Member-only story

GRAPHEDM-SERIES

Graph Representation Learning — Network Embeddings (Part 1)

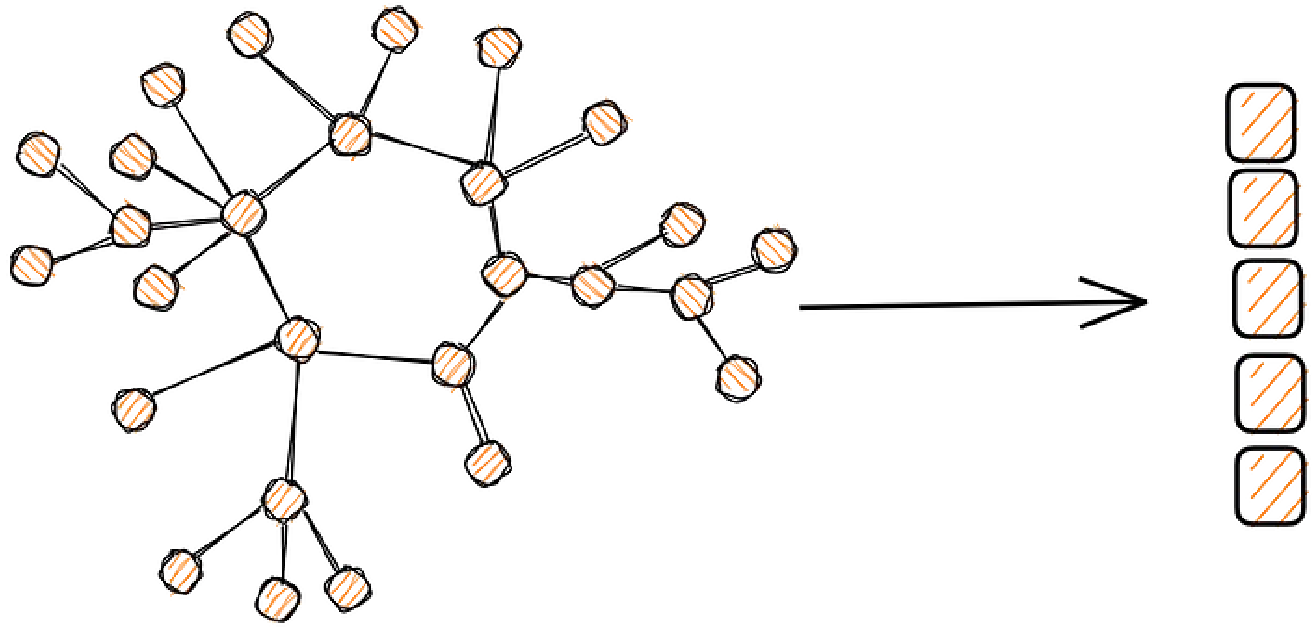
Warm-up on the main concepts

Giuseppe Futia · Follow

Published in Towards Data Science

7 min read · May 11, 2021

Listen Share More



Representation of node embeddings— Image from the Excalidraw library modified by the author

This series summarizes a comprehensive taxonomy for machine learning on graphs and reports details on GraphEDM (Chami et. al), a new framework for unifying different learning approaches

Graphs are universal structures to model relational data. From social network connections to protein interactions, graph structures emphasize the links between data points, enabling the development of a new generation of network-based systems. Considering the widespread of graphs in real scenarios and the recent success of representation learning techniques, there has been a tremendous proliferation of Graph Representation Learning (GRL) methods. As a consequence, given this impressive growth, there is a huge need to summarize and unify GRL methods in a comprehensible framework. Therefore, in this series I report my personal synthesis of the following paper:

> *Chami, I., Abu-El-Haija, S., Perozzi, B., Ré, C., & Murphy, K. (2020). Machine learning on graphs: A model and comprehensive taxonomy. arXiv preprint arXiv:2005.03675.*

This manuscript provides a solid foundation for understanding the main intuitions behind GRL methods. Moreover, the proposed taxonomy is particularly useful to achieve a broad overview of available applications and support the selection of the best tool for a given problem. In this new series, I will focus on the general idea of the *Graph Encoder Decoder Model* (*GraphEDM*), which is able to describe a broad range of supervised and unsupervised methods for learning representations of graph data (for details on the single approaches, you can directly refer to the paper).

The series is structured as follows:

1. *An overview of the embedding problem in the context of GRL (reported in this article).*
2. *A description of the GraphEDM framework, including a summary of its main components.*
3. *A detailed explanation of the objective (or loss) functions, the encoding approaches, and the supervised applications defined in the framework.*

Embeddings in Graph Representation Learning

The main goal of GRL methods is to learn low-dimensional continuous representations, known as *embeddings*, from discrete graph data. The properties of the graph have to be preserved in the embedding space: for instance, nodes characterized by similar connections in the original graph achieve a close vector representation at the end of the learning process (in a few lines, we will see how this is correct specifically for structural embeddings). From another perspective, node embedding learning can be viewed as a dimensionality reduction process useful for scalability purposes. Indeed, the dimension of the learned vectors is strictly lowered than the number of the nodes in the original graph.

