Workshop DHd25:

Wissensgraphen und große Sprachmodelle in den Digital Humanities

Julian Stalter[1], Matthias Springstein[2,3], Maximilian Kristen[1], Eric Müller-Budack[2], Stefanie Schneider[1], Elias Entrup[2,3], Hubertus Kohle[1], Ralf Krestel[4], und Ralph Ewerth[2,3]

[1]LMU München
[2]TIB – Leibniz-Informationszentrum Technik und Naturwissenschaften, Hannover
[3]Forschungszentrum L3S, Leibniz UNiversität Hannover
[4]ZBW–Leibniz-Informationszentrum für Wirtschaftswissenschaften, Kiel

Gefördert durch
DFG Deutsche Forschungsgemeinschaft

https://tibhannover.github.io/ReflectAI-DHd2025/

Montag

14:00 – 14:30: Begrüßung der Teilnehmenden und Einführung in ReflectAI und iART

14:30 – 15:00: “Einführung in Wissensgraphen” (Ralf Krestel)

15:00 – 15:30: Pause

15:30 – 16:00: Vorstellung der Annotationsrichtlinien von ReflectAI und Beispiele für manuelle Triplet-Extraktion

16:00 – 17:00: Erstellen von Ontologien und extrahieren von Triples aus Texten - manuell oder mithilfe eines unterstützenden Tools wie Inception

17:00 – 17:30: Einrichten der gemeinsamen Arbeitsumgebung in Google Colab

Dienstag

14:00 – 14:30: Einführung in verschiedene Methoden zur automatisierten Triplet-Extraktion mit großen Sprachmodellen, u. a. Gollie und DeepKE

14:30 – 15:30: Testen der automatisierten Extraktion mithilfe der vorgestellten Methoden mit Beispieltexten

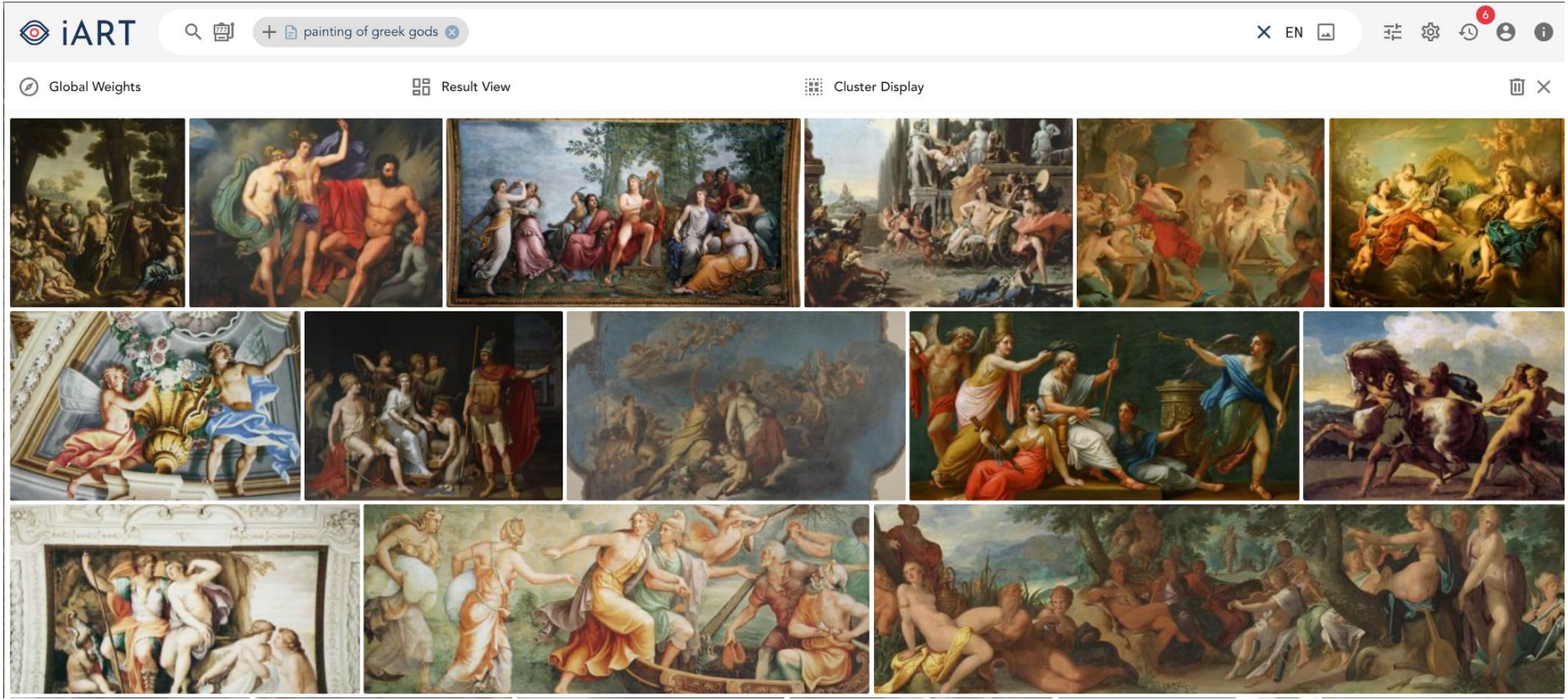
15:30 – 15:45: Pause

15:45 – 16:30: Vergleichen der Ergebnisse der manuellen und automatisierten Extraktion

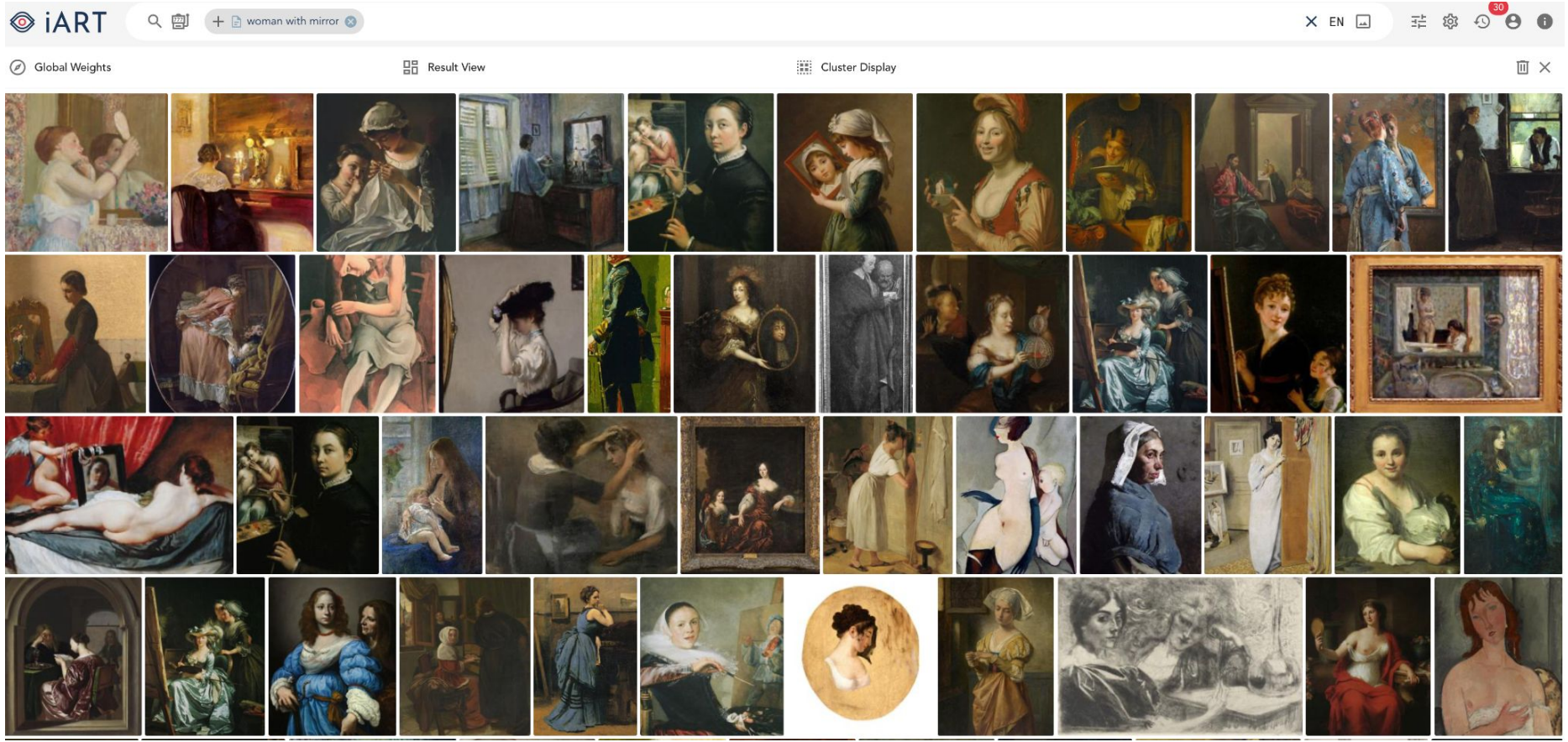
16:30 – 17:00: Abgleichen der extrahierten Entitäten mit Wikidata und kurze Vorstellung der Abfragesprache SPARQL, um gezielt Informationen aus Wissensgraphen zu extrahieren

