data viewer. Retrieved from https://shorturl.at/yZU5c

ResolutionUnit	2
Color Space	65535
ExifImageWidth	5712
ExifImageHeight	4284
MPImageFormat	0
ColorSpaceData	RGB
Image Width	5712
Image Height	4284
Bits Per Sample	8

Camera Settings

Camera Make	Apple
Camera Model	iPhone 15 Plus
F-Number	1.6
Exposure Program	2
ISO	50
ApertureValue	1.59999999932056
Exposure Compensation	0
Metering Mode	5
Flash	16
Focal Length	5.96
MakerNoteVersion	15
FocusDistanceRange	0.1953125 0.07421875
FocusPosition	157
FlashpixVersion	0100

ExposureMode	0
White Balance	0
FocalLengthIn35mmFormat	26
LensInfo	1.539999962 5.960000038 1.6 2.4
LensMake	Apple
Lens Model	iPhone 15 Plus back dual wide camera 5.96mm f/1.6
Aperture	1.6
FocalLength35efl	26
LensID	iPhone 15 Plus back dual wide camera 5.96mm f/1.6

GPS Data

GPS Latitude Ref	N
GPS Longitude Ref	W
GPSAltitudeRef	0
GPS Time Stamp	13:44:16
GPSSpeedRef	K
GPSSpeed	0.09046229328
GPSImgDirectionRef	T
GPSImgDirection	187.3952637
GPSDestBearingRef	T
GPSDestBearing	187.3952637
GPS Date Stamp	2025:05:10
GPSHPositioningError	4.242100245
GPS Altitude	223.3838065
GPSDateTime	2025:05:10 13:44:16Z

GPS Latitude	51.9762138888889
GPS Longitude	-1.57027222222222
GPSPosition	51.9762138888889 -1.57027222222222

Date & Time

File Modify Date	0000:00:00 00:00:00
Modify Date	2025:05:10 14:44:17
Exposure Time	0.0003219575016
Original Date	2025:05:10 14:44:17
Create Date	2025:05:10 14:44:17
OffsetTime	+01:00
OffsetTimeOriginal	+01:00
OffsetTimeDigitized	+01:00
RunTimeFlags	1
RunTimeValue	628789443516958
RunTimeScale	1000000000
RunTimeEpoch	0
SubSecTime	559
SubSecTimeOriginal	559
SubSecTimeDigitized	559
ProfileDateTime	2022:01:01 00:00:00
DigitalCreationTime	14:44:17
DigitalCreationDate	2025:05:10
TimeCreated	14:44:17+01:00
DateCreated	2025:05:10

RunTimeSincePowerUp	628789.443516958
SubSecCreateDate	2025:05:10 14:44:17.559+01:00
SubSecDateTimeOriginal	2025:05:10 14:44:17.559+01:00
SubSecModifyDate	2025:05:10 14:44:17.559+01:00
DateTimeCreated	2025:05:10 14:44:17+01:00
DigitalCreationDateTime	2025:05:10 14:44:17

Software & Processing

JFIFVersion	1 1
Software	18.4.1
ExifVersion	0232
MPFVersion	0100
ProfileVersion	1024
ProfileCreator	appl
ApplicationRecordVersion	2

Metadata & Description

Image Description	Costwolds Trip with Amma
SubjectArea	2851 2139 3141 1880
ProfileDescription	Display P3
ProfileCopyright	Copyright Apple Inc., 2022

Technical Details

ComponentsConfiguration	1 2 3 0
MPImageLength	287721
ProfileCMMType	appl
ProfileClass	mntr

ProfileConnectionSpace	XYZ
ProfileFileSignature	acsp
ProfileID	236 253 163 142 56 133 71 195 109 180 189 79 122 218 24 47
ColorComponents	3

Other Data

MIME Type	image/jpeg
ExifByteOrder	MM
HostComputer	iPhone 15 Plus
YCbCrPositioning	1
ShutterSpeedValue	0.000321999998038031
BrightnessValue	9.690370434
AEStable	1
AETarget	194
AEAverage	198
AFStable	1
AccelerationVector	-0.01860005224 -0.7497911453 -0.6758506896
ImageCaptureType	12
LivePhotoVideoIndex	8595185700
PhotosAppFeatureFlags	0
HDRHeadroom	1.00999999
AFPerformance	18 268435507
SignalToNoiseRatio	61.91085814
PhotoIdentifier	7189F0E1-2778-4540-92DF-80200889284C
ColorTemperature	5276

