Vector Representations of Multi-Modal Data
(2-hour tutorial)

Toni Taipalus

Tampere University, Finland

Jiaheng Lu

University of Helsinki, Finland

Agenda

1. Introduction
2. Intra-modal Vector Representations
3. Inter-modal Vector Representations
4. Methods for Unifying Inter-modality Vectors
5. Vector Databases for Multi-modal Data
6. A categorical framework for multi-model database
7. Open challenge and conclusion

Introduction

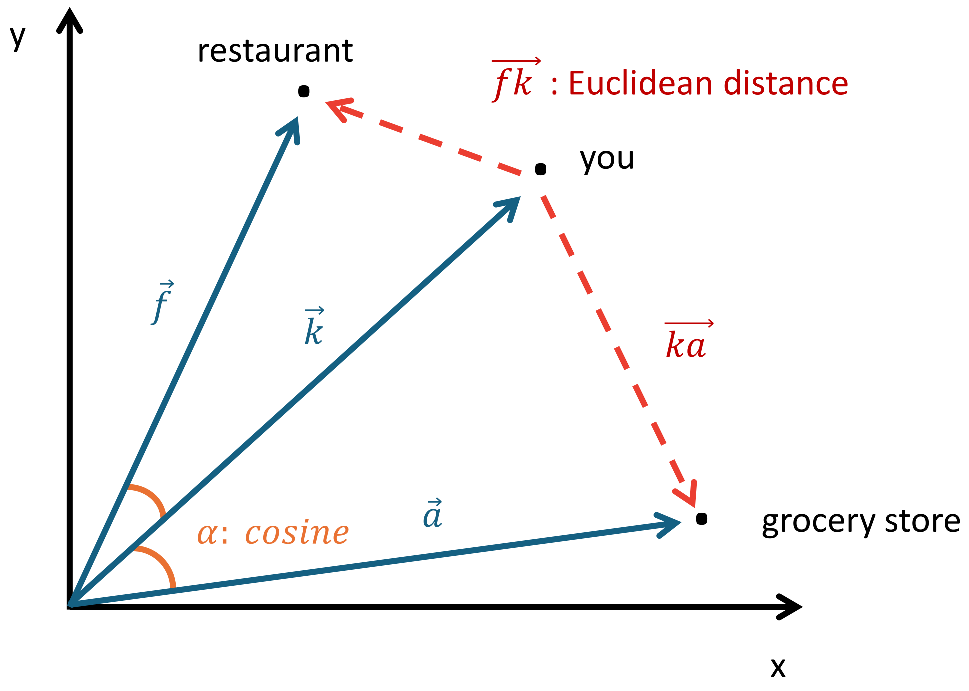
A word on terminology

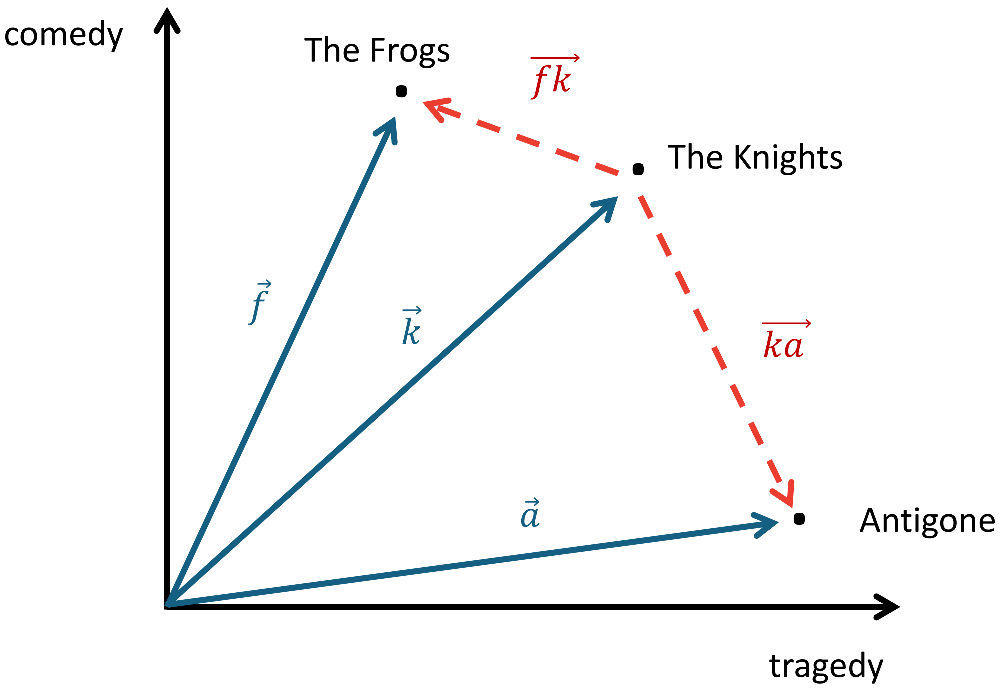
- Data modal / modality: the nature of data (e.g., text, image, audio)
- Not to be confused with data *model* (e.g., relational).

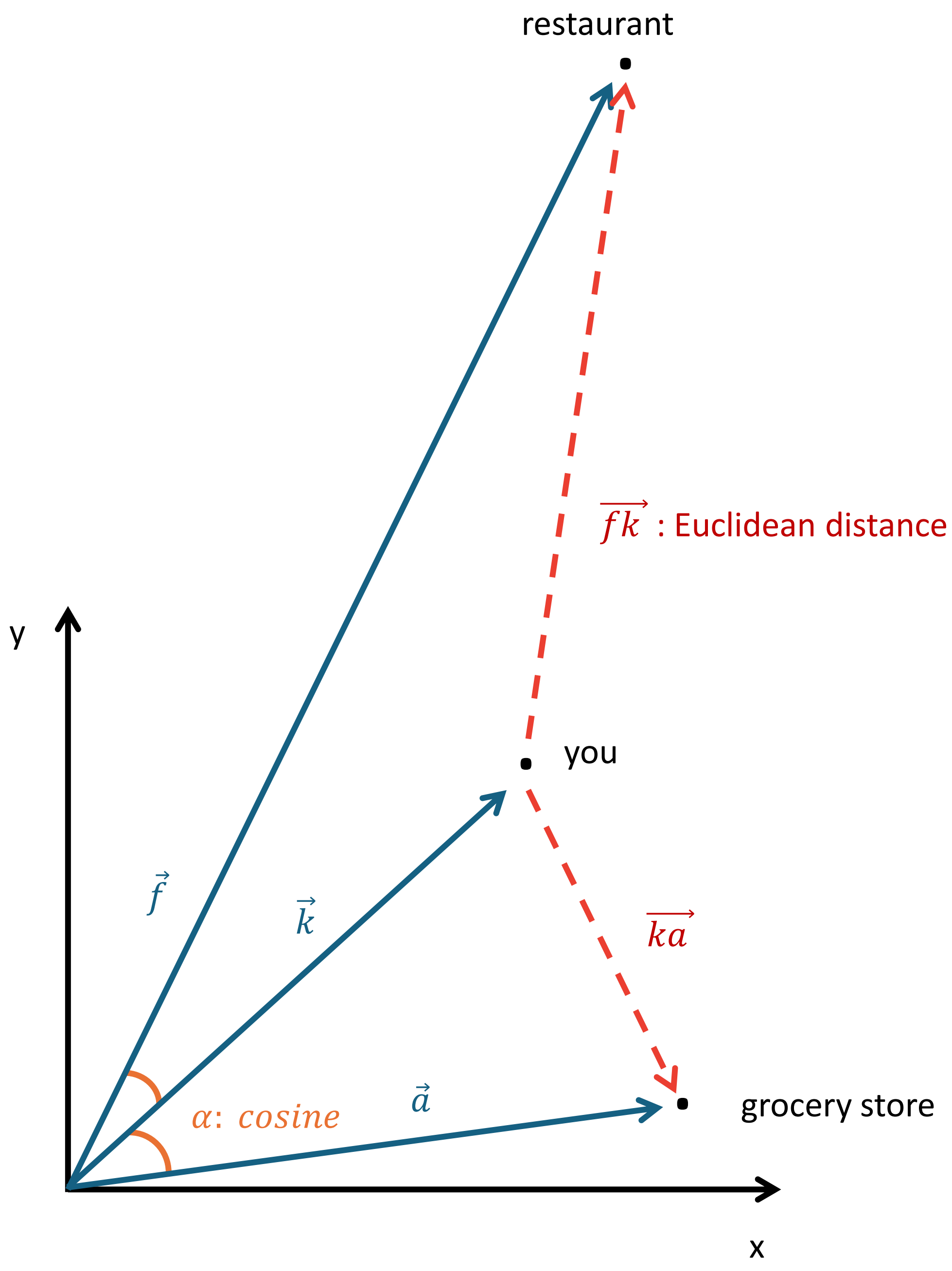
- Vector (embedding): a list of numbers representing something in a mathematical space
  - E.g., [1, 0, -2], a vector of three dimensions.
  - Dimensions capture the features of an object.
  - They measure similarity (close vector = similar object).
  - Can be low or high-dimensional. Can be dense or sparse.

Vectors can represent pretty much anything from location data to Greek plays

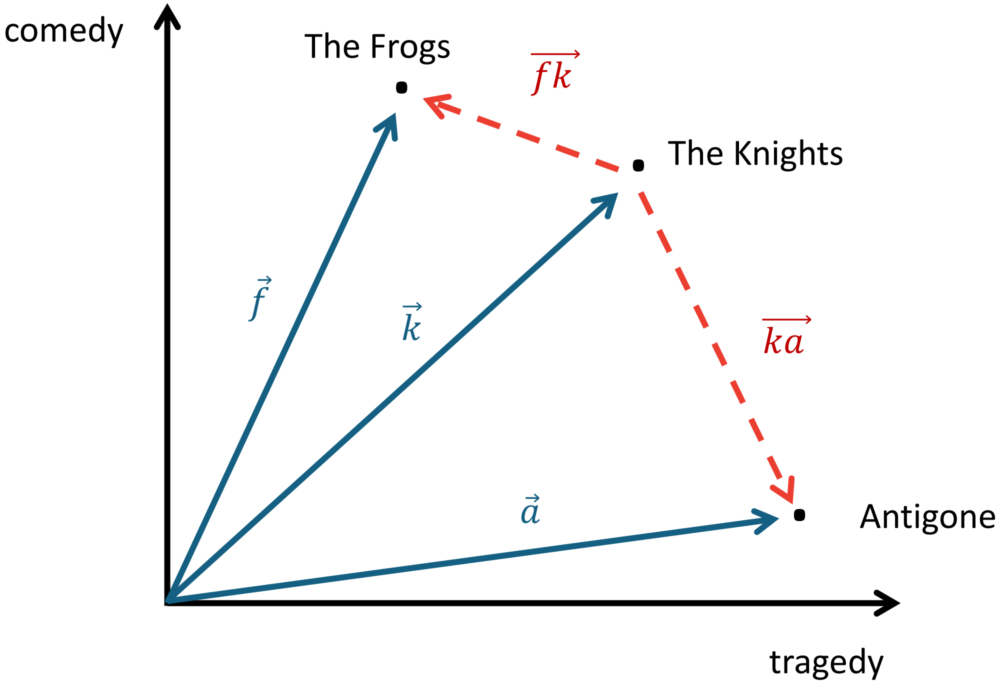
	tragedy	comedy	drama	scifi
The Frogs	3	9	2	0
The Knights	6	8	5	0
Antigone	9	2	6	0







restaurant
$\overrightarrow{fk}$: Euclidean distance
y
you
$\vec{f}$
$\vec{k}$
$\overrightarrow{ka}$
α: cosine
$\vec{a}$
grocery store
x