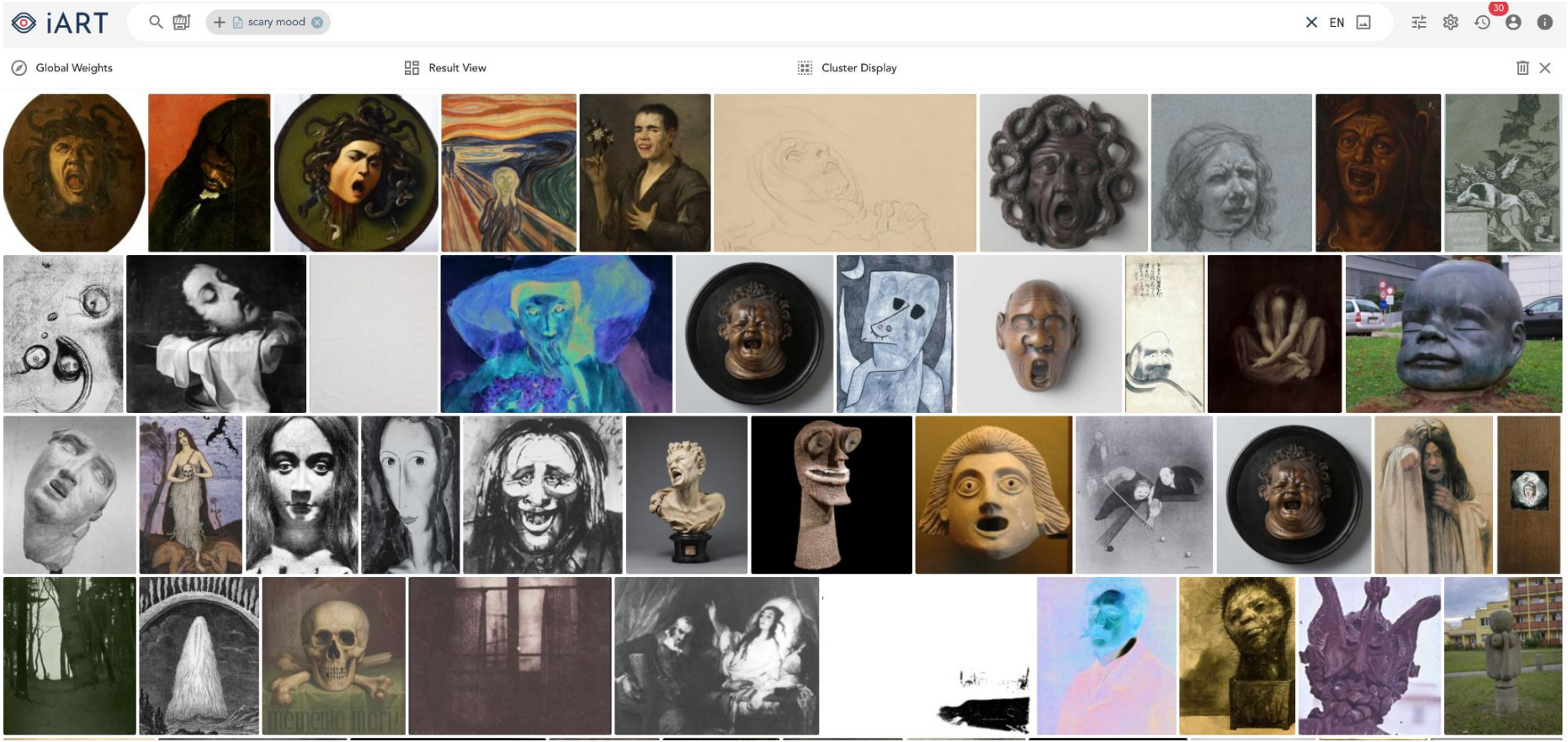
17:00 – 17:30: Zusammenfassung der wichtigsten Ergebnisse und Gelegenheit für Feedback und offene Fragen