CameraType	1
HDRGain	0.7108429074
SemanticStyle	{_0=1,_1=-0.5,_2=0,_3=3}
SensingMethod	2
SceneType	1
Scene Capture Type	0
CompositeImage	2
NumberOfImages	3
MPImageFlags	0
MPImageType	0
MPImageStart	3590066
DependentImage1EntryNumber	0
DependentImage2EntryNumber	0
PrimaryPlatform	APPL
CMMFlags	0
DeviceManufacturer	APPL
DeviceAttributes	0 0
RenderingIntent	0
ConnectionSpaceIlluminant	0.9642 1 0.82491
MediaWhitePoint	0.96419 1 0.82489
RedMatrixColumn	0.51512 0.2412 -0.00105
GreenMatrixColumn	0.29198 0.69225 0.04189
BlueMatrixColumn	0.1571 0.06657 0.78407

RedTRC	(Binary data 32 bytes, use -b option to extract)
ChromaticAdaptation	1.04788 0.02292 -0.0502 0.02959 0.99048 -0.01706 -0.00923 0.01508 0.75168
BlueTRC	(Binary data 32 bytes, use -b option to extract)
GreenTRC	(Binary data 32 bytes, use -b option to extract)
HDRGainCurveSize	267
HDRGainCurve	(Binary data 2040 bytes, use -b option to extract)
CurrentIPTCDigest	798387bb5597d43bfcb320f898a5f53d
CodedCharacterSet	%G
Caption-Abstract	Costwolds Trip with Amma
IPTCDigest	798387bb5597d43bfcb320f898a5f53d
EncodingProcess	0
YCbCrSubSampling	2 2
Image Size	5712 4284
Megapixels	24.470208
ScaleFactor35efl	4.36241610738255
ShutterSpeed	0.0003219575016
MPImage2	(Binary data 273975 bytes, use -b option to extract)
MPImage3	(Binary data 287721 bytes, use -b option to extract)
CircleOfConfusion	0.00688752743646326
FOV	69.3903656740024
HyperfocalDistance	3.22336284026481
LightValue	13.9569859246564

Appendix 2: JSON Schema of the News Atom (v1.0)

JSON

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "News Atom Schema v1.0",
  "description": "Structured, semantic, sentence-level units of journalism
with comprehensive metadata",
  "type": "object",
  "additionalProperties": false,
  "required": [
    "atom_id",
    "version",
    "atom_status",
    "knowledge_frame",
    "statement",
    "semantic_frame",
    "primary_expression",
    "event_frame",
    "topic_ids",
    "language",
    "review_process",
    "origin",
    "license"
  ],
  "properties": {
    "atom_id": {
      "type": "string",
      "pattern": "^[A-Z]{3}[0-9]{4,}$",
      "description": "Unique identifier: 3-letter organisation code + 4+
digit sequence"
    },
    "version": {
      "type": "string",
      "enum": ["v1.0"],
      "description": "Schema version (semantic versioning)"
    },
    "atom_status": {
      "type": "string",
      "enum": ["active", "superseded", "retracted", "draft"],
      "description": "Current lifecycle status of the atom",
      "default": "active"
    },
```

```
    "supersedes_atom_id": {
      "type": "string",
      "pattern": "^[A-Z]{3}[0-9]{4,}$",
      "description": "ID of the previous atom this one replaces; omit if
original"
    },
    "knowledge_frame": {
      "type": "object",
      "description": "Epistemic and typological classification for this
atom.",
      "required": ["information_type", "knowledge_type", "direct_quote"],
      "additionalProperties": false,
      "properties": {
        "information_type": {
          "type": "string",
          "enum": ["observed_fact", "sensemaking"],
          "description": "Binary epistemic flag to distinguish between what
happened and what it means."
        },
        "knowledge_type": {
          "type": "string",
          "enum": [
            "action",
            "reaction",
            "consequence",
            "context",
            "previous_episode",
            "history",
            "narrative",
            "evaluation",
            "expectation"
          ],
          "description": "Primary journalistic category."
        },
        "subtype": {
          "type": "string",
          "description": "Optional refinement; valid only for certain
knowledge types."
        },
        "source": {
          "oneOf": [
            { "type": "string", "minLength": 1 },
            { "type": "array", "minItems": 1, "items": { "type": "string",
"minLength": 1 } }
          ],
          "description": "Full in-sentence attribution phrase(s) as printed
- captures both substantive and reporting attribution."
        },
```