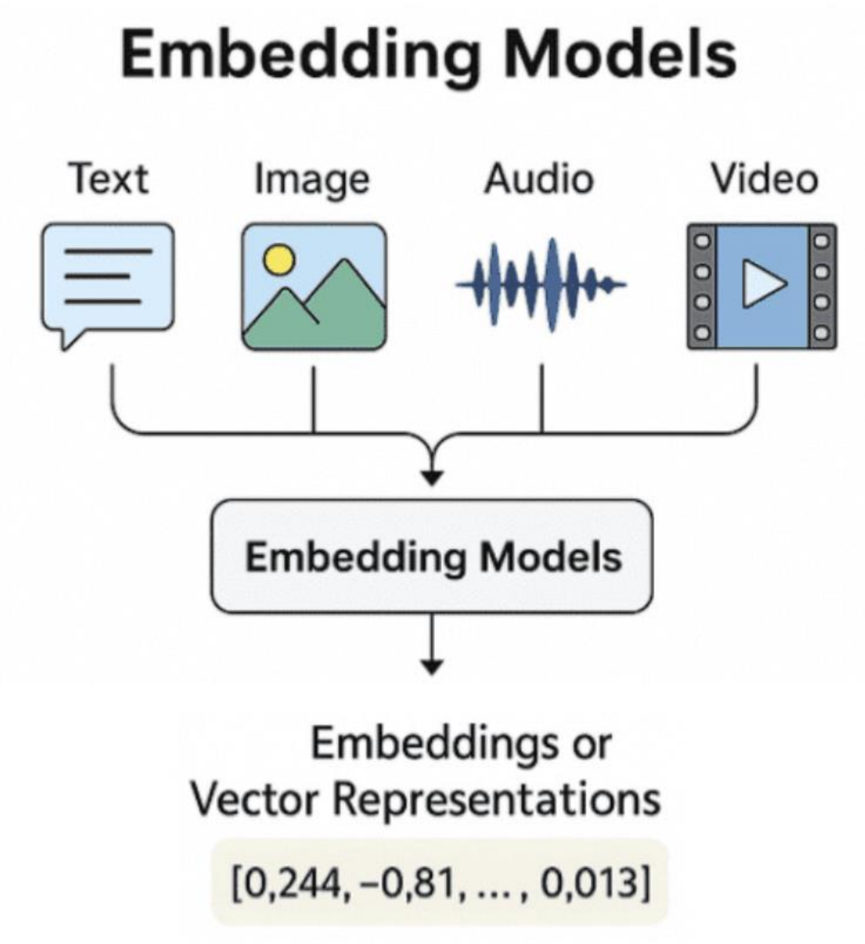


comedy
The Frogs
$\overrightarrow{fk}$
The Knights
$\vec{f}$
$\vec{k}$
$\overrightarrow{ka}$
$\vec{a}$
Antigone
tragedy

Intra-modal Vector Representations

Intra-Model Vector Representations

- Text embedding
- Image embedding
- Audio embedding
- Video embedding
- Time-series embedding



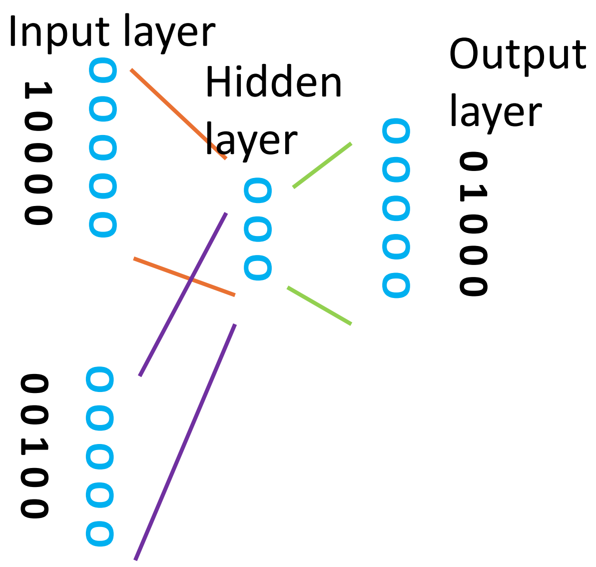
https://opencv.org/blog/vector-embeddings/

Text embedding: Word2Vec

- Word2Vec: Learn word associations via skip-gram or Continuous Bag-of-Words (CBOW)
- Word2Vec embeddings can capture analogical relationships, such as
- woman + (king - man) ≈ queen.
- Germany = Berlin + (France - Paris)

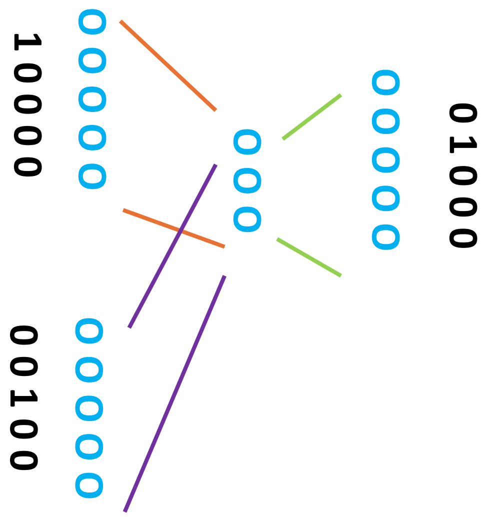
Continuous Bag of Words CBOW

- A Word2Vec architecture that predicts a target word based on the context words within a fixed window size.
- Learn word embeddings by training a neural network to maximize the likelihood of predicting the target word given its context.



Example of CBOW

- The objective of CBOW is to maximize the likelihood of predicting the target word given the context words.
- Consider the sentence: "The cat sleeps on the mat.“
- Step 1: Target word: cat; Context words: The, sleeps (with window size 3)
- Step 2: Vocabulary and One-Hot Encoding
  cat: [0, 1, 0, 0, 0] (target), on: [0, 0, 0, 1, 0], the: [1, 0, 0, 0, 0],
  mat: [0, 0, 0, 0, 1], sleeps: [0, 0, 1, 0, 0]



Step 3: Training the network multiple iterations to obtain the weight matrix.

Step 4: Compute the vectors based on the multplication of one-hot vector and weight matrix.

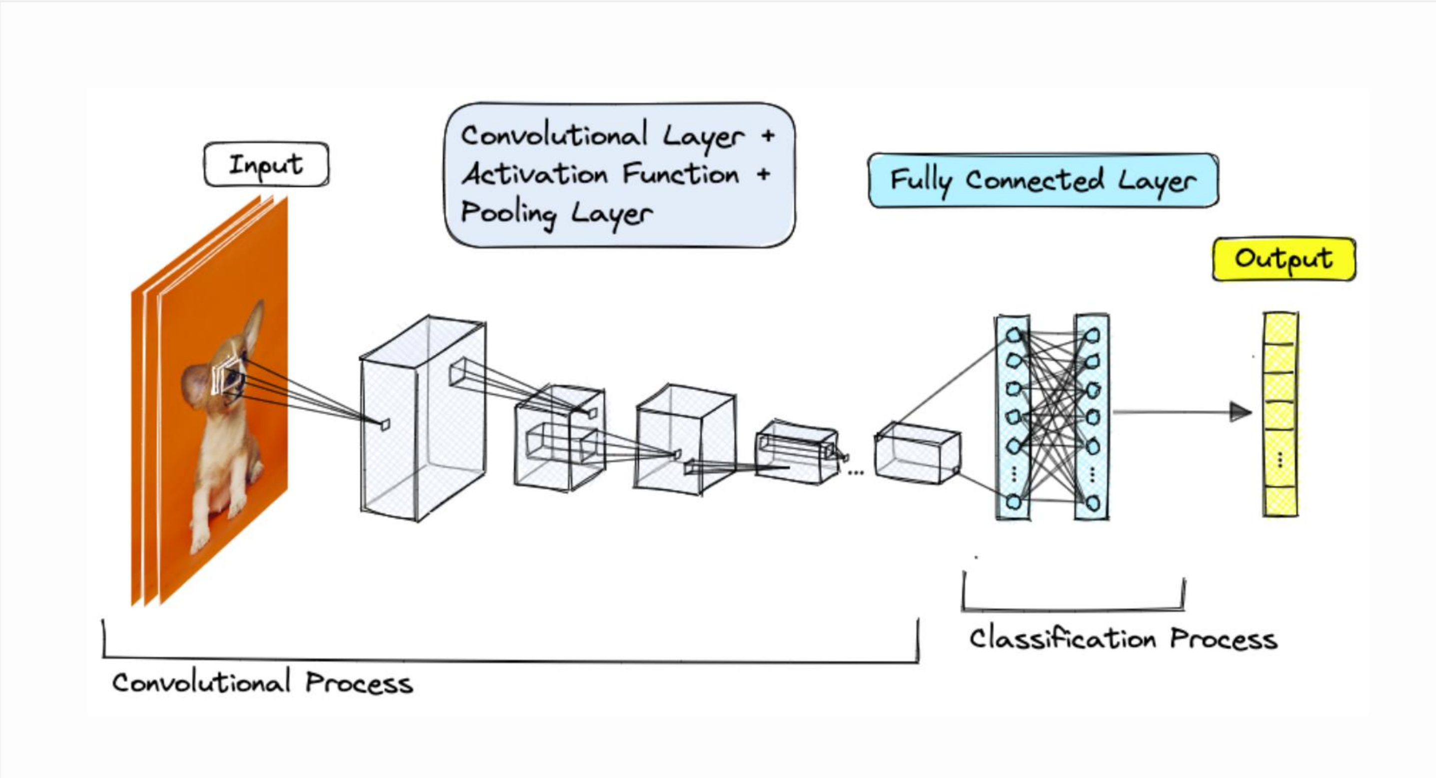
More text embedding methods

- Word2Vec: Learn word associations via skip-gram or CBOW.
- GloVe: Global word co-occurrence matrix factorization.
- TF-IDF: Frequency-based sparse embeddings
- BERT: Contextual embeddings using transformer architecture, which outperform static embeddings for nuanced tasks.

M, S., & B, A. (2024). A Survey of Machine Learning Technique for Topic Modeling and Word Embedding. 2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS), 1, 1-6. https://doi.org/10.1109/ICACCS60874.2024.10716820.

Image embedding

Image embedding algorithms extract distinct features in an image and represent them with dense vectors in a different dimensional space.



Source of the image: https://www.picsellia.com/post/image-embeddings-explained

Image embedding

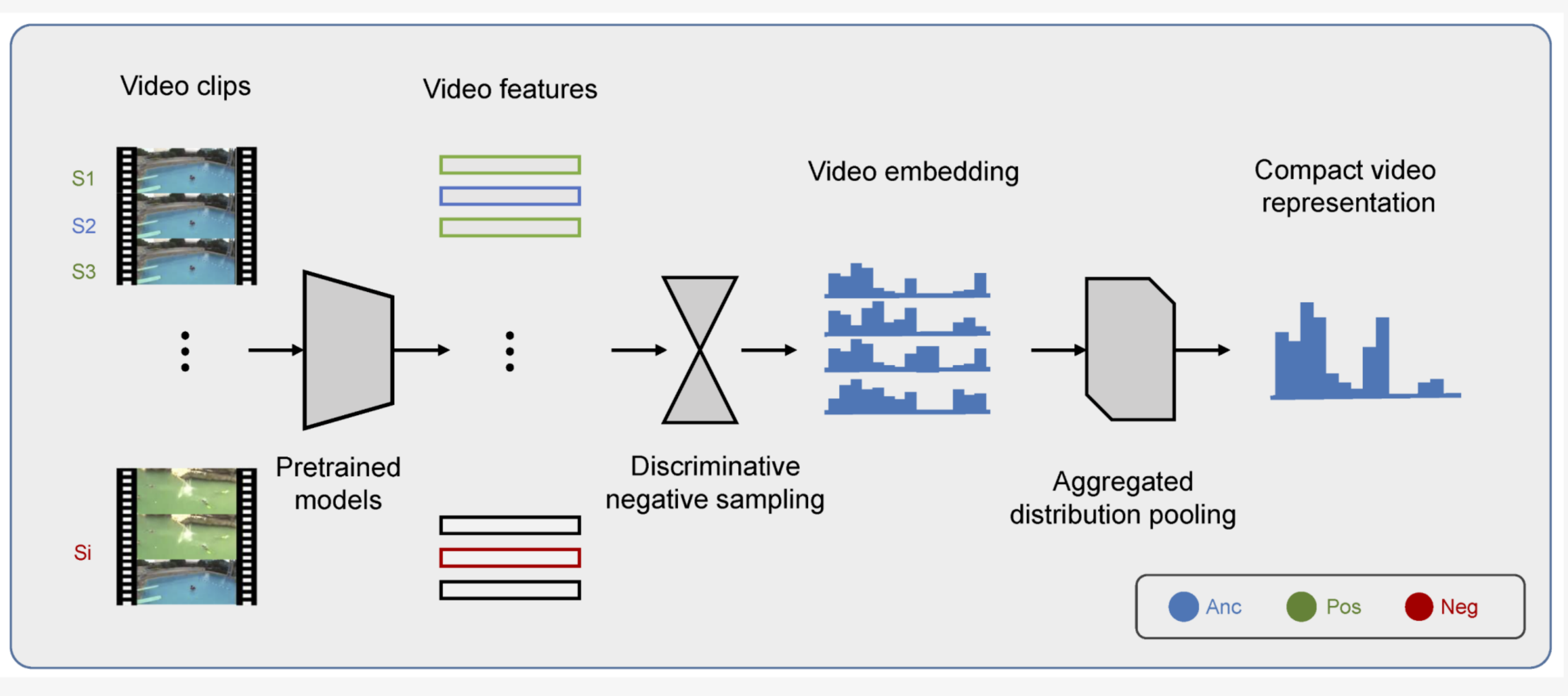
- Pixel value vectorization: Flatten the image's pixel values into a one-dimensional vector. E.g. 28*28= 784
- Feature extraction: Extract key features (e..g keypoints, textures, color histograms) and represent them as vectors
- Convolutional Neural Network (CNN): Use a pre-trained deep learning model to extract high-level features from fully connected or convolutional layers
- Image embeddings: Use specialized embedding models (e.g. Vision transformer, CLIP) to map iamges to a low-dimensional semantic space.
- Autoencoders: Train a neural network (e.g. variational autoencoder, VAE) to compress images into a low dimensional latent space, generating vector representation.