Einführung iART + ReflectAI

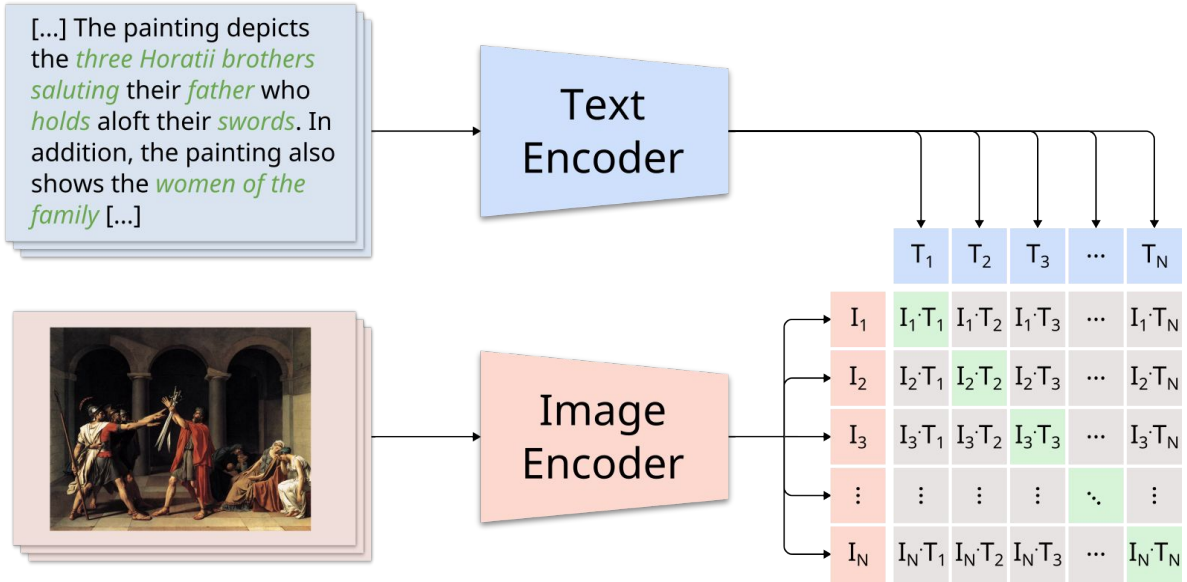


www.iart.vision


iART
woman with mirror
EN
30
Global Weights
Result View
Cluster Display

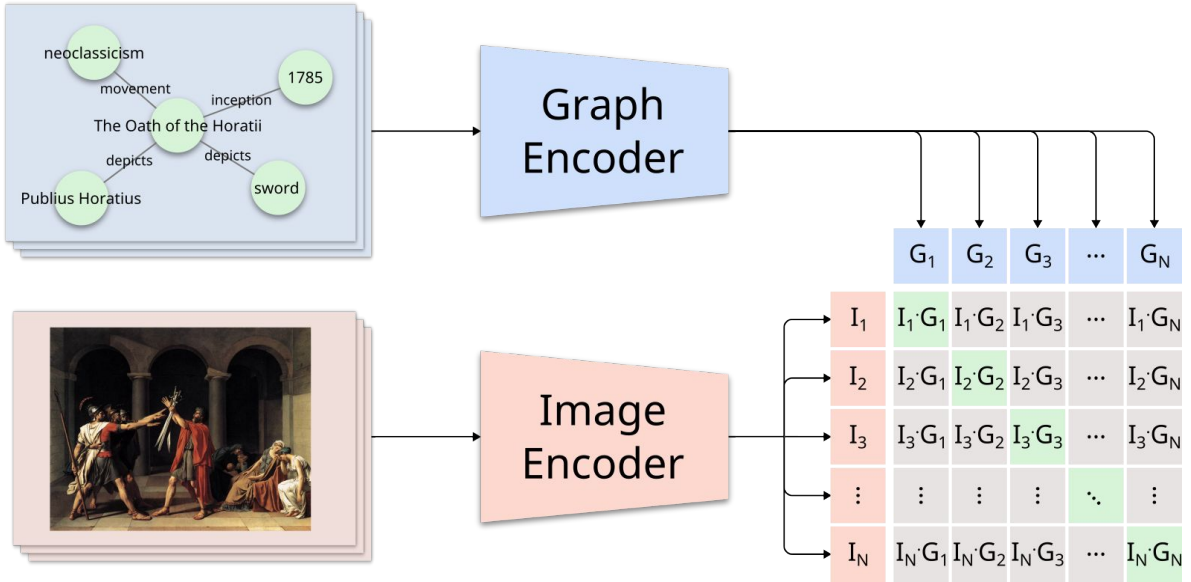

iART
scary mood
EN
30
Global Weights
Result View
Cluster Display

Multimodale Modelle (bspw. CLIP)

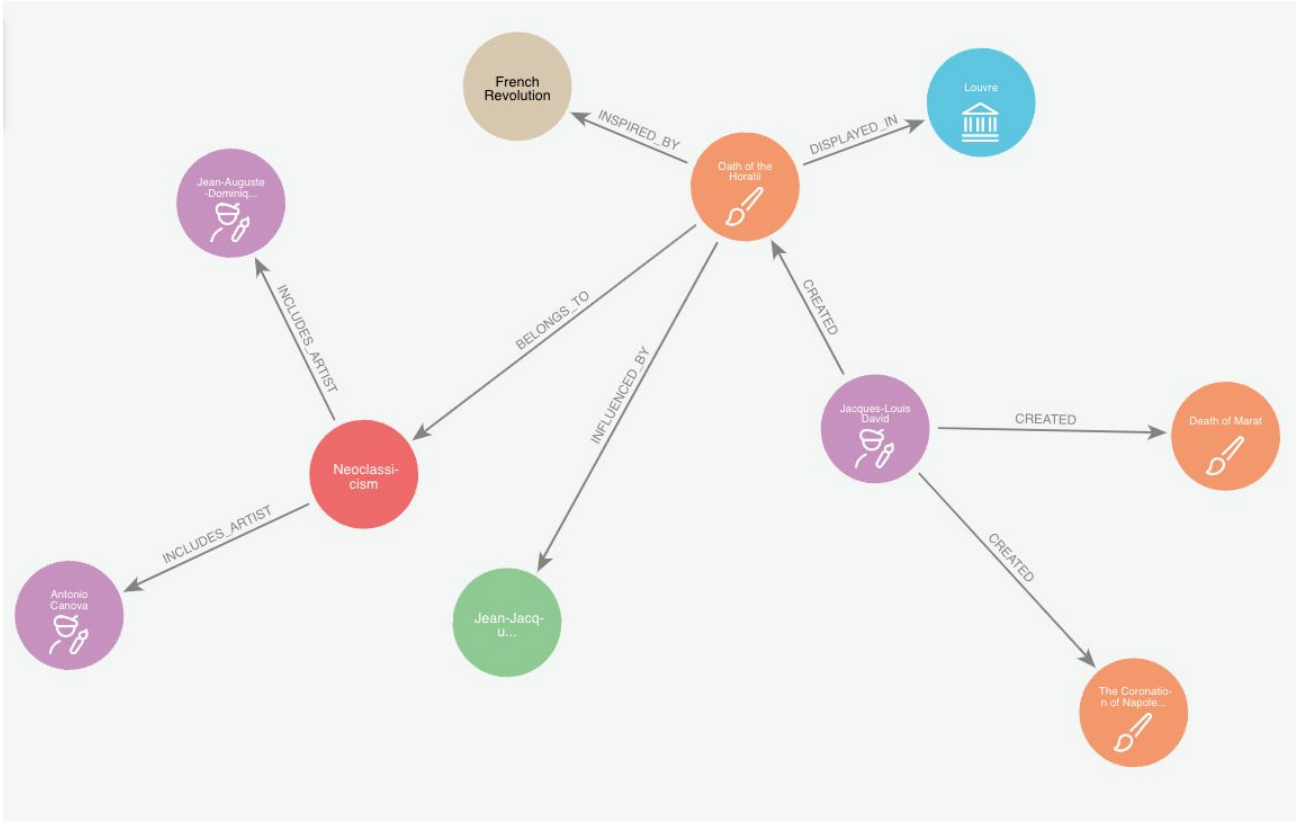


Maschinelles Lernen und Wissensgraphen

Wissensgraphen kombiniert mit neuronalen Netzen

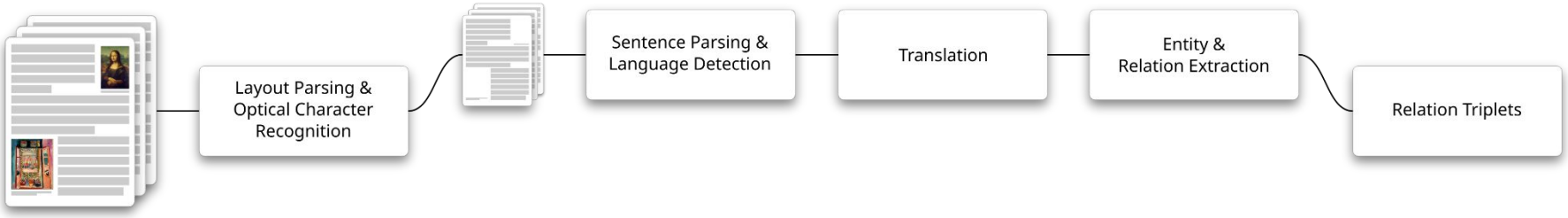


“Jacques Louis David, a key figure in Neoclassicism, created the painting *Oath of the Horatii*. This artwork, influenced by the ideas of Jean-Jacques Rousseau, reflects the principles of the French Revolution. *Oath of the Horatii* belongs to the Neoclassical movement and is currently displayed in the Louvre. In addition to this painting, David also created *Death of Marat* and *The Coronation of Napoleon*.”