The following subsections provide intuitive definitions in order to clearly state the problem setting of GRL. In particular, these subsections give a discussion on important differences between:

1. Euclidean vs non-Euclidean Geometry.

2. Positional vs Structural Embeddings.

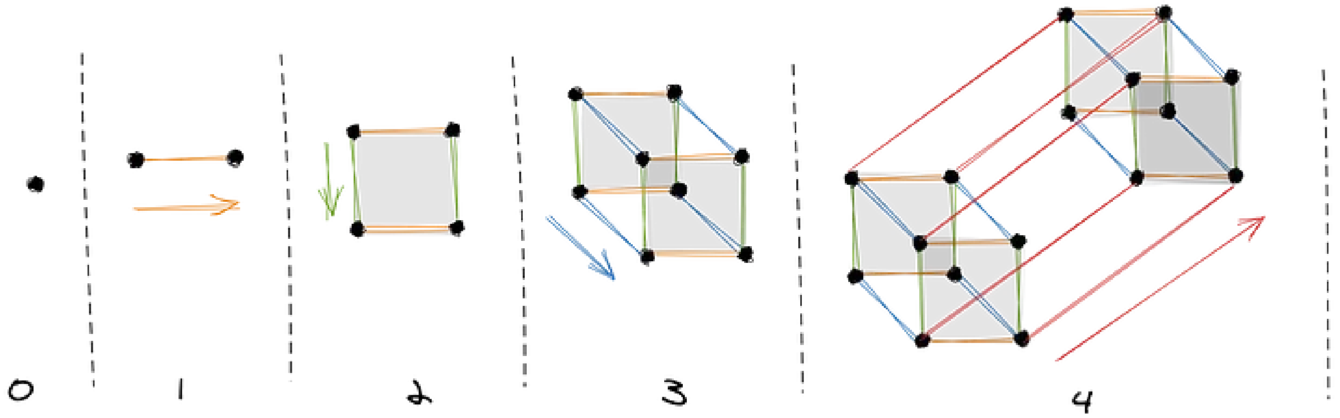
3. Transductive vs Inductive Learning.

4. Unsupervised vs Supervised Tasks.

Euclidean vs non-Euclidean Geometry

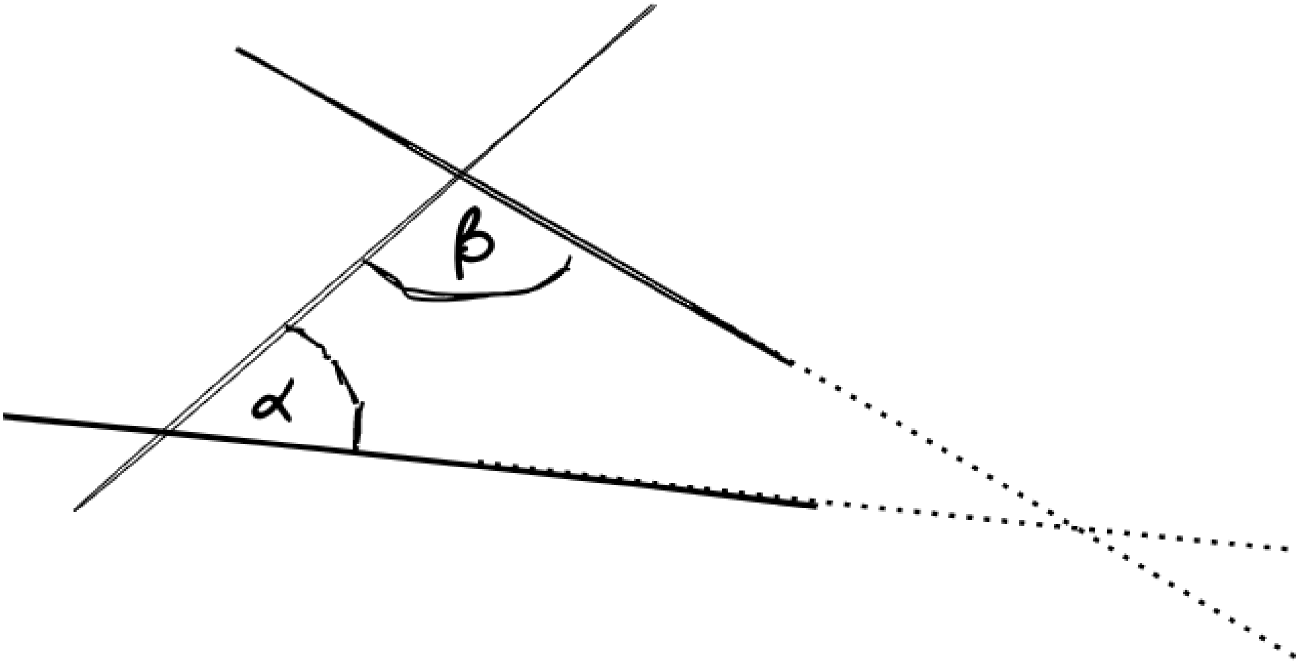
In most of the representation learning approaches, embeddings are modeled using vectors in the *Euclidean* space. For over 2000 years and until the 19th century, it was the only known conception of physical space and it still prevails as the most suitable manner for modeling the world as it is daily experienced. According to the Encyclopaedia Britannica, the Euclidean space is defined as:

