Vector Representations of Multi-modal Data

Toni Taipalus[1](✉) and Jiaheng Lu[2]

[1] Tampere University, Tampere, Finland
toni.taipalus@tuni.fi
[2] University of Helsinki, Helsinki, Finland
jiaheng.lu@helsinki.fi

Abstract. Multi-modal data processing is about exploring the interactions between various types of data to produce a more comprehensive or accurate understanding of a phenomenon such as health, emotions, or circumstances. Vectors as data representation methods have emerged as an important component in modern data management, driven by the growing importance for the need to computationally describe multi-modal data such as texts, images and video in various domains. In this tutorial, we provide a fundamental introduction on vector representations of multi-modal data, which includes intra-modal representation and inter-modal representation. The goal of our tutorial is to provide a centralized and condensed introduction regarding theories and applications of multi-modal data vectorization technologies for both database researchers and practitioners. We also discuss how to use vector database management systems for the management of multi-modal data.

Keywords: multi-modality · vector databases · vectorization · database · data management

1 Introduction

Multi-modality and vector databases have both gained substantial momentum in both research and practice [13, 21]. Multi-modality means several modalities. These modalities can include verbal (such as spoken language), visual (such as images or videos), auditory (such as sounds, noises or music), and spatial (such as physical space or arrangement), and efficiently utilizing multi-modality can achieve, for example, more accurate recommender systems [23] and richer medical diagnoses [1]. Vectorization of data, on the other hand, has the potential for more efficient storage, as well as representing data objects of different nature with similar terms, i.e., vectors. By combining the advances in the fields of both multi-modality data and vector databases, managing multi-modality data can be fundamentally more efficient, and reach new use-cases in various multidisciplinary domains.

Tutorial Overview: In this tutorial, we will discuss five modals of data: *text*, *image*, *audio*, *video* and *time-series* data. For each modal, we will present their

P. K. Chrysanthis et al. (Eds.): ADBIS 2025, LNCS 16043, pp. 272–279, 2026.
https://doi.org/10.1007/978-3-032-05281-0_18

respective vector representation method, and review cases for the combination of multiple modal data, including text+image, text+audio, image+audio, and text+image+audio. For each combination of modals, we will review the algorithms for vector alignment and representation fusion. In this tutorial, we will also show a fundamental introduction to applications of using multi-modality data with vector databases. Our tutorial discusses the basics behind vector database management systems and the process and unification of vectorization, as well as open problems and future opportunities of the intersection of multi-modality and vector databases.

The learning outcomes of this tutorial are to *(i)* understand the need behind vectors and multi-modality data management in today's landscape, *(ii)* understand how different data objects can be vectorized and why vectorization is useful for multi-modal data, *(iii)* learn different methods to unify and align vector representations for multi-modal data, and *(iv)* know of the current challenges and opportunities in the intersection of multi-modality and vector databases.

Intended Audience: This tutorial is designed for a broad audience, including academic researchers, students, industrial developers, and practitioners, who seek to understand vector representations for multi-modal data, explore how the convergence of vector databases and multi-modality enhances data management, and examine current challenges and future opportunities in inter-modality vectorization. A foundational understanding of databases and machine learning concepts is required to follow the tutorial effectively.

Related Tutorials: We are aware of tutorials which are tangential to this work. A SIGMOD'24 tutorial [15] gives a general survey on vector database management systems and vectorization techniques. A RecSys'21 tutorial [23] shows how multi-modality can relieve the challenge of sparsity of vectors in recommender systems. Finally, a 2020 *Information Fusion* tutorial [25] discusses multi-modality emotion recognition using a medical dataset.

Contributions: To the best of our knowledge, this is the first tutorial to discuss the methods of vectorization and use-cases of multi-modal data processing through vector databases. While previous tutorials have described both vector database management systems and multi-modality separately, this intersection has received no scientific tutorials, especially from a general perspective. The topic of vector databases and multi-modality covered in this tutorial may help industry professionals, researchers, educators, and students in understanding how vector databases can be used in multi-modality use-cases now and in the future, and what challenges this intersection currently faces.