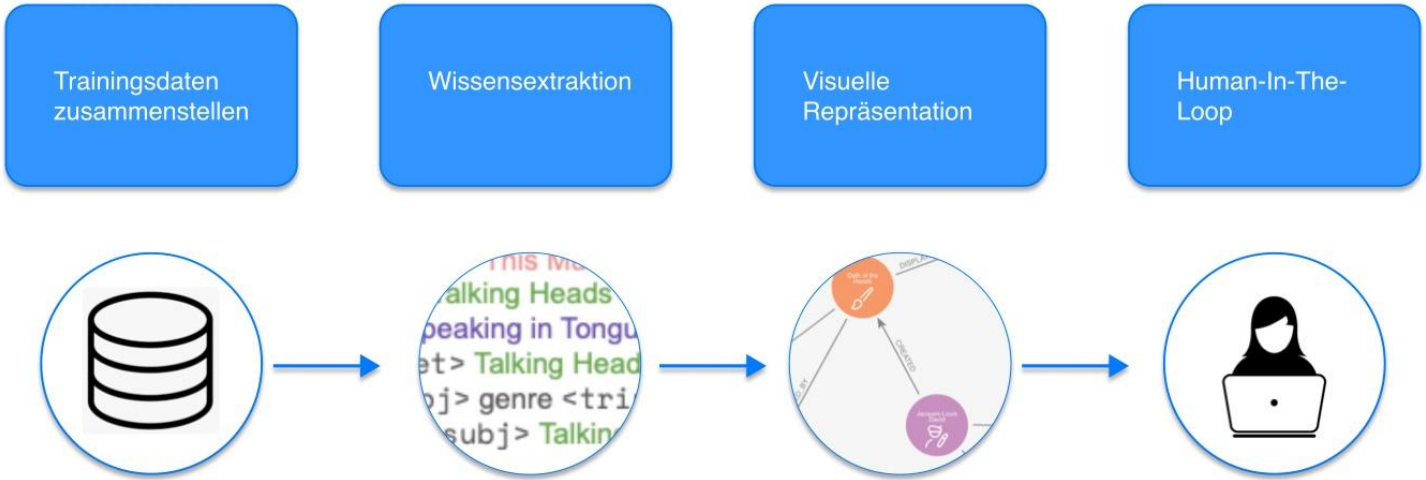


Methoden: Relationsextraktion (Klassischer Ansatz)

- Erkennen von Text- und Bildbereichen in einzelnen Dokumenten
- Zerlegung des Textes in einzelne Sätze
- Erkennung der Sprache der einzelnen Abschnitte
- Automatische Übersetzung der einzelnen Abschnitte ins Englische
- Extraktion von Entitäten und Relationen -> Überführung in Triplets



Erstellung der Wissensgraphen aus kunsthistorischen Texten



LLM

- Language model is used in isolation
- Lacks factual knowledge

Retrieval Augmented Generation

- Retrieves unstructured information to augment answers

Knowledge Augmented Generation

- Access to structured information sources, such as databases, knowledge graphs

Agentic Generation

- Use of external tools such as reasoners, APIs, computation services

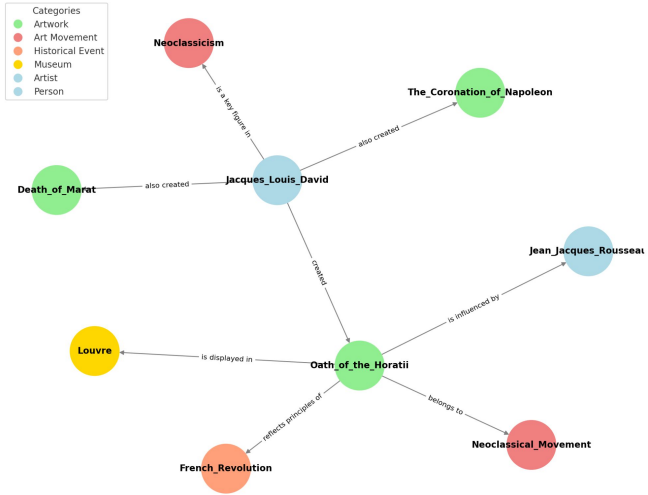
Artificial General Intelligence

- Self-reflection, goal setting

Vortrag Prof. Ralf Krestel

Manuelles Annotieren / Tagset ReflectAI

- (Jacques Louis David [**Artist**]) → (is a key figure in) → (Neoclassicism [**Art Movement**])
- (Jacques Louis David [**Artist**]) → (created) → (Oath of the Horatii [**Artwork**])
- (Oath of the Horatii [**Artwork**]) → (is influenced by) → (Jean-Jacques Rousseau [**Person**])
- (Oath of the Horatii [**Artwork**]) → (reflects the principles of) → (French Revolution **[Historical Event**])
- (Oath of the Horatii [**Artwork**]) → (belongs to) → (Neoclassical movement [**Art Movement**])
- (Oath of the Horatii [**Artwork**]) → (is displayed in) → (Louvre [**Museum**])
- (Jacques Louis David [**Artist**]) → (also created) → (Death of Marat [**Artwork**])
- (Jacques Louis David [**Artist**]) → (also created) → (The Coronation of Napoleon [**Artwork**])



Stefanie Schneider, Julian Stalter, Miriam Göldl:

Annotation Guidelines for the Project *Reflect AI*

The project *ReflectAI* seeks to investigate both practical and theoretical approaches to employing image similarity assessments in art history. As part of this effort, art-historical texts—i.e., from essays, books, and catalogs—are first transformed into domain-specific knowledge graphs, which are then used to train hybrid *Artificial Intelligence* (AI) models. This document outlines the guidelines applied in the process of annotating these resources, including annotator instructions. It also provides illustrative examples from the annotated texts, specifying expressions that should and should not be marked.

https://guideline.open-develop.org/

Work of Art

Description

aesthetic item or artistic creation

Rules

We annotate titles or specific names of artworks (e.g., “Mona Lisa” or “The Sistine Chapel”) in both the *metadata layer* and the *content layer*. General references to instances of artworks (e.g., “painting”) are excluded; instead, type of work of art should be used. We do not tag references to art genres or movements, which are tagged separately.

Examples

One example is Albrecht Dürer's *Melencolia I* (1514) in the Metropolitan Museum of Art:

Work of Art Melencolia I is a depiction of the intellectual situation of the artist and is thus, by extension, a spiritual Type of Work of Art self-portrait of Person Dürer .