Audio embedding

- **Spectrograms**: Convert audio to a time-frequency representation using Short-Time Fourier Transform (STFT).
- **Mel-Spectrograms**: Apply a Mel filterbank to the spectrogram for a perceptually relevant representation
- **MFCCs (Mel-Frequency Coefficients):** Capture time aspects of audio, widely used in speech and music processing.
- **Chroma Features:** Represent harmonic content, useful for music analysis
- Based on the above features, dimensionality reduction and embedding.

Video embedding

- Converting a video into vectors involves transforming its visual and (optionally) audio components into numerical representations (vectors) in a lower-dimensional space



From paper: A Continuous Semantic Embedding Method for Video Compact Representation. Appl. Sci. 2021, 11(7), 3214; https://doi.org/10.3390/app11073214

Video embedding

- Sample frames at a fixed rate (e.g., 1 frame per second)
- Handcrafted Features: Use traditional computer vision techniques like SIFT, SURF, or HOG to extract keypoints or descriptors
- Pretrained CNNs: Use convolutional neural networks (CNNs) like ResNet, VGG, or EfficientNet to extract features from each frame.
- Temporal Aggregation: Videos have a temporal dimension, so combine frame-level features into a single vector or sequence of vectors.
- Normalization: Apply L2 normalization to ensure embeddings lie on a unit hypersphere for similarity tasks.

Time series embedding

- Converting time series data into vectors involves transforming sequential data points into a numerical representation (vectors)

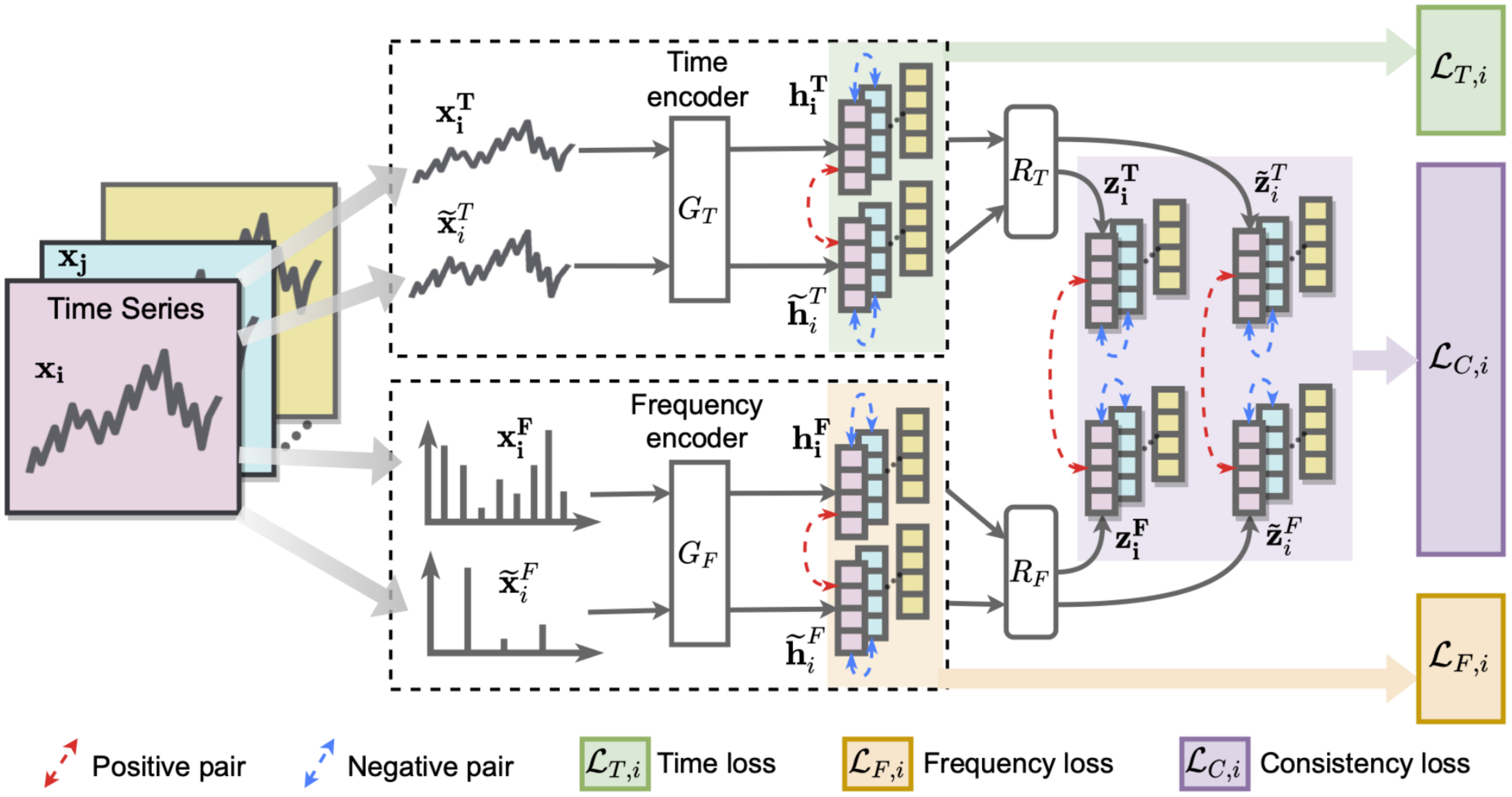


Image source: TF-C project: https://zitniklab.hms.harvard.edu/projects/TF-C/

Time series embedding

- Extract Features:
  - Statistical Features: Compute statistics like mean, median, variance, min, max, skewness
  - Frequency-Domain Features: Use Fourier Transform (FFT) or Wavelet Transform
  - Time-Domain Features: Extract autocorrelation, trends, or seasonality
  - Shape-Based Features: Use metrics like Dynamic Time Warping (DTW) distances or shapelets to capture patterns.
- Convert extracted features into compact vectors
  - Aggregation
  - Recurrent Neural Networks (RNNs): Use LSTM to process sequences and take the final hidden state as the embedding.

Inter-modal Vector Representations

Inter-modal vector representations

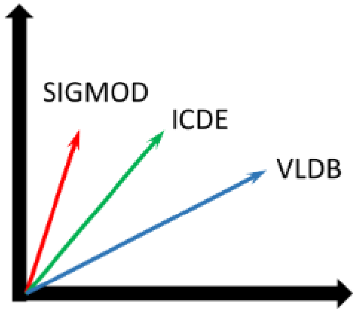
- **Intra**-modal vector representation: a vector space of data objects of single modality (e.g., image).

- **Inter**-modal vector representation: either
  a) A shared vector space for multiple modalities. Encoders are trained so that their embeddings land in the same space (the word '*cat'* is close to the image of a cat).
  b) Single vectors created from multi-modality data objects (a fusion of several data objects, e.g., text + audio).

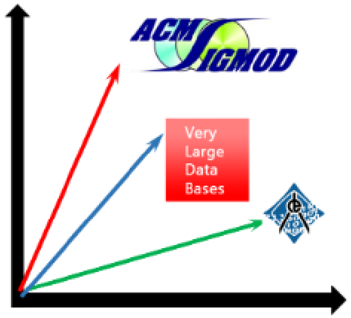
- Definition depends on the context.

Why inter-modal?

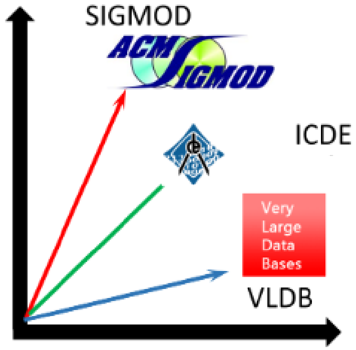
- Data takes different forms: perhaps search should cover all sources of information, not merely one.



(a) Vector representations for text



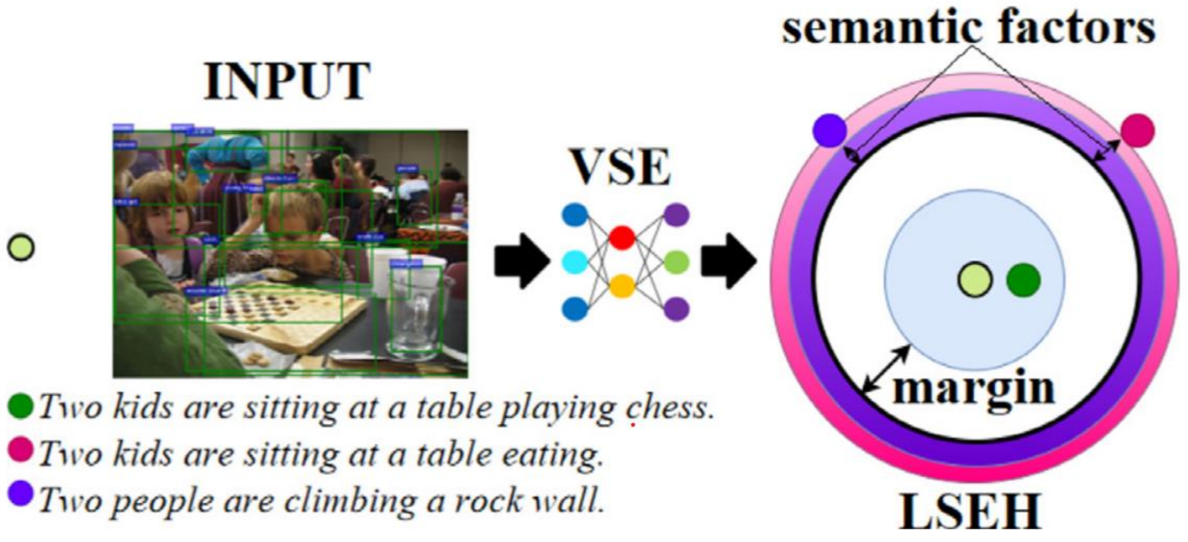
(b) Vector representations for image



(c) Joint vector representations for text+image

Modal combinations: text + image

- Image captioning & cross-modal retrieval
- Applications: search engines, accessibility
- Connect to embedding into a *common vector space*



[Gong et al. PR'22]

Modal combinations: text + audio

- Speech-to-text alignment
- Summarization: audio → text → summarization model
- Speech2Vec: convert audio into text embeddings
- Search podcasts with textual queries…
- …or retrieve text transcripts by audio queries.
- Speaker identification + emotion recognition: recognize tone, pitch & accent → align with textual sentiment vectors

Modal combinations: image + audio

- Synchronization, e.g., lip sync
- Emotion recognition: facial features + voice cues

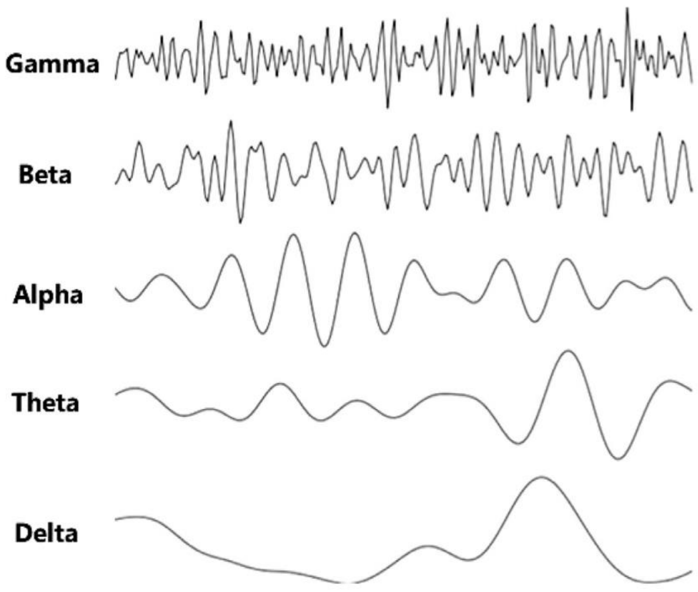


Fig. 3. The waveforms of five typical EEG rhythms.