2 Intra-modal Vector Representations

Vectors can be used to represent various types of data, including text, audio, video, images and time-series data. By vectorizing the features of data objects such as images, various features are given numerical representations. Vectors created from various data objects can be relatively efficiently stored and compared

[21]. It is highly context-dependent on which features of data objects are vectorized, as vectorizing all features is rarely feasible or even possible. A vectorized data object is often called a *feature vector* or a *vector embedding.*

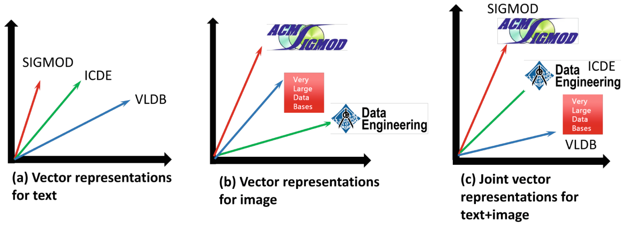
Text: Feature vectors have been used to enhance search accuracy by understanding synonyms and contextual relevance [16], in classifying documents into predefined categories for spam detection and sentiment analysis [5], for identifying and classifying entities within texts using contextual embeddings, for improving machine translation quality by representing words and phrases in a continuous vector space, and for grouping similar documents together using their vector representations for more accurate information retrieval [18].

Image: Vector representations generated by convolutional neural networks (CNNs) have been utilized in finding similar images in a database and for classifying images into categories by using the CNN features as inputs to a classifier [11]. Furthermore, vectors can be used in identifying objects within images using embeddings to represent features, and for generating descriptive captions for images by combining image embeddings with language models [6].

Audio: Vectors can be used in speech recognition by simply converting speech to text using audio embeddings such as Speech2Vec [2]. Vectors can also be used in speaker identification based on their unique voice characteristics captured in embeddings [9], song recognition, sound classification [10], and music recommendations.

Video: Converting a video into a vector representation involves extracting meaningful features from selected frames and encoding them as vectors. This process is useful for various tasks such as video classification and understanding. For example VideoBERT [19] passes each frame through a pre-trained CNN to obtain a feature vector from a fully connected or global pooling layer. The final result is a single vector representing the video.

Time-Series: Time series data refers to a sequence of data points collected at successive points in time, typically at regular intervals. Converting time series data into a vector representation involves various feature extraction techniques including statistical methods, wavelets, and embedding-based approaches [8].