The tag always has to be annotated in both the *metadata layer* and the *content layer*:

Work of Art Melencolia I is a depiction of the intellectual situation of the artist and is thus, by extension, a spiritual Type of Work of Art self-portrait of Person Dürer .

See

W Q838948

dhd25_Zahl

dhd_2025

Religious Character

Description

character of a religious work, alleged to be historical

Rules

We annotate names of religious figures (e.g., “Jesus” or “Adam”) when they are visually represented or symbolically referenced in a work of art. Generalized terms for well-known religious figures (e.g., the “Apostles”) should be tagged where their identity is clear; however, generic references (e.g., to “saints”) should not be tagged unless they refer to a specific figure within a particular religious tradition. Non-religious characters should be tagged as person unless they are depicted with religious attributes.

Examples

The description of Boticelli's *Lamentation over the Dead Christ* (c. 1490/95) provides an example of marking a disciple and Jesus Christ:

Religious Character St John supports her and prevents the Quality radiant Anatomical Structure body of Religious Character Christ from slipping from her Anatomical Structure lap .

Relations

depicts (work of art → religious character)

See

W Q18563354

Posture

Description

physical configuration that a human can take

Rules

We annotate terms that describe body positions or postures of figures in the work of art (e.g., “sitting” or “standing”). Vague descriptions of body positions should not be tagged unless they describe a recognizable posture (e.g., “moving” is too general). Terms related to specific body parts (e.g., “head tilted”) are tagged even if they do not refer to the overall posture. However, abstract or symbolic descriptions not related to physical expressions (e.g., “poised for action”) are excluded.

Examples

John Singer Sargent's reclining woman in *Nonchaloir (Repose)* (1911) is described by several references to her posture:

> The Posture reclining woman, Quality casually Posture posed in an atmosphere of Quality elegiac Emotion calm and consummate luxury, seems the epitome of nonchalance—the Type of Work of Art painting's original title.

Descriptions of a particular movement resulting in a posture are also annotated; see, for example, the following description of Sandra Botticelli's *Lamentation of Christ* (c. 1490/1495):

> Religious Character Mary is Emotion overwhelmed by her Emotion sorrow and Posture faints to the side .
> Religious Character John Posture supports her and prevents the Quality radiant Anatomical Structure body of Religious Character Christ from slipping off her Anatomical Structure lap .

Relations

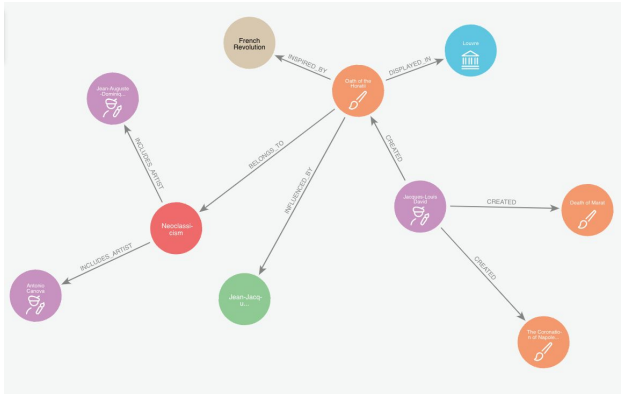
has characteristic (person → posture)
if no person is given, *depicts* (work of art → posture)

See

𝕎 Q8514257

Arbeitsaufgabe: Entwickle eine Ontologie mit Entitäten und Relationen

1. **Identifiziere die zentralen Begriffe**
   a. Extrahiere die wichtigsten Entitäten aus Deinem Text (z. B. Konzepte, Objekte, Akteure).
2. **Kategorisiere die Entitäten**
   a. Ordne die Entitäten in übergeordnete Klassen und Unterkategorien ein. (z.B. Künstler, Emotion, Genre)
3. **Definiere Relationen zwischen den Entitäten**
   a. Beschreibe, wie die Entitäten miteinander in Verbindung stehen (z. B. „ist Teil von“, „beeinflusst“).
4. **Verknüpfe die Entitäten mit Wikidata QIDs**
   a. Recherchiere und verknüpfe passende Wikidata QIDs für relevante Entitäten.
5. **Strukturiere die Ontologie als Diagramm oder Modell**
   a. Erstelle eine visuelle Darstellung der Ontologie (z. B. als Graph oder Tabelle).



Arbeitsaufgabe: Annotation in INCEpTION

- 1. **Text auswählen**
  - Bestimme einen geeigneten Text als Grundlage für die Ontologie.
- 2. **Mit der Software vertraut machen**
  - Arbeite Dich in **INCEpTION** ein.
- 3. **Entitäten und Relationen in INCEpTION extrahieren**
  - Markiere zentrale Entitäten und definiere Relationen direkt in **INCEpTION**.
- 4. **Struktur der Ontologie optimieren**
  - Kategorisiere die Entitäten und passe die Relationen an, um eine konsistente Ontologie zu entwickeln.
- 5. **Wikidata QIDs zuweisen**
  - Ergänze relevante Wikidata-QIDs für die extrahierten Entitäten und prüfe, ob sie bereits in Wikidata existieren.

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Forschungszentrum L3S Research Center

Leibniz Universität Hannover

Workshop DHd25:

Wissensgraphen und große Sprachmodelle in den Digital Humanities

Stalter, Julian; Springstein, Matthias; Kristen, Maximilian; Müller-Budack, Eric; Schneider, Stefanie; Entrup, Elias; Kohle, Hubertus; Krestel, Ralf; Ewerth, Ralph

Gefördert durch DFG Deutsche Forschungsgemeinschaft

https://tibhannover.github.io/ReflectAI-DHd2025/

iART: Projekt

Förderungszeitraum: **2019-2022**

Gefördert durch: **Deutsche Forschungsgemeinschaft (DFG)**

Projektziel: **Entwicklung einer Suchplattform für die inhaltsbasierte Suche in kunsthistorischen Bildbeständen**

Projektpartner:

- **Prof. Dr. Ralph Ewerth**
  Technische Informationsbibliothek (TIB)
- **Prof. Dr. Hubertus Kohle**
  Ludwig-Maximilians-Universität München
- **Prof. Dr. Eyke Hüllermeier**
  Universität Paderborn

TIB

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

ReflectAI

Förderungszeitraum: **2023-2026**

Gefördert durch: **Deutsche Forschungsgemeinschaft (DFG)**

Projektziel: **Verbesserung der Erklärbarkeit von KI-Ergebnissen in der Kunstgeschichte durch Wissensgraphen**