```
        "direct_quote": {
          "type": "boolean",
          "description": "True if the sentence contains a direct quotation."
        }
      },
      "allOf": [
        { "if": { "properties": { "knowledge_type": { "const": "reaction" }
} },
          "then": {
            "required": ["subtype"],
            "properties": { "subtype": { "enum": ["claim", "allegation",
"position_statement", "denial", "appeal"] } }
          }
        },
        { "if": { "properties": { "knowledge_type": { "const": "consequence"
} } },
          "then": {
            "required": ["subtype"],
            "properties": { "subtype": { "enum": ["trend",
"statistical_outcome", "immediate_outcome"] } }
          }
        },
        { "if": { "properties": { "knowledge_type": { "const": "context" } }
},
          "then": {
            "required": ["subtype"],
            "properties": { "subtype": { "enum": ["analysis", "definition",
"comparison", "methodology"] } }
          }
        },
        { "if": { "properties": { "knowledge_type": { "const": "evaluation"
} } },
          "then": {
            "required": ["subtype"],
            "properties": { "subtype": { "enum": ["proposal",
"risk_assessment", "responsibility"] } }
          }
        },
        { "if": { "properties": { "knowledge_type": { "const": "expectation"
} } },
          "then": {
            "required": ["subtype"],
            "properties": { "subtype": { "enum": ["forecast", "prediction",
"schedule", "scenario", "speculation"] } }
          }
        },
        { "if": { "properties": { "knowledge_type": { "enum": ["action",
"previous_episode", "history", "narrative"] } } },
```

```
          "then": { "not": { "required": ["subtype"] } }
        },
        { "if": { "properties": { "direct_quote": { "const": true } } },
          "then": { "required": ["source"] }
        }
      ]
    },
    "statement": {
      "type": "object",
      "description": "Structured representation of the sentence's
grammatical and semantic content",
      "required": ["subject", "predicate", "object", "original_text"],
      "additionalProperties": false,
      "properties": {
        "subject": {
          "oneOf": [
            { "type": "string" },
            { "type": "array", "items": { "type": "string" } }
          ],
          "description": "Who or what is performing the action"
        },
        "predicate": {
          "oneOf": [
            { "type": "string" },
            { "type": "array", "items": { "type": "string" } }
          ],
          "description": "The action, state or relationship being described"
        },
        "object": {
          "oneOf": [
            { "type": "string" },
            { "type": "array", "items": { "type": "string" } }
          ],
          "description": "What the action is being performed on or toward"
        },
        "date": {
          "type": "string",
          "format": "date",
          "description": "When the action occurred (only if specified in the
sentence)"
        },
        "location": {
          "type": "string",
          "description": "Where the action occurred (only if specified in
the sentence)"
        },
        "original_text": {
          "type": "string",
```

```
          "description": "The exact sentence as it appears in the source"
        }
      }
    },
    "semantic_frame": {
      "type": "object",
      "description": "Links to structured knowledge and flexible event
frameworks",
      "required": ["entities", "semantic_grounding"],
      "additionalProperties": false,
      "properties": {
        "entities": {
          "type": "array",
          "minItems": 1,
          "items": {
            "type": "object",
            "required": ["name", "type"],
            "additionalProperties": false,
            "properties": {
              "name": { "type": "string" },
              "type": {
                "type": "string",
                "enum": ["person", "organization", "location", "event",
"concept", "date", "quantity"]
              },
              "wikidata_id": {
                "type": "string",
                "pattern": "^Q[0-9]+$",
                "description": "Optional Wikidata identifier (Q-number)"
              },
              "geonames_id": {
                "type": "string",
                "pattern": "^[0-9]+$",
                "description": "Optional GeoNames identifier for locations"
              }
            }
          }
        },
        "semantic_grounding": {
          "type": "array",
          "minItems": 1,
          "items": {
            "type": "object",
            "required": ["frame_type", "frame_name", "roles"],
            "additionalProperties": false,
            "properties": {
              "frame_type": {
                "type": "string",
```

```
              "enum": ["framenet", "custom"]
            },
            "frame_name": { "type": "string" },
            "roles": {
              "type": "object",
              "additionalProperties": { "type": "string" }
            }
          }
        }
      }
    },
    "primary_expression": {
      "type": "object",
      "description": "How journalism was originally conceived and
structured, based on The Directory of Liquid Content taxonomy",
      "required": ["content_type", "content_format", "title"],
      "additionalProperties": false,
      "properties": {
        "content_type": {
          "type": "string",
          "pattern": "^CT[0-9]+$",
          "description": "Directory of Liquid Content code for content type
(e.g., CT1, CT2)"
        },
        "content_format": {
          "type": "string",
          "pattern": "^CF[0-9]+$",
          "description": "Directory of Liquid Content code for content
format (e.g., CF1, CF2)"
        },
        "title": {
          "type": "string",
          "description": "The headline or title of the original work"
        }
      }
    },
    "media_anchor": {
      "type": "object",
      "description": "Technical bridge from multimedia to text atomization
(only for non-text sources)",
      "required": ["modality", "file_url", "transcript_text",
"timestamp_start", "timestamp_end"],
      "additionalProperties": false,
      "properties": {
        "modality": {
          "type": "string",
          "enum": ["audio", "video"],
```