(a) Vector representations for text

(b) Vector representations for image

(c) Joint vector representations for text+image

Fig. 1. Representations alignment for text and image

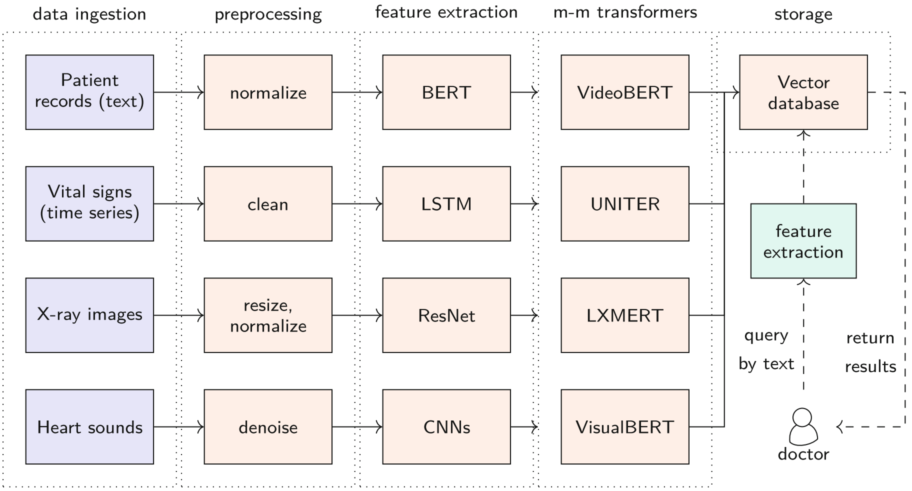


Fig. 2. A general process of storing multi-modality data into a common vector space, and into a vector database management system with an example of healthcare data; illustrated software, libraries, and techniques such as BERT and UNITER are examples; multi-modality transformers (abbreviated *m-m transformers* above) are used to ensure that multi-modality data objects are stored into a common vector space, which can subsequently be queried by a doctor using only textual input

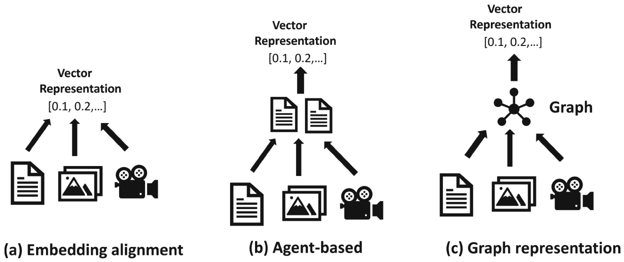

Vector Representation [0.1, 0.2,...]
(a) Embedding alignment
Vector Representation [0.1, 0.2,...]
(b) Agent-based
Vector Representation [0.1, 0.2,...]
Graph
(c) Graph representation

Fig. 3. Various approaches for unifying inter-modality vector representations

3 Inter-modal Vector Representations

In inter-modality contexts, vector representations can be used in tandem with multiple modals. In this section, we describe the cross-modal alignment between the embedding spaces of multiple modalities (such as text+image, cf. Fig. 1) learned from corpora of their respective modalities.

Text and Image: Image captioning inherently combines visual data with textual data. Visual features extracted from images and the sequential nature of text generation can be integrated into a unified model. Furthermore, this enables cross-modal retrieval, as querying by text can retrieve images and vice versa, as both text and images are embedded to a common embedding space [4]. Additionally, this approach enables answering questions about images when the cross-modal embeddings are passed to a language model.

Text and Audio: By aligning audio embeddings with text embeddings, speech can be converted to text and vice versa. This allows, for example, automated summarization of long speeches such as podcasts. Cross-modal alignment between embedding spaces of speech and text learned from corpora of their respective modalities in an unsupervised fashion has been of interest in scientific works [3]. The proposed framework learns the individual speech and text embedding spaces, and attempts to align the two spaces via adversarial training and a refinement procedure.

Image and Audio: For audio-image vector representations, several CNN-based feature extractors have been proposed, including *EfficientNet* [22] and *Inception ResNet* [20]. By aligning audio and visual content, vector embedding can be used in audiovisual synchronization, for example for synchronizing lip movements with speech. Multi-modality through image and audio can also be used to enhance emotion recognition by considering both facial or bodily expressions as well as audio cues [25].

Text, Image and Audio: The combination of text, image, and audio data into a common embedding space enables multi-modal search engines which can retrieve results across several types of data objects. Similar embeddings can also be used in multi-modal recommender systems [23]. This allows content recommendations which are not based on data objects of similar type. That is, the recommender system may suggest, e.g., text documents based on videos watched, and these recommendations are not limited to metadata, but to analyzed (and vectorized) content. Additionally, inter-modality with feature vectors can serves as a basis for both inputs and outputs of digital assistants. Fig. 2 illustrates a general example use-case and example techniques for storing multi-modal data objects into a vector database, using a common vector space.

4 Methods for Unifying Inter-modality Vectors

In this section, we introduce basic approaches on how to unify different vector representations for multi-modal data to integrate and reconcile vectors from different modalities with a representation that can capture the essence of several modalities simultaneously. Figure 3 illustrates various approaches to produce a unified vector representation.