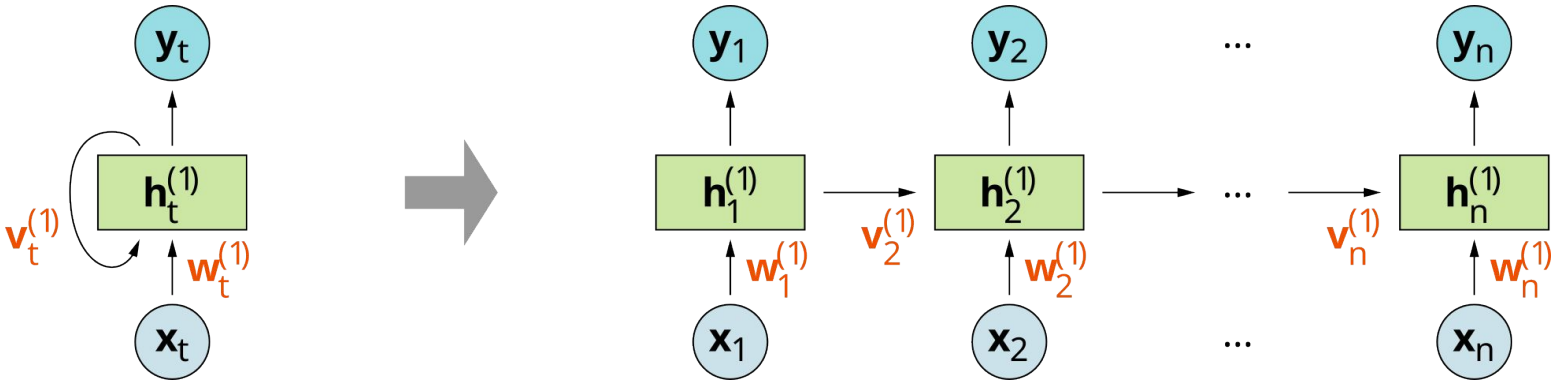
Projektpartner:

- **Prof. Dr. Ralph Ewerth**
  Technische Informationsbibliothek (TIB)
- **Prof. Dr. Hubertus Kohle**
  Ludwig-Maximilians-Universität München

Einführung in verschiedene Methoden zur automatisierten Triplet-Extraktion mit großen Sprachmodellen

Recurrent Neural Networks (RNN)

- RNNs are designed to process sequential data (text, audio, etc.)
- RNNs have an internal memory h_t
- Variants:
  - Long Short-Term Memory (LSTM)
  - Gated Recurrent Units (GRUs)
- The recurrent structure of an RNN can be seen as “unrolling” in time
- RNNs struggle to capture long-range dependencies effectively



Transformers (Vaswani et al. 2017)

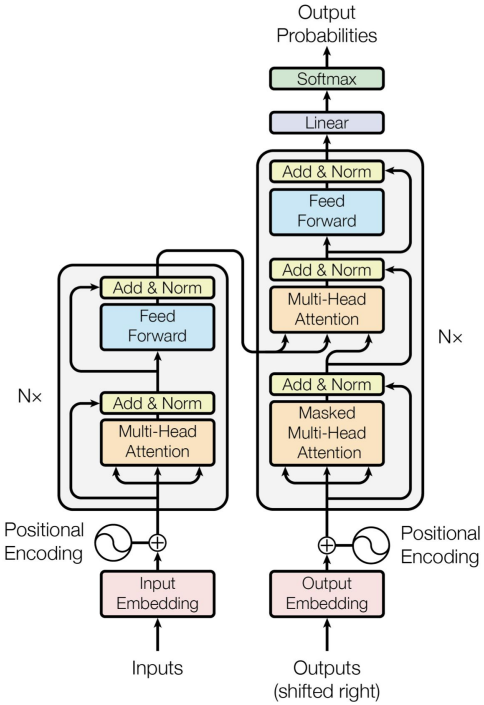
- Model for sequence processing without recurrent connections
- Self-attention models relations of words over long distances

Architecture:

- Encoder (Left) - Decoder (Right)
- Individual blocks consisting of:
  - Positional embeddings (Modeling word order)
  - Multiple attention blocks (Multi-Head Attention)
  - Simple neural network that looks at a single point in time

Examples:

- BART
- T5, Flan-T5



Transformers (Bidirectional Transformers / Encoder)

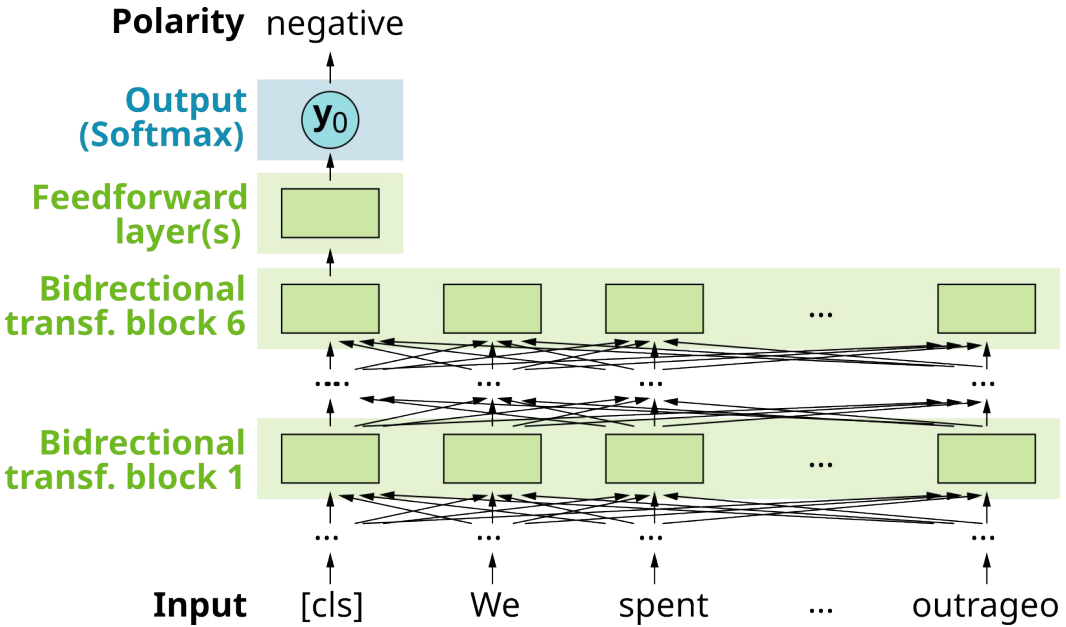
- Attention can look at all inputs
- Allow solving tasks that require information from all inputs

Use Case:

- Sentiment analysis
- Part-of-speech tagging

Examples:

- BERT
- XLNet



Transformers (Bidirectional Transformers / Encoder)

https://colab.research.google.com/github/TIBHannover/ReflectAI-DHd2025/blob/main/notebooks/bert.ipynb

Transformers (Left-to-Right / Decoder)

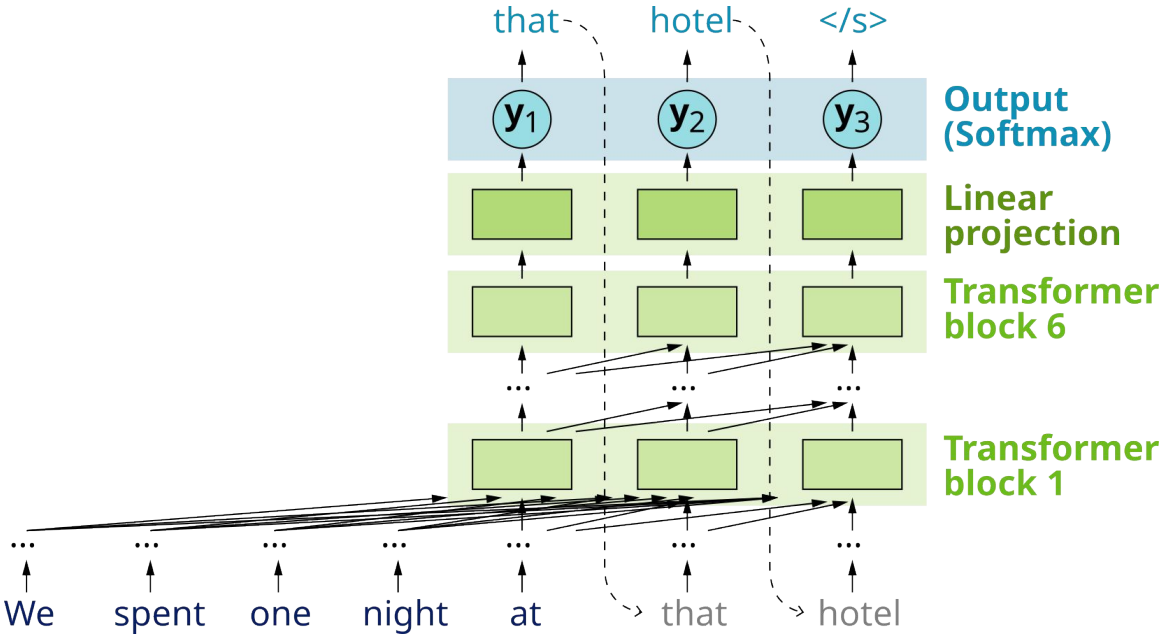
- Attention can only look at previous inputs
- This enables the transformers to do autoregressive generation