```
          "description": "Whether source is audio or video"
        },
        "file_url": {
          "type": "string",
          "format": "uri",
          "description": "Direct link to the multimedia file (can be from
YouTube, Spotify, or any repository)"
        },
        "transcript_text": {
          "type": "string",
          "description": "The transcribed text that became this atom"
        },
        "timestamp_start": {
          "type": "string",
          "pattern": "^\\d{2}:\\d{2}:\\d{2}\\.\\d{3}$",
          "description": "When this sentence begins in the audio/video"
        },
        "timestamp_end": {
          "type": "string",
          "pattern": "^\\d{2}:\\d{2}:\\d{2}\\.\\d{3}$",
          "description": "When this sentence ends in the audio/video"
        }
      }
    },
    "event_frame": {
      "type": "array",
      "minItems": 1,
      "description": "Canonical event(s) this atom refers to.",
      "items": {
        "type": "object",
        "required": ["event_id", "event_label"],
        "properties": {
          "event_id": {
            "type": "string",
            "pattern": "^[A-Z]{3}[0-9]{4,}$",
            "description": "Event ID: org-local, opaque, stable (e.g.,
SCR1001)."
          },
          "event_label": {
            "type": "string",
            "description": "Human-readable event name: [DATE]
[PRIMARY_ACTOR] [ACTION_CODE] [OBJECT] [@LOCATION]."
          }
        }
      }
    },
    "topic_ids": {
      "type": "array",
```

```
      "minItems": 1,
      "items": {
        "type": "string",
        "pattern": "^medtop:[0-9]{8}$"
      },
      "description": "IPTC MediaTopic codes for thematic classification"
    },
    "language": {
      "type": "string",
      "pattern": "^[a-z]{2}$",
      "minLength": 2,
      "maxLength": 2,
      "description": "Two-letter ISO 639-1 language code",
      "examples": ["en", "hi", "es", "fr", "de", "zh", "ar"]
    },
    "review_process": {
      "type": "object",
      "required": ["automated_annotation", "human_review"],
      "additionalProperties": false,
      "properties": {
        "automated_annotation": {
          "type": "object",
          "required": ["annotated_by", "timestamp"],
          "properties": {
            "annotated_by": {
              "type": "string",
              "description": "The LLM model that performed the initial
annotation"
            },
            "timestamp": {
              "type": "string",
              "format": "date-time",
              "description": "When the automated annotation was completed"
            }
          }
        },
        "human_review": {
          "type": "object",
          "required": ["status"],
          "properties": {
            "status": {
              "type": "string",
              "enum": ["reviewed", "pending", "not_required"],
              "description": "Current review status"
            },
            "reviewer_id": {
              "type": "string",
              "description": "Identifier of the human reviewer"
```

```
            },
            "changes_made": {
              "type": "array",
              "items": {
                "type": "string",
                "enum": [
                  "corrected_knowledge_frame",
                  "corrected_statement",
                  "corrected_semantic_frame",
                  "updated_knowledge_frame",
                  "updated_statement",
                  "updated_semantic_frame",
                  "no_changes_needed"
                ],
                "description": "List of corrections or updates made during
review"
              }
            },
            "timestamp": {
              "type": "string",
              "format": "date-time",
              "description": "When the human review was completed"
            }
          }
        }
      },
      "allOf": [
        {
          "if": { "properties": { "human_review": { "properties": {
"status": { "const": "reviewed" } } } } },
          "then": { "properties": { "human_review": { "required":
["reviewer_id", "timestamp"] } } } }
      ]
    },
    "origin": {
      "type": "object",
      "description": "Publication metadata establishing accountability and
attribution",
      "required": ["organization", "journalist", "url", "created_at"],
      "additionalProperties": false,
      "properties": {
        "organization": {
          "type": "string",
          "description": "Publishing organization name"
        },
        "journalist": {
          "type": "string",
          "description": "Author or reporter byline"
```

```
        },
        "url": {
          "type": "string",
          "format": "uri",
          "description": "Canonical URL of the source article"
        },
        "created_at": {
          "type": "string",
          "format": "date-time",
          "description": "Publication timestamp of the original article"
        },
        "source_article_id": {
          "type": "string",
          "description": "Stable identifier of the article in the
publisher's CMS"
        }
      }
    },
    "license": {
      "type": "object",
      "required": ["type", "terms_url"],
      "additionalProperties": false,
      "properties": {
        "type": {
          "type": "string",
          "enum": ["all_rights_reserved", "cc_by", "cc_by_nc",
"syndicated_feed"],
          "description": "Standardized license type for this content"
        },
        "terms_url": {
          "type": "string",
          "format": "uri",
          "description": "URL to complete licensing terms and conditions"
        }
      }
    }
  }
}
```