\+

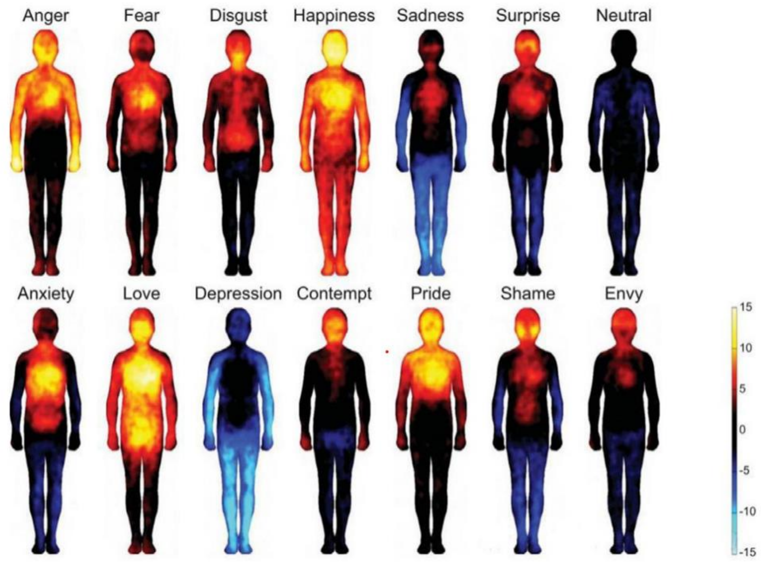


Fig. 1. The bodily map of human emotions.

[Zhang et al. IF'20]

Modal combinations: text + image + audio

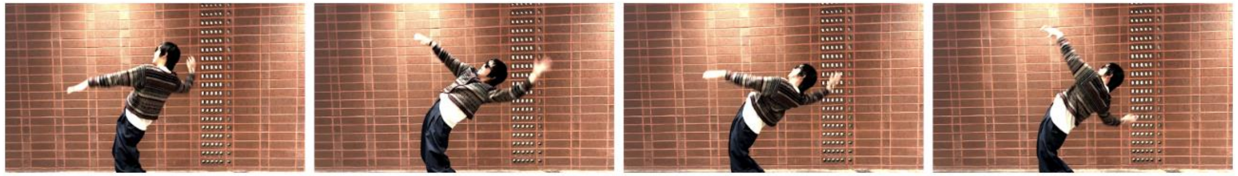
- Multi-modal recommender systems and digital assistants

Big families of models

- Inter-modal representations are not an abstract idea:
  - CLIP (text + image)
  - AudioCLIP (adds audio)
  - VisualBERT / LXMERT (vision + language transformers)

Types of inter-modal tasks

- Cross-modal retrieval (query one modality, retrieve another)
- Cross-modal generation (text → image, text → speech, etc.)
- Cross-modal reasoning ("visual question answering")



```
Guess what movie I'm acting out.
```

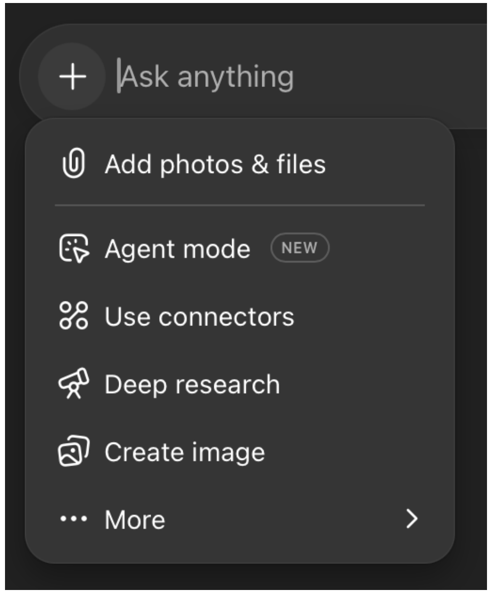
```
Gemini: The Matrix
```

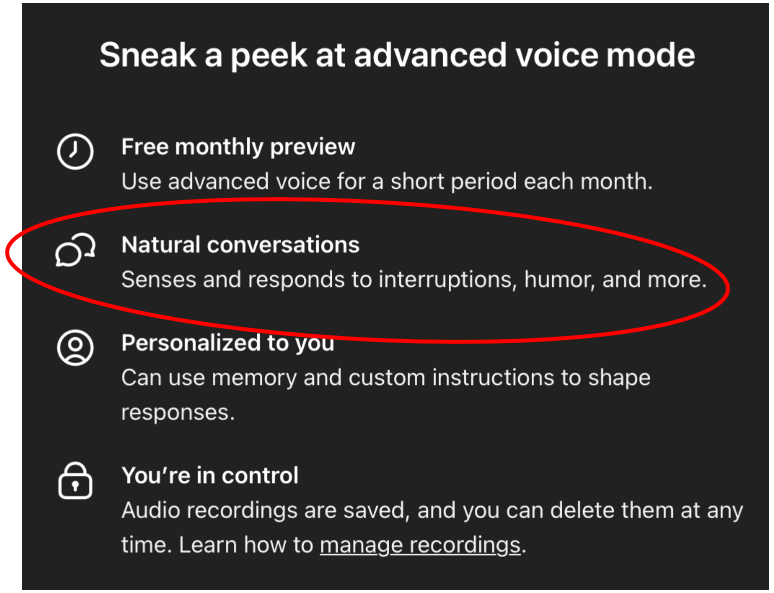
```
Nice! But which part specifically? Look at my body movements.
Gemini: The part where Neo dodges bullets.
```

[Google Geminil]

Where is inter-modality going?

- Increasing role of large multimodal foundation models (Gemini)
- Integration with vector DBMSs for real-time applications
- Personalized multimodality (adapt embeddings for users)





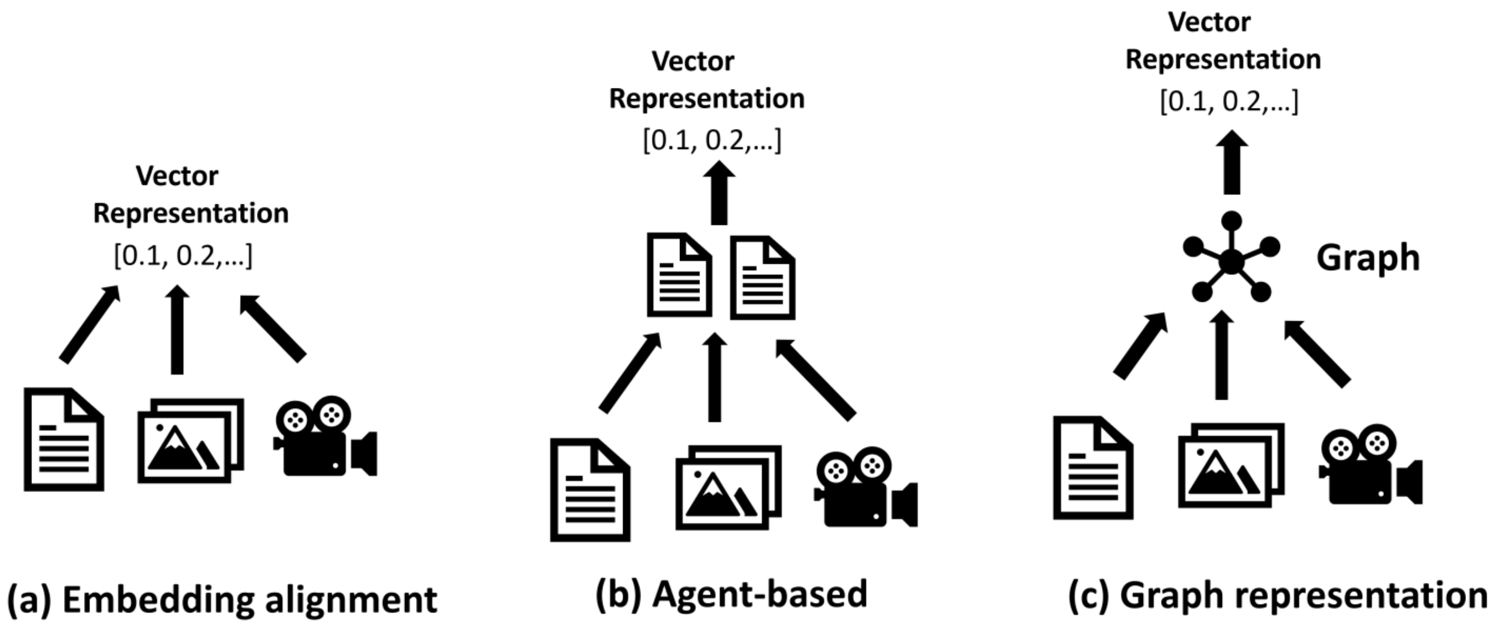
[OpenAI]

Inter-modal vectors: challenges

- How to align continuous with discrete spaces.

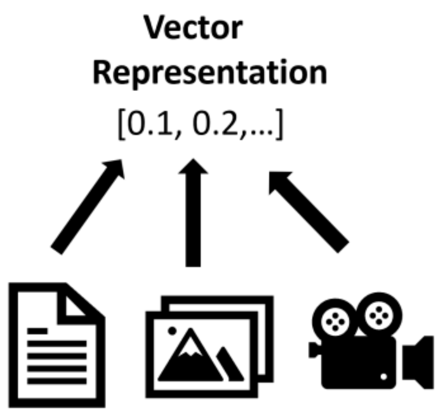
Methods for Unifying Inter-modality Vectors

Three methods for unification



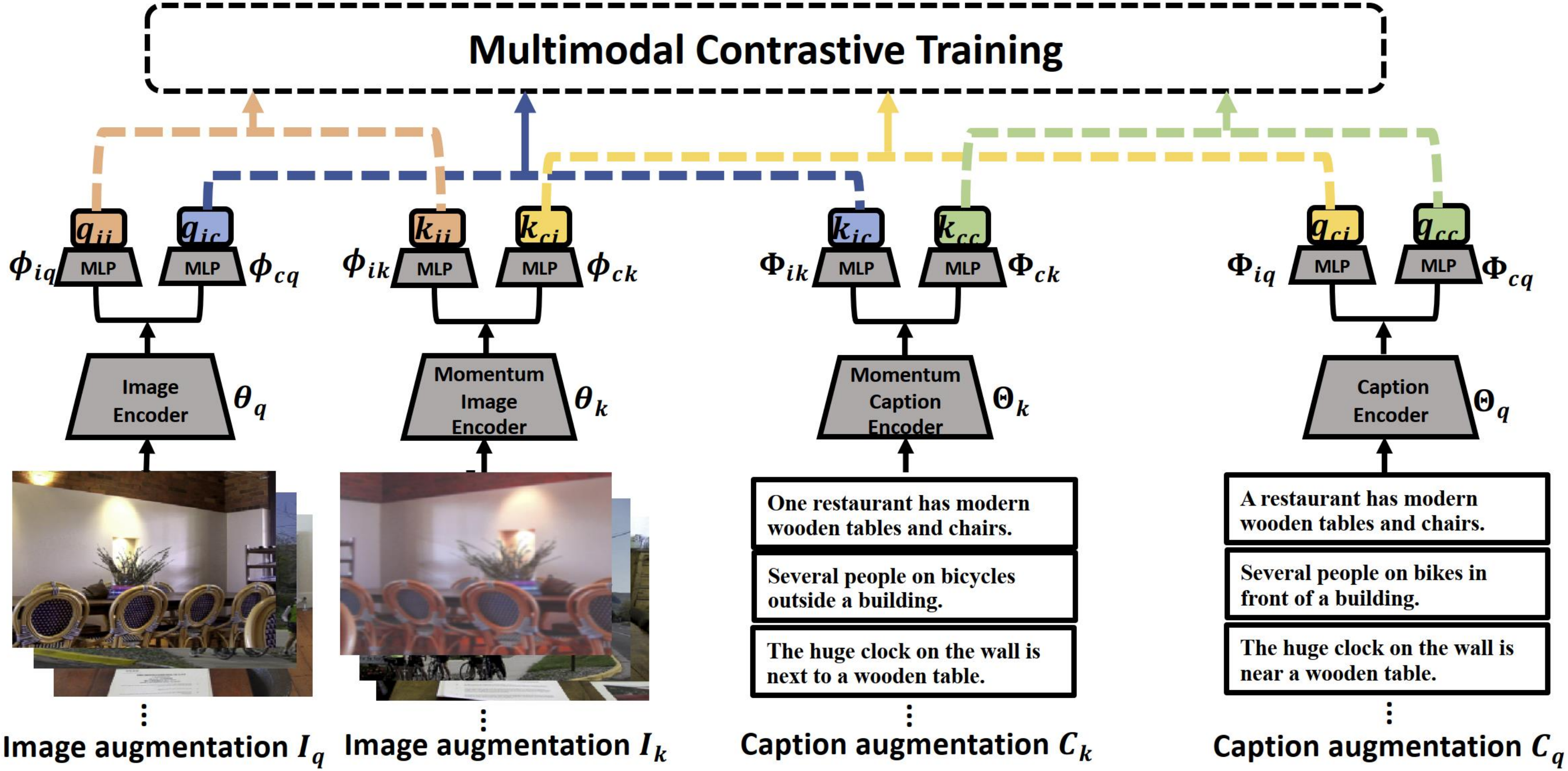
Embedding alignment

- Models are trained with the aim of learning a joint embedding space where representations from different modalities are embedded such that similar instances across modalities are closer together in the space, and the distance between dissimilar instances is maximized.