Use Case:

- Large Language Model
- Information extraction

Examples:

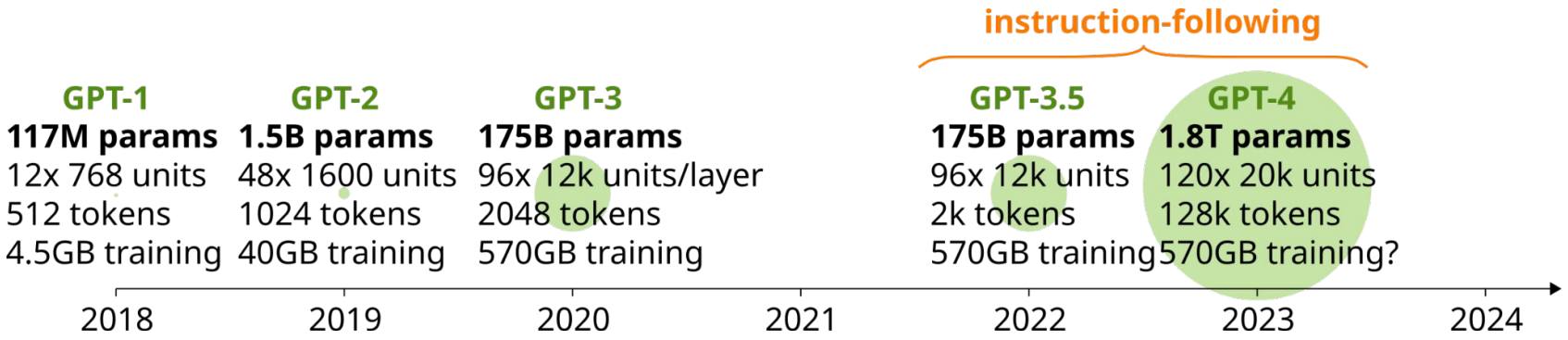
- GPT-X
- LLaMA
- Gemma
- Mixtral



From Transformers to Large Language Models

- Primarily use decoder only architectures
- Scaling architecture parameter 0.2B -> 8B, …
- Scale the size of the training dataset
- Change the training strategy:
  - Pretraining (self-supervised tasks)
  - Instruction fine-tuning (Enables zero-shot usage)

	8B	70B	405B
Layers	32	80	126
Model Dimension	4,096	8192	16,384
FFN Dimension	14,336	28,672	53,248
Attention Heads	32	64	128
Key/Value Heads	8	8	8
Peak Learning Rate	3×10^{-4}	1.5×10^{-4}	8×10^{-5}
Activation Function	SwiGLU		
Vocabulary Size	128,000		
Positional Embeddings	RoPE ($\theta = 500,000$)		



Large Language Models

https://colab.research.google.com/github/TIBHannover/ReflectAI-DHd2025/blob/main/notebooks/gpt2.ipynb

Use of an Instruction-tuned Large Language Model

- In-context Learning

  3 + 2 = ? Answer: 5
  5 + 3 = ? Answer:

- Prompting:
  - Zero-shot Learning
  - Few-shot Learning
  - Chain-of-thought

  Let’s break this down step by step: What is 27 ÷ 3 × 2?

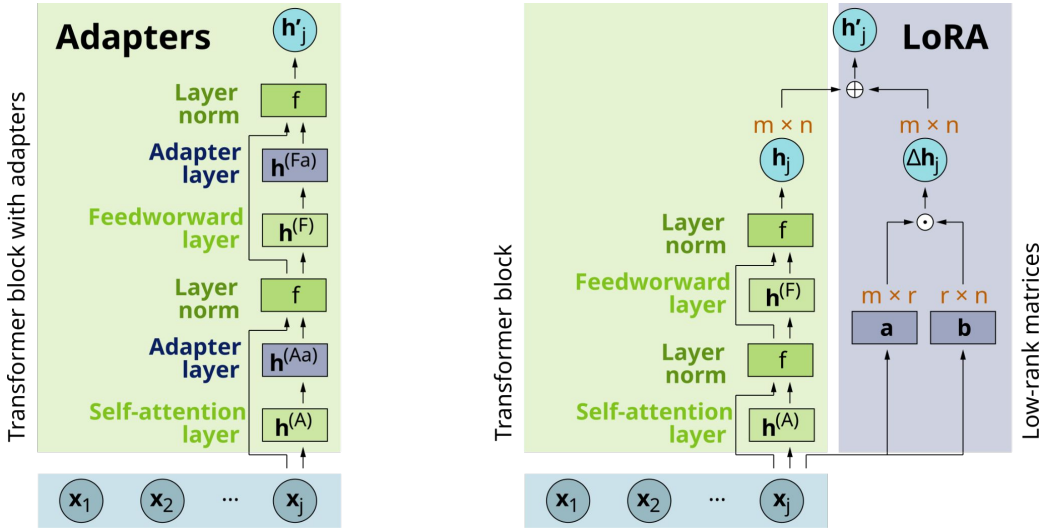
- Prompt Learning

Instruction Tuned Large Language Models

https://colab.research.google.com/github/TIBHannover/ReflectAI-DHd2025/blob/main/notebooks/inst.ipynb

Large Language Models (Fine-tuning)

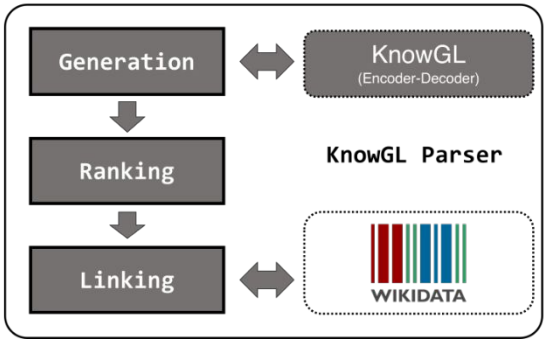
- Types of LLM fine-tuning:
  - Full fine-tuning. Adjust all transformer weights (may take long)
  - Adapters. Train extra layers added to each transformer block.
  - Low-Rank Adaptation (LoRA). Train two extra low-rank matrices in each layer, whose product is added to the original weights.