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. This is typically achieved using a contrastive learning objective or a triplet loss function [24]. For example, Fig. 1 illustrates the vector alignment for text and image data in a joint space.

Agent-Based Alignment: Another method of dealing with multi-modality data is simply 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, as the vectorization process is only applied to text. We call this an agent-based alignment, rooted in the presumption that different models (or *agents*) are responsible for creating feature vectors for different data objects. **-to-text* models such as CLIP [17] with GPT for image-to-text, BART [12] for text-to-text (e.g., summarizing and translation), and *Whisper* for audio-to-text have already shown substantial results in their respective tasks.

Fusion Techniques: These techniques include early fusion and late fusion [26]. Early fusion involves concatenating or stacking vectors from different modalities into a single vector representation before inputting them into a neural network, which outputs the final vector to be stored in a vector database. These vectors can be high-dimensional, making it more difficult for the model to learn efficiently. In contrast, late fusion combines the outputs from models trained separately on each modality, which can involve averaging or weighting predictions from the individual models to generate a final representation. The challenges in late fusion are typically related to relatively low coupling between the modalities.

Graph Representation: Construct a graph where nodes represent instances and edges capture relationships or similarities between instances across different modalities [14]. Graph neural networks can then be applied to learn a unified representation by aggregating information from the graph.

5 Vector Databases for Multi-modal Data

After relevant data objects have been vectorized, these feature vectors can be stored for subsequent use by, e.g., customer-facing applications. These applications, such as the healthcare-related systems described in Fig. 2, may either use libraries, a dedicated vector DBMS, or a more general-purpose DBMS to index and query the vectors. Each of these approaches relieve the need to implement, e.g., vector indexing techniques such as *product quantization* or *hierarchical navigable small world*, and have distinct use-cases.

Vector management-dedicated software libraries such as FAISS [7] typically offer different ways of indexing vectors and querying them with similarity search. Typically, this approach is often computationally fast, yet provides little more for effective data management [21]. Dedicated vector DBMSs such as Milvus or Chroma, on the other hand, have been built from the ground up to serve the needs of vector data management. This relatively novel approach typically supports several vector index types, and strives for interoperability between the DBMS and other system components. Finally, more general DBMSs such as PostgreSQL, SQLite, MongoDB, Cassandra, and Redis also support vector data management, some through third party extensions. This approach potentially adds data management features such as concurrency control, access control, ACID-compliance, back-up and recovery, and an expressive query language to support vector data management. For example, with PostgreSQL's `pgvector` extension, a software developer may query vectors using SQL, as well as utilize vector metadata in queries. For example, a query with a query vector can be complemented with more traditional SQL expressions.

References

1. Cheng, D., Liu, M.: CNNs based multi-modality classification for ad diagnosis. In: 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pp. 1–5 (2017). https://doi.org/10.1109/CISP-BMEI.2017.8302281
2. Chung, Y., Glass, J.R.: Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech. In: Yegnanarayana, B. (ed.) 19th Annual Conference of the International Speech Communication Association. pp. 811–815. ISCA (2018https://doi.org/10.21437/INTERSPEECH.2018-2341
3. Chung, Y.A., Weng, W.H., Tong, S., Glass, J.: Towards unsupervised speech-to-text translation. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7170–7174. IEEE (2019)
4. Gong, Y., Cosma, G.: Improving visual-semantic embeddings by learning semantically-enhanced hard negatives for cross-modal information retrieval. Pattern Recogn. **137**, 109272 (2023). https://doi.org/10.1016/j.patcog.2022.109272
5. Guo, S., Yao, N.: Document vector extension for documents classification. IEEE Trans. Knowl. Data Eng. **33**(8), 3062–3074 (2021). https://doi.org/10.1109/TKDE.2019.2961343
6. Herdade, S., Kappeler, A., Boakye, K., Soares, J.: Image captioning: transforming objects into words. Adv. Neural Inf. Processing Syst. **32** (2019)
7. Johnson, J., Douze, M., Jegou, H.: Billion-scale similarity search with GPUs. IEEE Trans. Big Data **7**(3), 535–547 (2021). https://doi.org/10.1109/tbdata.2019.2921572
8. Kazemi, S.M., et al.: Time2vec: Learning a vector representation of time (2019). https://arxiv.org/abs/1907.05321
9. Kinnunen, T., Karpov, E., Franti, P.: Real-time speaker identification and verification. IEEE Trans. Audio Speech Lang. Process. **14**(1), 277–288 (2006). https://doi.org/10.1109/TSA.2005.853206
10. Ko, K., Park, S., Ko, H.: Convolutional feature vectors and support vector machine for animal sound classification. In: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 376–379 (2018). https://doi.org/10.1109/EMBC.2018.8512408
11. Kriegel, H.P., Brecheisen, S., Kröger, P., Pfeifle, M., Schubert, M.: Using sets of feature vectors for similarity search on voxelized CAD objects. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, pp. 587–598. SIGMOD '03, ACM (2003). https://doi.org/10.1145/872757.872828
12. Lewis, M.: Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension (2019). https://arxiv.org/abs/1910.13461
13. Lu, J., Holubová, I.: Multi-model databases: A new journey to handle the variety of data. ACM Comput. Surv. **52**(3) (2019). https://doi.org/10.1145/3323214