(a) Embedding alignment

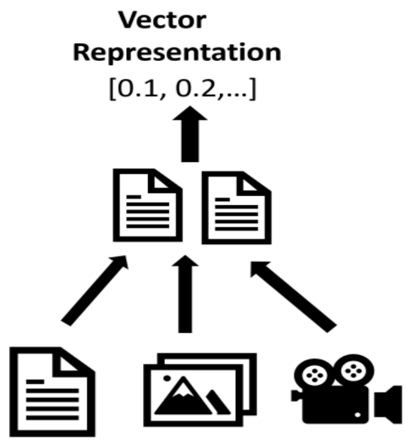
Embedding alignment with contrastive learning



From paper: Multimodal Contrastive Training for Visual Representation Learning https://arxiv.org/pdf/2104.12836

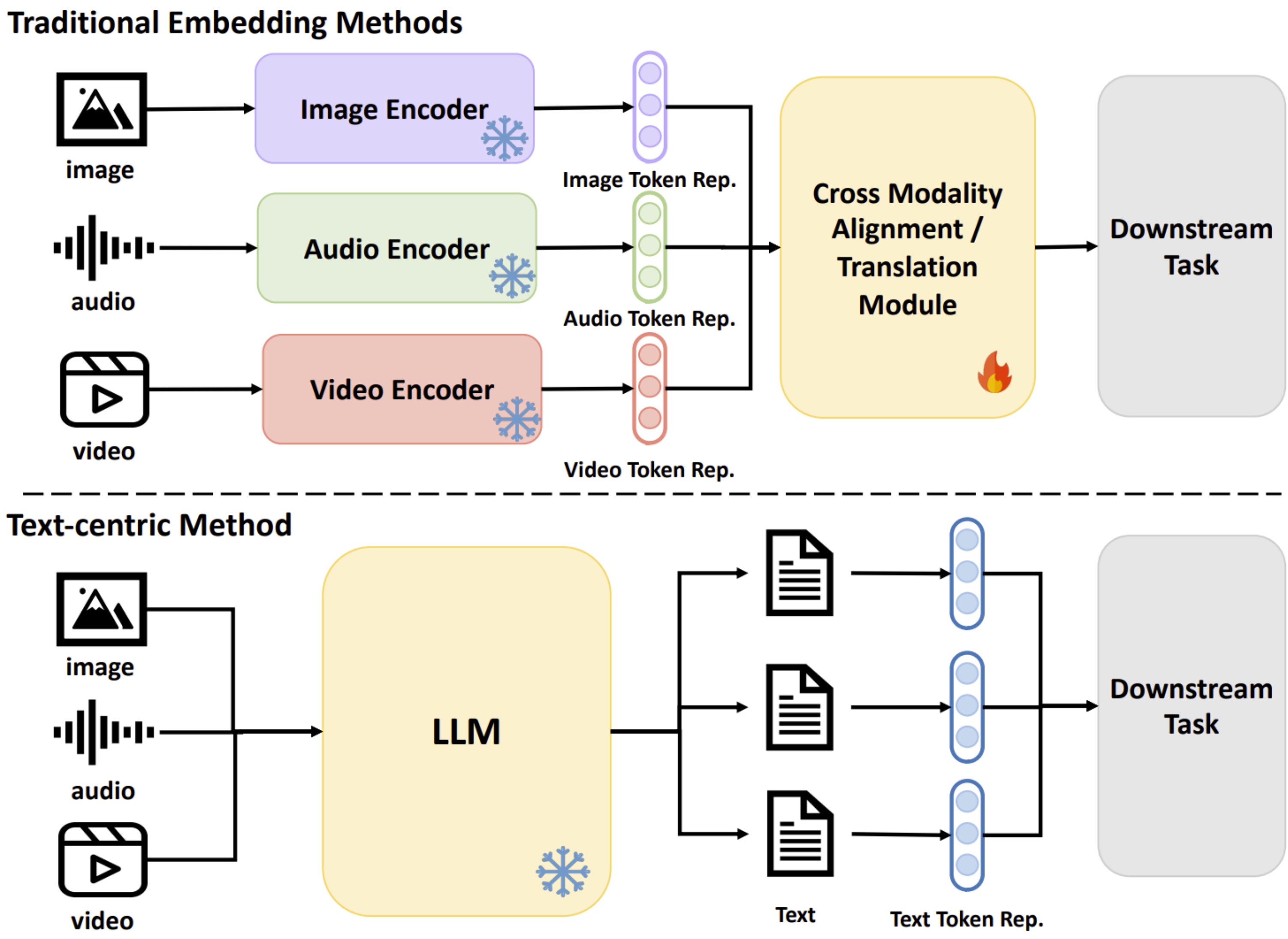
Agent-based alignment:

- Converting all data objects to a single modality, e.g., textual descriptions, which are then vectorized. This approach circumvents the challenge of vectorizing multi-modality data objects directly into the same vector space.
- Limit: Describing audio fully in text can be challenging, potentially limiting accuracy.



(b) Agent-based

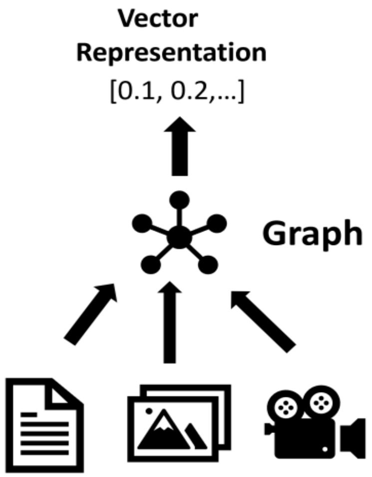
Enhance the Robustness in Text-Centric Multimodal Alignments



Paper link: https://arxiv.org/pdf/2407.05036

Graph representation:

- Construct a graph-based representation where nodes represent instances and edges capture relationships or similarities between instances across different modalities.
- Knowledge graph can then be applied to learn a unified representation by aggregating information

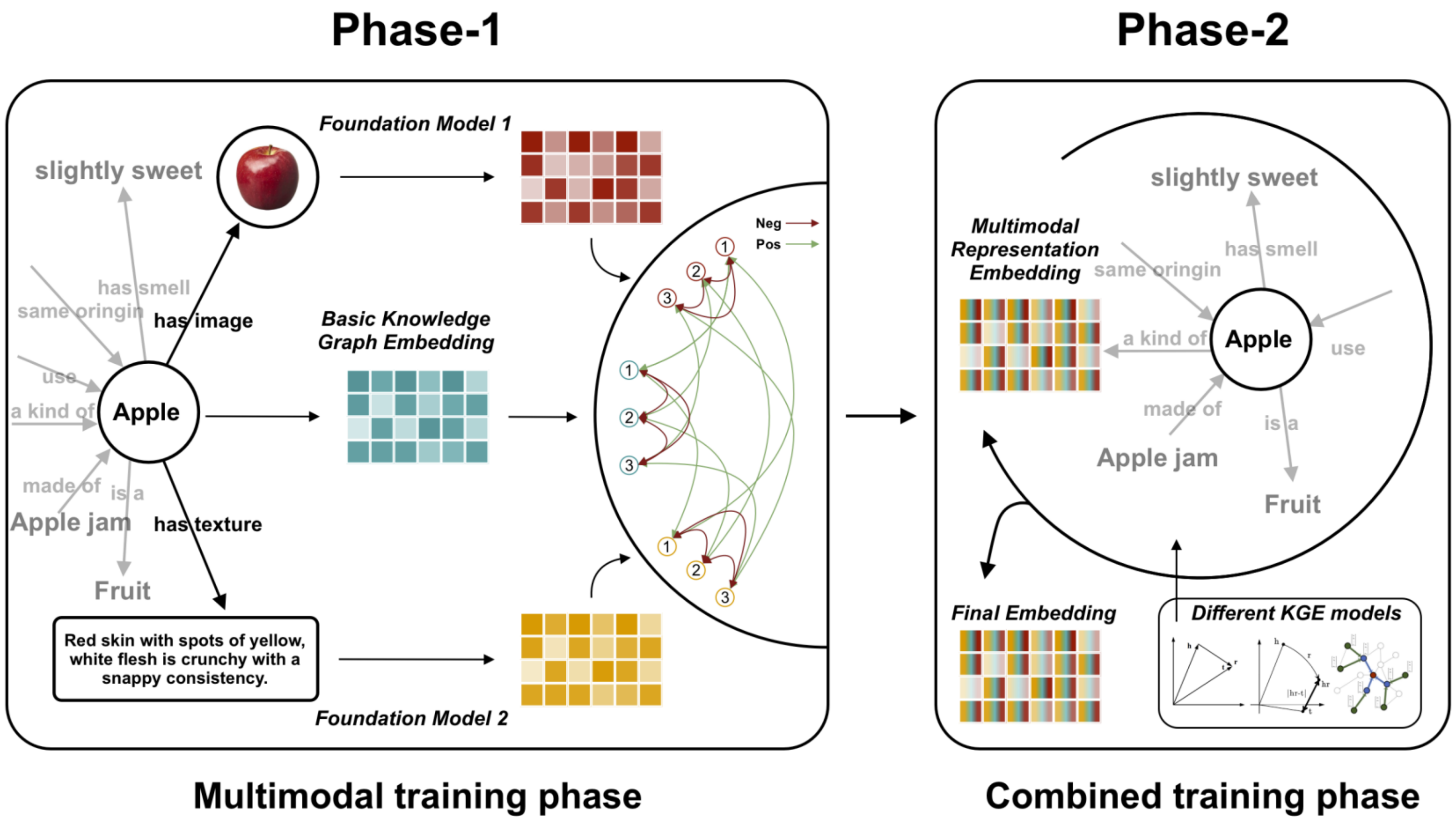


(c) Graph representation

Graph representation:

Two different modalities of an entity are retrieved.

These two distinct representations, along with outputs from the basic KGE method, are integrated using a triple contrastive learning module to enhance alignment.

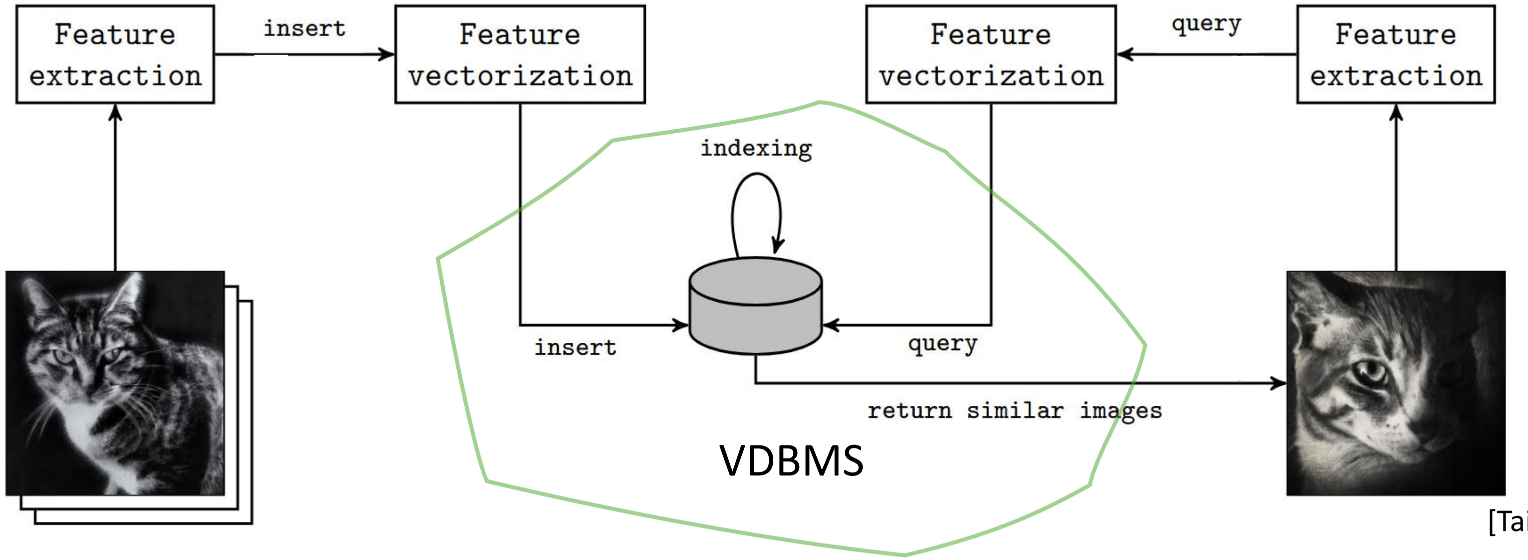


From the paper: Enhancing Multimodal Knowledge Graph Representation Learning through Triple Contrastive Learning

Vector Databases for Multi-modal data

Why vector databases (or VDBMSs)?