KnowGL (Rossiello et al. 2023)

- Use a fine-tuned BART model for prediction
- Automatically triplet extraction
- Map detected relations against Wikidata

*For the **semantic web** to function, computers must have access to structured collections of information and sets of **inference rules.***




```
[{
  "subject": {
    "mention": "semantic web",
    "entity_label": "Semantic Web",
    "type_label": "academic discipline",
    "entity_link": "Q54837",
    "type_link": "Q11862829"
  },
  "relation": {
    "label": "uses",
    "link": "Property:P2283"
  },
  "object": {
    "mention": "inference rules",
    "entity_label": "inference",
    "type_label": "process",
    "entity_link": "Q408386",
    "type_link": "Q619671"
  },
  "score": -0.98
}]
```

KnowGL (Rossiello et al. 2023)

https://colab.research.google.com/github/TIBHannover/ReflectAI-DHd2025/blob/main/notebooks/knowgl.ipynb

Instructor (https://github.com/instructor-ai/instructor)

https://colab.research.google.com/github/TIBHannover/ReflectAI-DHd2025/blob/main/notebooks/instructor.ipynb

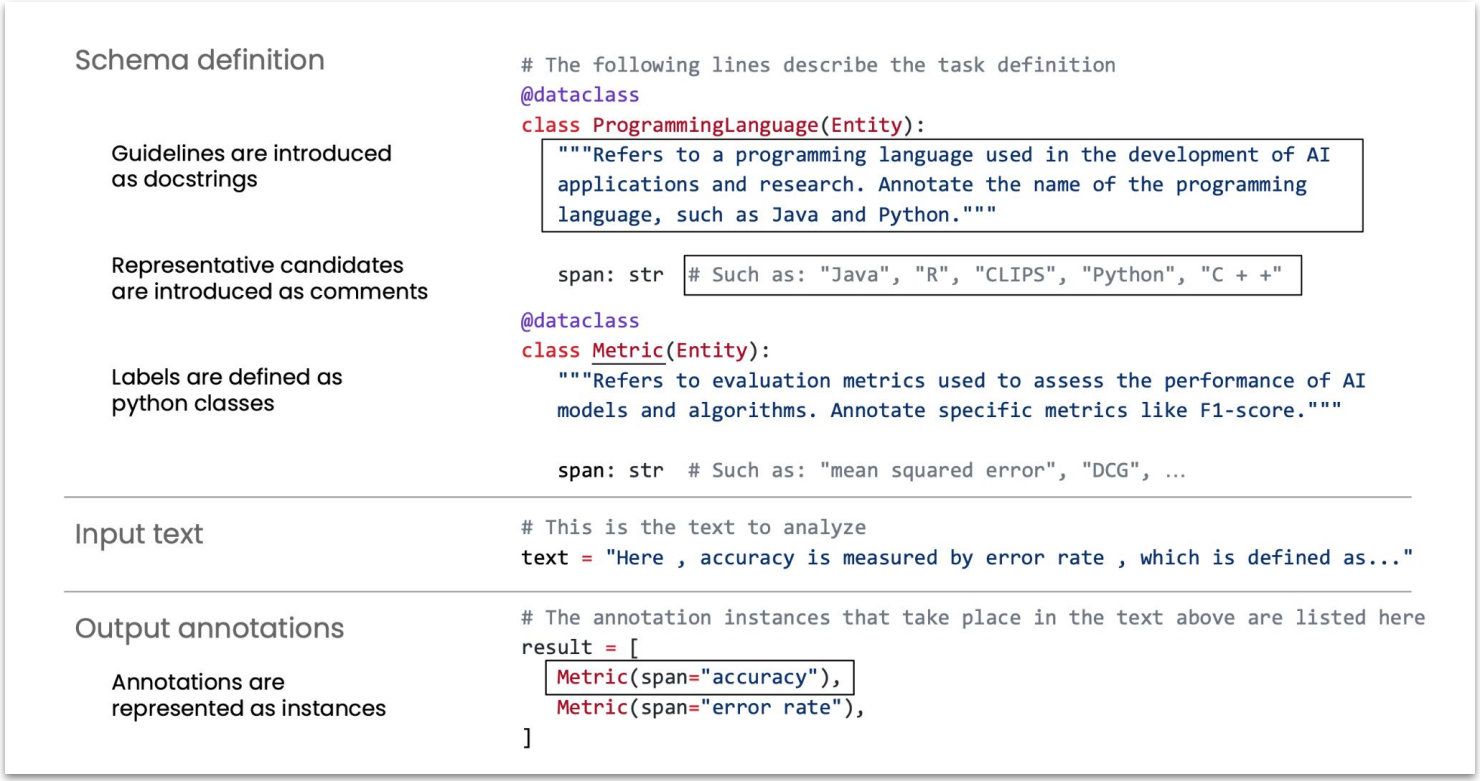
DeepKE

https://colab.research.google.com/github/TIBHannover/ReflectAI-DHd2025/blob/main/notebooks/instructor.ipynb

GoLLIE (Sainz et al. 2024)

- Usage of an LLM for information extraction
- Code-LLaMA (Rozière et al. 2023) with QLoRA (Hu et al. 2022; Dettmers et al. 2023)
- Using Source Code definition as a guideline for preventing hallucination
- Adaptation to new tasks:
    - Providing a new class definition with parameters name and type
    - Comments for task description or extraction examples

GoLLIE (Sainz et al. 2024)



```python
@dataclass
class Attributes(Template):
    """Objects depicted in the painting that have symbolic value like a key or a tool used for certain professions"""

    painting: str  # The name of the painting, i.e. The Last Supper
    objects: List[str]  # The objects depicted in the painting, i.e. key, sword, scale


@dataclass
class ReligiousFigures(Template):
    """Religious Figures depicted in the painting which are part of the content. These religious figures could depict scenes from the Bible or other religious sources."""

    painting: str  # The name of the painting, i.e. The Last Supper
    figure: List[str]  # The figures depicted in the painting, i.e. Jesus, Mary Magdalene, Angel etc.


@dataclass
class MythologicalFigures(Template):
    """Mythological Figures depicted in the painting which are part of the content. These figures could depict scenes from Greek and Roman Mythologies."""

    painting: str  # The name of the painting, i.e. The Last Supper
    figure: List[str]  # The figures depicted in the painting, i.e. Venus, Volcan, Jupyter
```

```
def Attributes_relation_to_triplet(package: Attributes):
    return [
        {
            "subject": {
                "label": package.painting,
            },
            "relation": {
                "label": "shown with features",
                "wikidata_id": "wdt:P1354",
            },
            "object": {
                "label": object,
            },
        }
        for object in package.objects
    ]
```

SPARQL und Reconciliation mit Wikidata

- SPARQL zur Kommunikation LLM <-> Wissensgraph
- Reconciliation um nicht einen gesamten Graphen aufbauen zu müssen

SPARQL-Beispiel

```
# A query example for DHD2025
SELECT ?child ?childLabel ?image WHERE {
  ?child wdt:P22 wd:Q1339.
  OPTIONAL { ?child wdt:P18 ?image. }
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],mul,en". }
}
```

https://w.wiki/DHi3

https://w.wiki/CQBA

Abgleich mit Wikidata

- Ziel Abgleich von neu extrahierten Triplets mit existenten Graph (Wikidata)
- Matching von schon angelegten Beziehungen/Erkennung von neuen
- →“Entity Linking”
  Begriff “Reconciliation” kommt aus der praktischen Anwendung von OpenRefine -> Reconciliation API
- Alternative Ansätze mit Transformer, z.B ReFinED von Amazon

Reconciliation

DEMO