14. Lu, Y., Zhao, W., Sun, N., Wang, J.: Enhancing multimodal knowledge graph representation learning through triple contrastive learning. In: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, pp. 5963–5971. ijcai.org (2024)
15. Pan, J.J., Wang, J., Li, G.: Vector database management techniques and systems. In: Companion of the 2024 International Conference on Management of Data, pp. 597–604. ACM (2024)
16. Perrin, P., Petry, F.E.: Extraction and representation of contextual information for knowledge discovery in texts. Inf. Sci. **151**, 125–152 (2003)
17. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning, pp. 8748–8763. PMLR (2021)
18. Singh, S.P., Kumar, A., Darbari, H., Singh, L., Rastogi, A., Jain, S.: Machine translation using deep learning: An overview. In: 2017 International Conference on Computer, Communications and Electronics (Comptelix), pp. 162–167 (2017). https://doi.org/10.1109/COMPTELIX.2017.8003957
19. Sun, C., Myers, A., Vondrick, C., Murphy, K., Schmid, C.: VideoBERT: A joint model for video and language representation learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7464–7473 (2019)
20. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, Inception-ResNet and the impact of residual connections on learning. In: Singh, S., Markovitch, S. (eds.) Proceedings of the 31st AAAI Conference on Artificial Intelligence, pp. 4278–4284. AAAI Press (2017). https://doi.org/10.1609/AAAI.V31I1.11231
21. Taipalus, T.: Vector database management systems: Fundamental concepts, use-cases, and current challenges. Cogn. Syst. Res. **85**, 101216 (2024). https://doi.org/10.1016/j.cogsys.2024.101216
22. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. vol. 97. PMLR (2019)
23. Truong, Q.T., Salah, A., Lauw, H.: Multi-modal recommender systems: Hands-on exploration. In: Proceedings of the 15th ACM Conference on Recommender Systems, pp. 834–837. RecSys '21, ACM (2021). https://doi.org/10.1145/3460231.3473324
24. Yuan, X., et al.: Multimodal contrastive training for visual representation learning. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6995–7004. Computer Vision Foundation / IEEE (2021)
25. Zhang, J., Yin, Z., Chen, P., Nichele, S.: Emotion recognition using multi-modal data and machine learning techniques: A tutorial and review. Inf. Fusion **59**, 103–126 (2020). https://doi.org/10.1016/j.inffus.2020.01.011
26. Zhang, Y., Sidibé, D., Morel, O., Mériaudeau, F.: Deep multimodal fusion for semantic image segmentation: A survey. Image Vision Comput. **105**, 104042 (2021). https://doi.org/10.1016/j.imavis.2020.104042