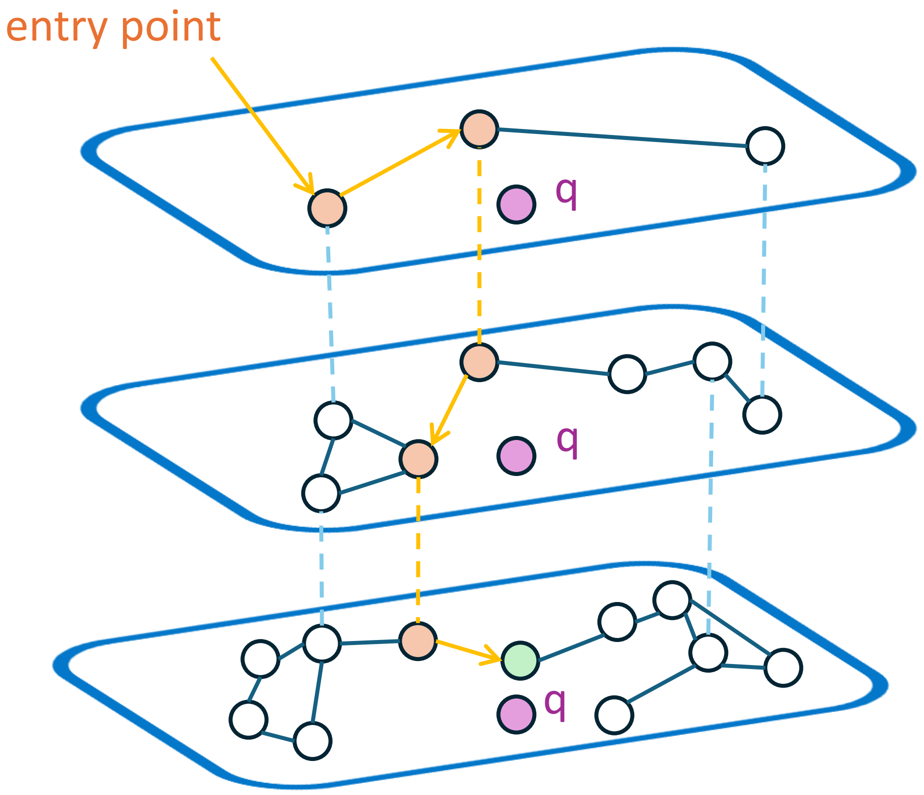
- Efficient storage, indexing, and querying of the vectors.
- Traditional DBMSs handle different data models.
- Need for similarity search (ANN, k-NN).



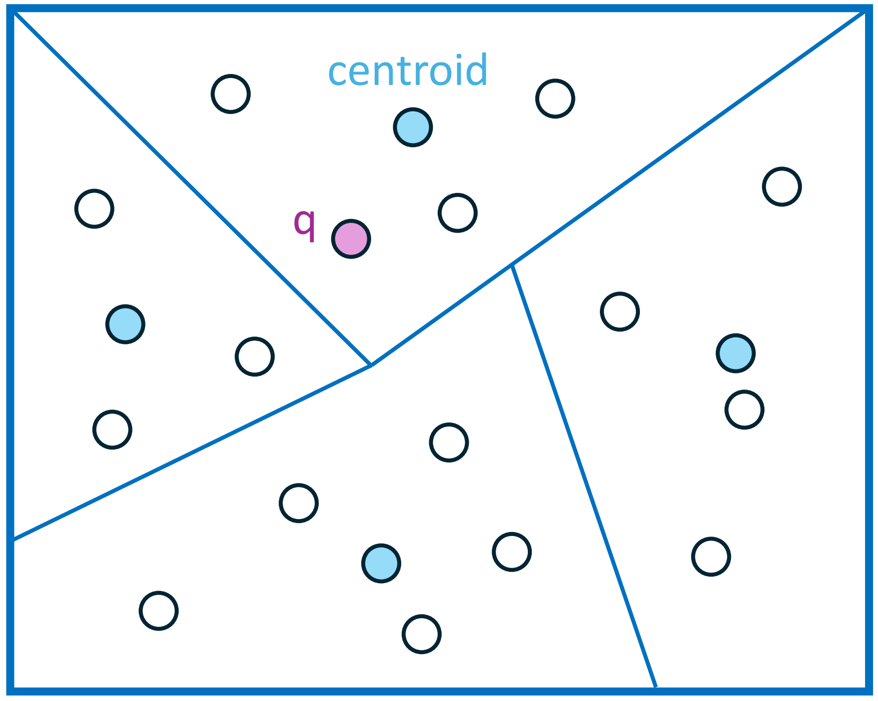
[Taipalus CSR'24]

Vector indices

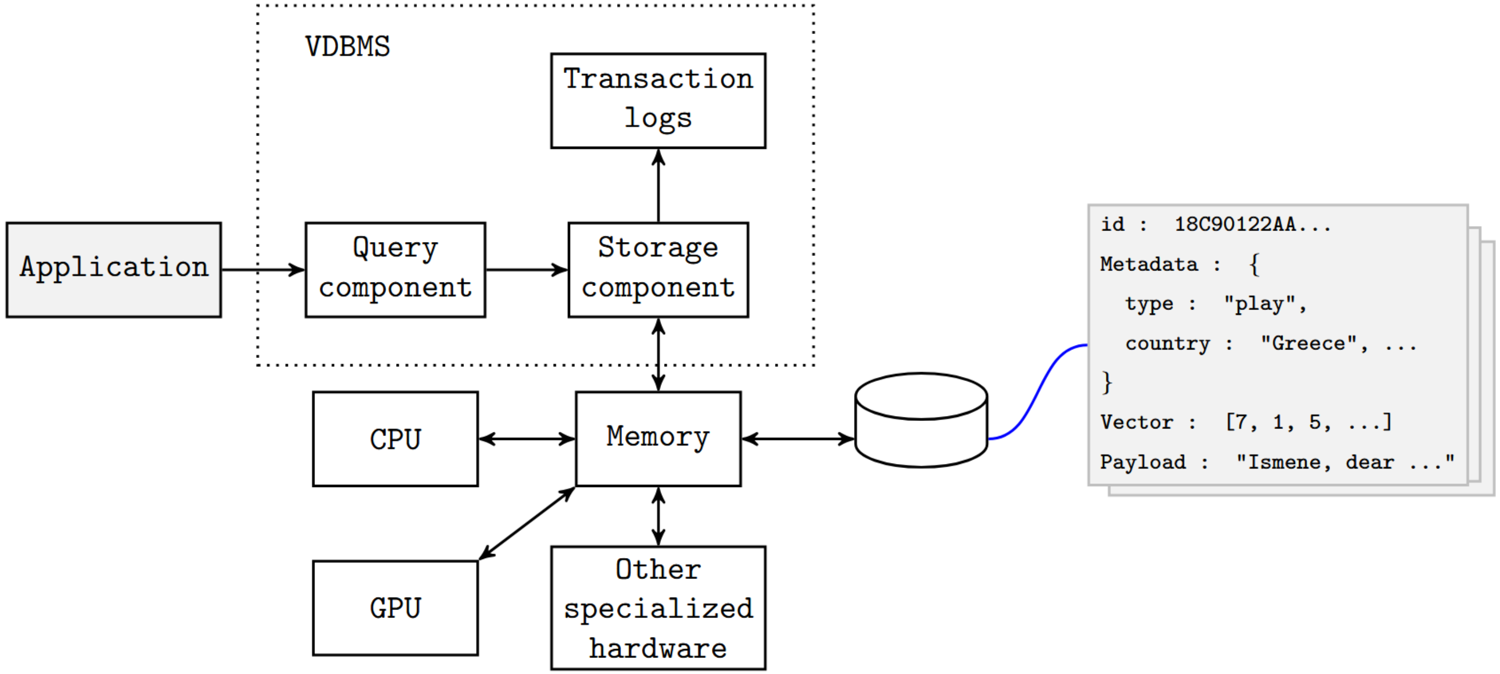
Graph-based



Tree-based



Why vector databases?



[Taipalus CSR'24]

"Vector databases" come in different flavors

- Search engines and libraries (not exactly VDBMSs, but here for comparison)
  - Apache Lucene, FAISS (META), Annoy (Spotify), ScaNN (Google), etc.
  - Aimed at specific searches
  - Not a system *per se*
  - High performance & high-quality search
- Native VDBMSs
  - Pinecone, Weaviate, Milvus, Chroma, etc.
  - Aimed at different use cases
  - System-level features
- Vectors as an extension to a DBMS
  - PostgreSQL (pgvector, PASE), Redis, MongoDB, Oracle Database, etc.
  - Adapt existing database structures to vectors

"Vector databases" come in different flavors

	VDBMSs	Extended DBMSs	Libraries
Query processing	ANN k-NN Range search Keyword search Full-text search Multi-vector search…	ANN (pgvector ≥ 0.5) k-NN Full-text search Relational queries	ANN k-NN Range search
Storage and indexing	Replication Sharding Backups HNSW, PQ…	Replication Sharding Backups HNSW, PQ...	HNSW, IVF, PQ…
Query optimization	Cost-based (Milvus)	Cost-based	-
Storage manager	Scatter-gather	MVCC ACID Scatter-gather	Scatter-gather

How to handle multi-modality?

- Store vectors as a column in a relational database.
- Store vectors as a value in KV-store or document DB.
- Query with relative ease:
  - Hum a tune (audio vector) → retrieve similar songs with text metadata
  - Type text ("chest pain") → retrieve X-ray embeddings + patient notes
- Hybrid search:
  - Combine vector similarity search with structured predicates

```sql
  SELECT * --general
    FROM t1
   WHERE genre = 'rock'
ORDER BY distance(q, vector_column)
   LIMIT 10;
```

```sql
  SELECT * --pgvector range search
    FROM t1
   WHERE genre = 'rock'
     AND vector_column <-> q < 1.0;
```

```sql
  SELECT * --pgvector ANN/k-NN
    FROM t1
   WHERE genre = 'rock'
     AND vector_column <-> q
   LIMIT 5;
```

Number of dimensions

- Balancing latency and precision
  - Indices abstract and divide vectors
  - How many dimensions are needed for a particular use case?
    - “System A supports” up to 32,768 dimensions
    - “B supports” up to 16k dimensions, but everything above 2k is very slow
    - “C supports” 10k+ dimensions
    - “D optimized” for 100 to 1,000 dimensions
    - “E practical with” a few hundred dimensions
- Above ~2k dimensions: the curse of dimensionality 😵
  - Distance becomes less meaningful
  - Indices become less efficient

Lifecycle management

- Models update → vectors drift.
- Real-world updates → vectors drift.
- Vector versioning

Current challenges

- Keeping up with new and improved embedding models

- Vectors are not human readable
- Vectors may contain sensitive data

```sql
SELECT *
  FROM t1
 WHERE genre = 'rock';
```

```sql
SELECT *
  FROM t1
 WHERE song <-> '[0,1,0,2,...]';
```

A categorical framework for multi-model database

- Difference between Multi-model and Multi-modal

- Examples of Data Models :
  - Relational (tables with rows/columns), Document (JSON, BSON, XML), Graph (nodes/edges), Key-value, Columnar, Time-series, Vectors
  -
- Examples of Modalities :
  - Text (documents, queries), Images (photos, diagrams), Audio (speech, music), Video (sequences of images + audio), Sensor data (e.g., IoT, biomedical signals)

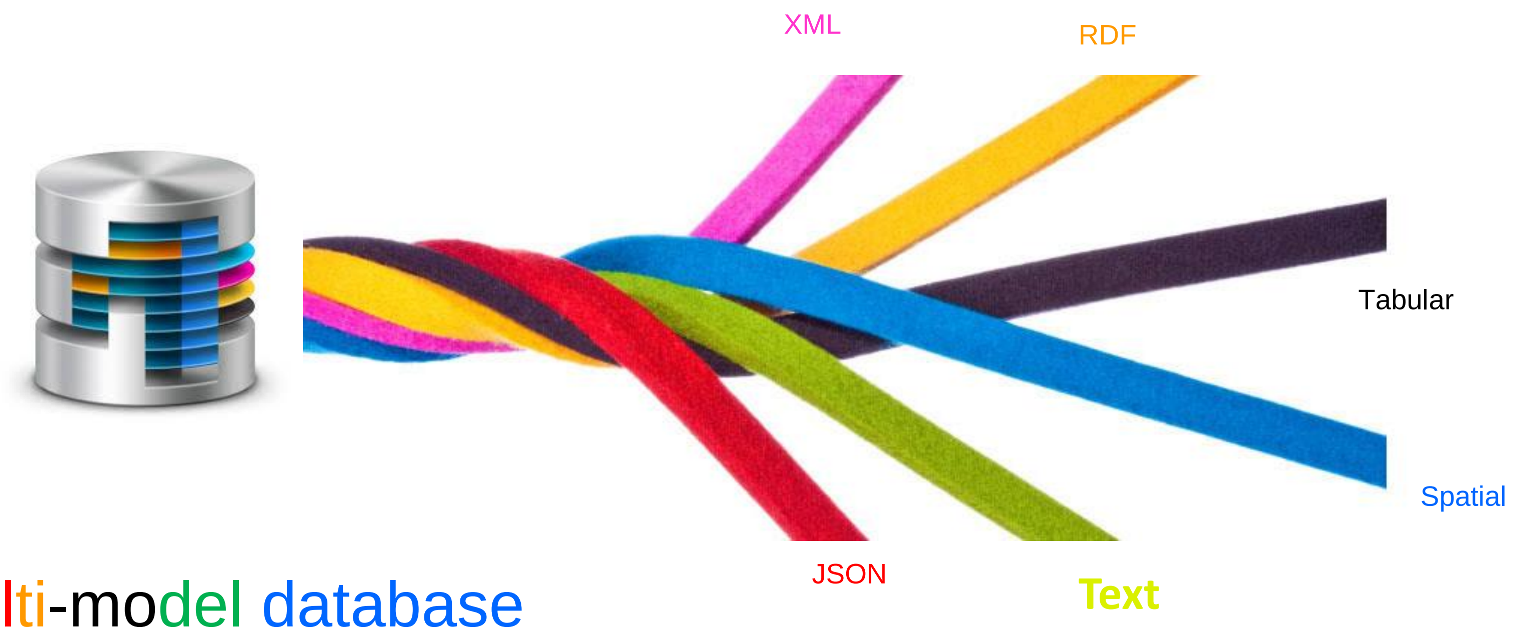
What is a multi-model database system?

- A multi-model database management system is designed to support multiple data models against a single, integrated backend.

- Document, graph, relational, key-value and vector models are examples of data models that may be supported by a multi-model database.

Multi-Model Databases System

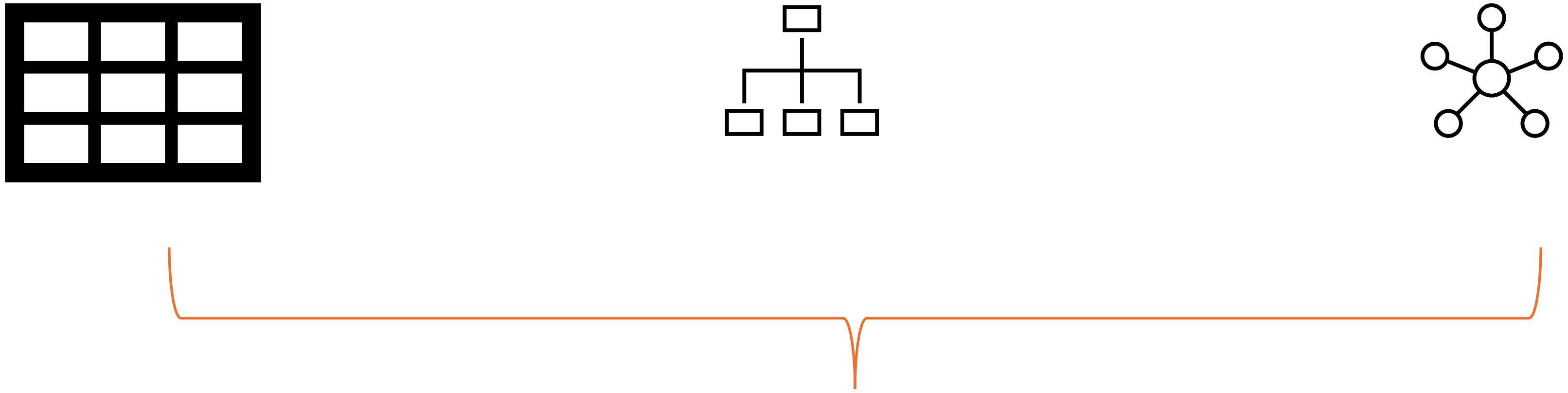
- One unified database system for multi-model data



Multi-model database systems

Research challenge:

How to build a unified model and framework for multi-model data



?

Research challenge:

Research goal:

Call for a unified model and theory for multi-model data!

The theory of traditional relations is not adequate to mathematically describe modern database systems.

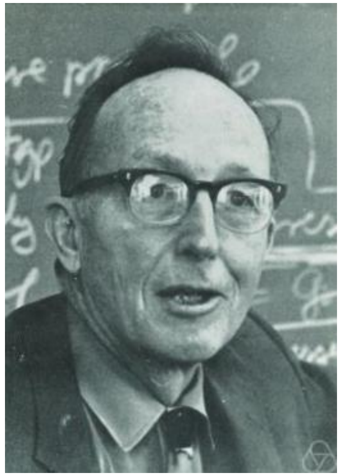
Research significance: AI + DB key technology

- Enhanced Querying and Analytics: Graph RAG and Multi-model RAG
- Simplified Architecture: Unified storage for structured, semi-structured, and unstructured
- Improved Interoperability

Theory foundation: Category Theory

- Introduced to mathematics world by Samuel Eilenberg and Sauders MacLane in 1944
- Developed for a unified language of topology and algebra

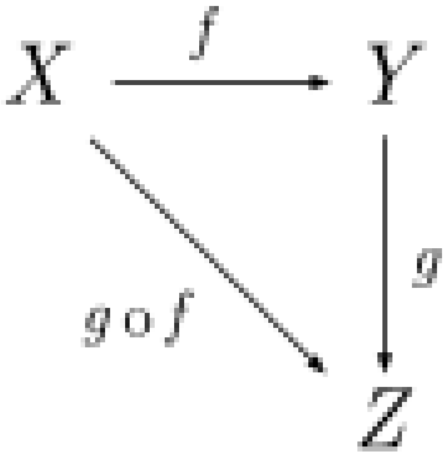
Samuel Eilenberg



Sauders MacLane

Categories Defined

- A category C is ….
    - a collection of objects ob(C) .. {X,Y, Z ….}
    - a collection of morphisms {f, g ….}
        - A set of morphisms from object X into Y is denoted by Hom (X, Y) or X→Y.

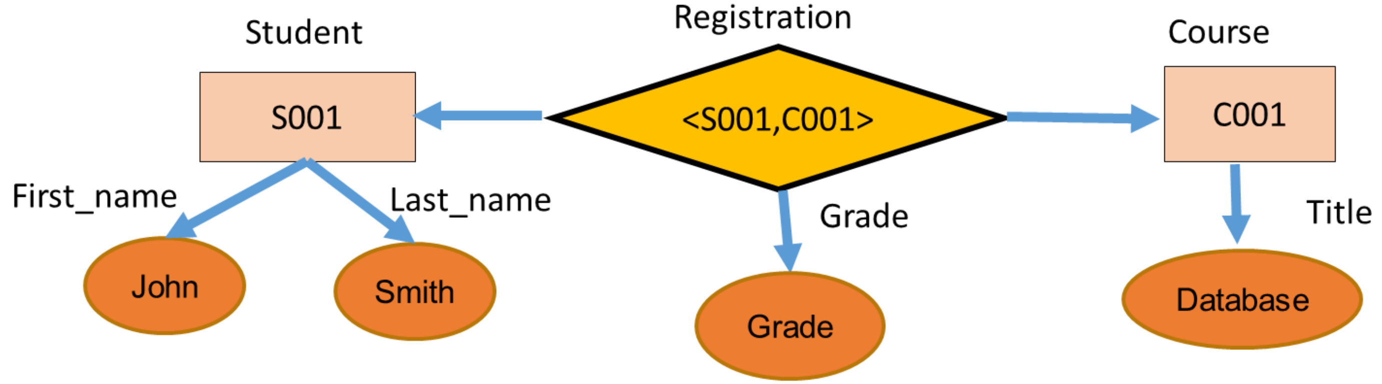


Set Category

- Databases are set categories:
  - Objects are sets and morphisms are functions

- We assume that it is a thin Category (or Posetal Category)
  - Given a pair of objects X and Y in a category C, and any two morphisms f , g: $X \to Y$, we say that C is a thin category if and only if the morphisms f and g are equal.

An example of Categorical Unification



Unified Category for three types of data

Student

Key	FN	LN
S001	John	Smith

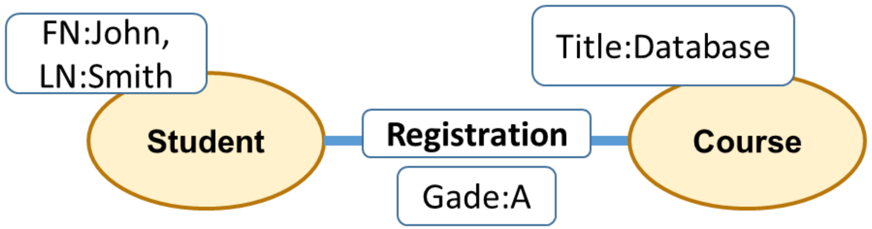
Course

Key	Title
C001	Database

Registration

SKey	CKey	Grade
S001	C001	A

Relation



Property graph

<Student key="S001">
<First_name> John </First_name>
<Last_name> Smith </Last_name>
</Student>
<Course key="C001">
<Title> Database </Title>
</Course>
<Registration skey="S001" ckey="C001">
<Grade> A </Grade>
</Registration>

XML

Relational algebra and relational calculus

- In the field of relational databases, relational algebra and relational calculus are developed as two formal languages for query databases.

- Analogously, categorical algebra and categorical calculus are developed to query category databases.

Relational algebra

- Operators:
  - Selection: σ (sigma)
  - Projection: Π
  - Union: $\cup$
  - Intersection : $\cap$
  - Difference: -
  - Cartesian Product: $\times$
- Derived operators:
  - Joins (equi-join) $\bowtie$

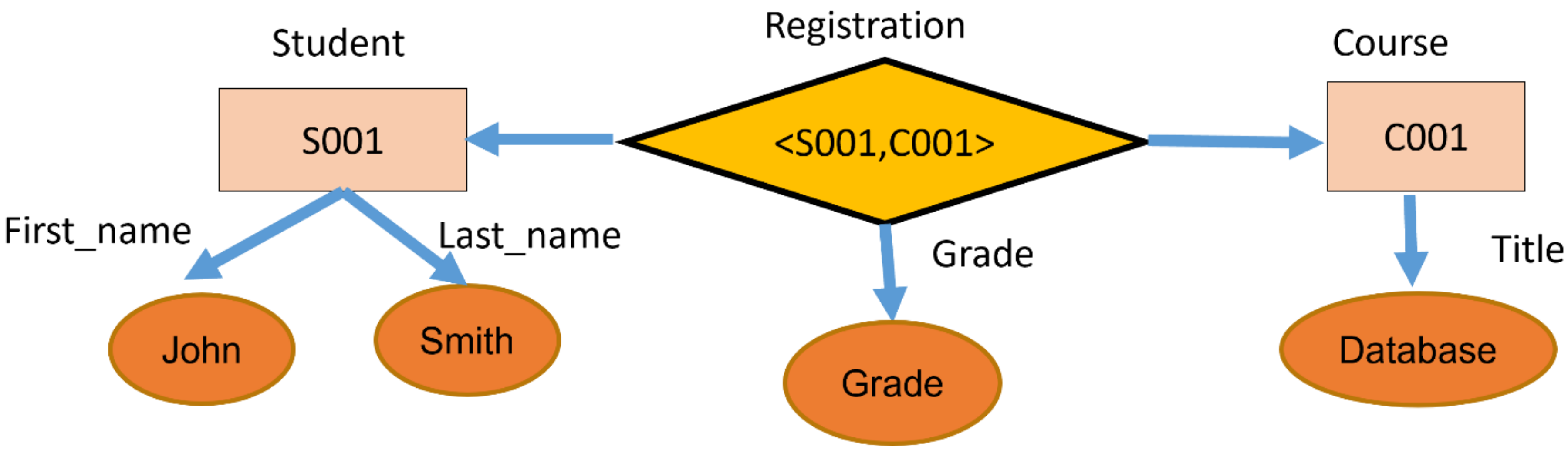
Categorical algebra

- Set operators:
  - Unary operator:
    - Map: f
    - Selection: σ (sigma)
    - Projection: Π
  - Binary operator:
    - Division: $\div$
    - getParent(D_1,D_2)
    - getAncestor(D_1,D_2)
  - Tenary operator:
    - getReach(S,T,E)
    - getNHop(S,T,E)

Categorical algebra

- Category operators:
  - Sets and Functions to Category:
    - $Cat(S_1,\ldots,S_n, f_1 : S_{i1} \rightarrow S_{j1} ,\ldots, f_m : S_{im} \rightarrow S_{jm})$
    - This operator, called **Categorification**, constructs a category using a given set of objects and morphisms.
  - Category to Set
    - Limit which converts a category into a relational object (set).
    - $Lim(Cat(S_1,\ldots,S_n, f_1 : S_{i1} \rightarrow S_{j1} ,\ldots, f_m : S_{im} \rightarrow S_{jm}))$

Example of categorical algebra: Selection



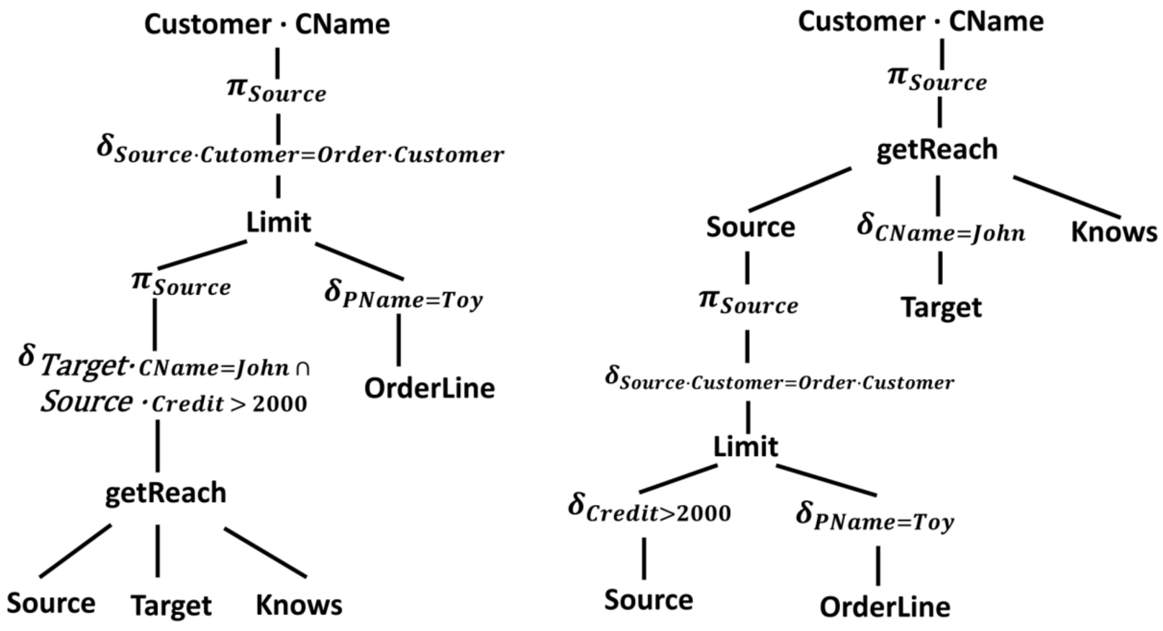
Query: Find all courses taken by ”Smith”

S1:= $\sigma_{\text{student} \cdot \text{Last\_name}=\text{“Smith”}}$ (Registration)

S2:= S1 ·Course ·Title

Return S2

Two examples of query plan with categorical algebra



(a) A query plan tree

(b) An optimized query plan tree that pushes down the limit and select operators

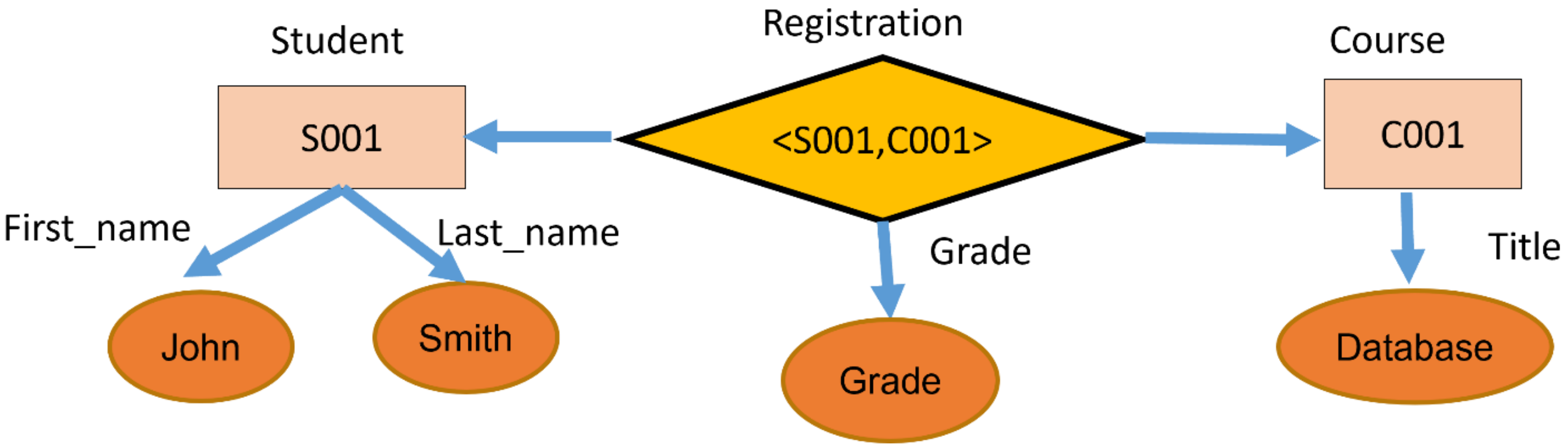
Figure 4: Two holistic query plans involving three types of data

Categorical calculus

- Categorical calculus, a **declarative** language for describing results in the category; Categorical algebra, a **procedural** language for listing operations in the category.
- The formulae of the Categorical Calculus

Formulae with range terms	Safe variables
$x_1 \in O_1$	x_1
$x_1 \in O_1 \wedge \neg(x_1 \in O_2)$	x_1
Formulae with function and range terms	**Safe variables**
$((f_1 : x_1 \rightarrow x_2) = f_2 \circ g_1) \wedge (x_1 \in S_1) \wedge (x_2 \in S_2)$	x_1, x_2
$(\pi_1 : (x_1, x_2) \rightarrow x_1) \wedge (x_1 \in S_1) \wedge (x_2 \in S_2)$	x_1, x_2
Formulae with predicate, range and function terms	**Data model**
$(x_1 \in S_1) \wedge (x_2 \in S_2) \wedge (x_1 \rightsquigarrow^E x_2) \wedge (x_1 \cdot \text{Name} = \text{"John"})$	Graph
$(x_1 \in D_1) \wedge (x_2 \in D_2) \wedge (x_1 \ isAncestor \ x_2)$	Tree
Formulae with unsafe terms	**Unsafe variable**
$x_2 \in S_2, x_3 \in S_3, \exists x_1 (x_1 > x_3 \wedge x_2 = 6)$	x_1
$\forall x_1 \exists x_2 \in S_2 (x_1 > x_2)$	x_1
$(x_1 \in S_1) \vee f(x_1) = a_1$	x_1

Categorical calculus and categorical algebra are equivalent (I)



Query: Find all courses taken by "Smith"

S1:= $\sigma_{\text{student} \cdot \text{Last\_name}=\text{“Smith”}}$ (Registration)

S2:= S1 ·Course ·Title

Equivalent calculus:

$\{\, x \mid x \in Title, \exists y \in Registration, y \cdot Student \cdot Last\_Name = \text{"Smith"} \wedge y \cdot \text{Course} \cdot \text{Title} = x \}$

Ongoing work

- Extend the categorical framework from multi-model data to multi-modality data.

- Include the vector data in the categorical framework.

- More details:
- Jiaheng Lu, A Categorical Unification for Multi-Model Data: Categorical Algebra and Calculus. International Conference on Applied Category Theory, 2025
- Link: https://arxiv.org/abs/2504.09515

Open Challenges and Conclusion

Handling Rare or Underrepresented Modalities

- Most multi-modal models are trained on dominant modalities like text and images, with less focus on audio or sensor data.
- Biased embeddings that underperform for underrepresented modalities.
- Research into inclusive datasets and techniques for balanced representation learning is needed.

Interpretability and Explainability

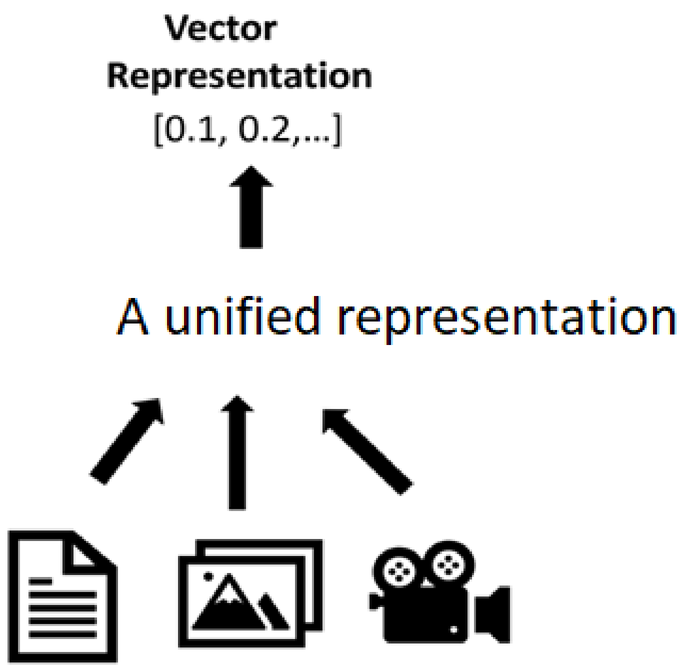
- Multi-modal embeddings are often black-box representations, making it hard to understand how different modalities contribute to the final embedding.
- Developing interpretable models that explain cross-modal interactions is crucial for trust and adoption in sensitive domains like healthcare.

Few-Shot and Zero-Shot Learning

- Enabling multi-modal embeddings to perform well in few-shot or zero-shot settings, where limited or no paired multi-modal data is available, is a significant challenge.
- Current models often rely on large paired datasets, limiting their flexibility in low-resource scenarios.
-

A Unified Theory Framework

- We propose a categorical theorical framework for multi-model. How to extend it to multi-modal data.
- It can unify different categories of methods under one framework.



Conclusion

- Vector presentations are used for multi-modal data processing
- Intra-modal representation (text, image, audio, video, time-series embedding) and inter-modal representation
- Vector databases for multi-modal data
- A categorical framework for multi-model databases

References

- Gong & Cosma (2022). Improving visual-semantic embeddings by learning semantically-enhanced hard negatives for cross-modal information retrieval. *Pattern Recognition 137*.
- Taipalus (2024). Vector database management systems: Fundamental concepts, use-cases, and current challenges. *Cognitive Systems Research 85*.
- Zhang, Yin, Chen & Nicele (2020**).** Emotion recognition using multi-modal data and machine learning techniques: A tutorial and review. *Information Fusion 59*.
- Jiaheng Lu (2025**)**, A Categorical Unification for Multi-Model Data: Categorical Algebra and Calculus. International Conference on Applied Category Theory, 2025