> *a two- or three-dimensional space in which the axioms and postulates of Euclidean geometry apply; also a space in any finite number of dimensions, in which points are designated by coordinates (one for each dimension) and the distance between two points is given by a distance formula.*



From 0th to 4th-dimensional euclidean space — Image by the author

At the beginning of its definition, the Encyclopaedia Britannica explains that a Euclidean space satisfies by definition the axioms of the Euclidean geometry. One of the distinctive axioms of Euclidean geometry is the *parallel postulate*. For a clear intuition of this postulate, consider the following picture.



Graphical representation of the parallel postulate — Image by the author

In this picture, a line segment intersects two straight lines. This intersection forms two interior angles *α* and *β* on the same side, whose sum is less than the sum of two right angles (<180°). According to the postulate, if the two lines are extended indefinitely, they will meet at a certain point on the side of α and β. The parallel postulate can be defined through equivalent statements, which are more interesting for a better understanding of the network embedding representation. These statements include for instance the Playfair's axiom:

> ***there is at most one line*** *that can be drawn parallel to another given one through an external point*

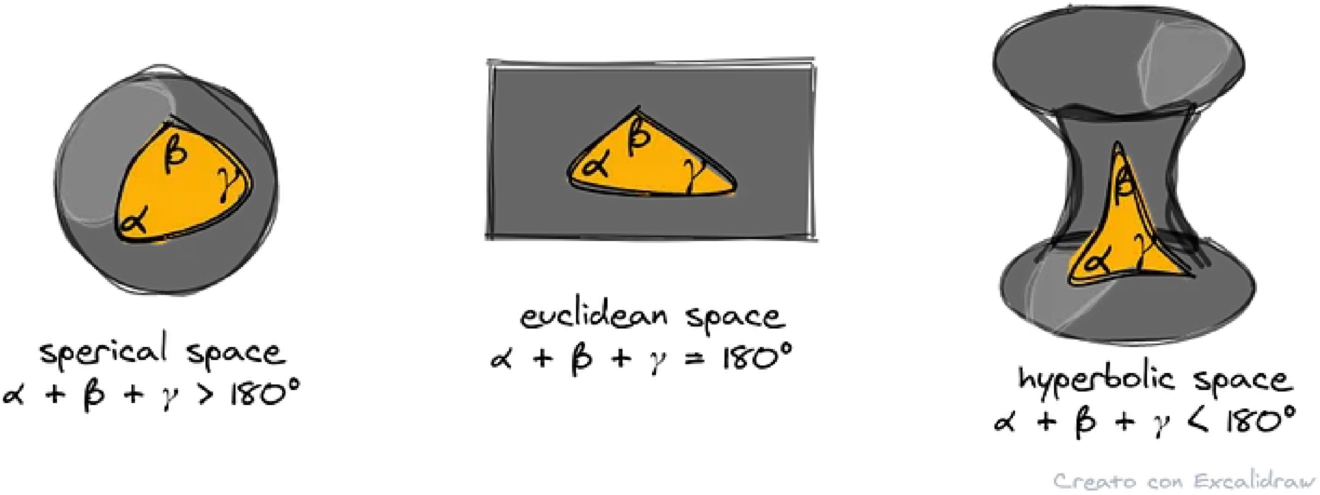
and the triangle postulate:

> *the sum of the angles in every triangle is 180°*

Although the Euclidean embeddings are particularly effective to represent grid-structured or sequential data, in the case of graphs there is a growing interest in *non-Euclidean* representation learning. In this case, the main goal is to learn embeddings in non-Euclidean spaces, such as the spherical and hyperbolic spaces. In non-Euclidean spaces, the parallel postulate and its equivalent statements do not hold: for instance, in the case of the hyperbolic space, the Playfair’s axiom is refuted because:

> ***there are at least two lines** that can be drawn parallel to another given one through an external point*

A clear intuition can be achieved by observing how the triangle postulate is not satisfied in non-Euclidean space:



sperical space
$\alpha + \beta + \gamma > 180^\circ$
euclidean space
$\alpha + \beta + \gamma = 180^\circ$
hyperbolic space
$\alpha + \beta + \gamma < 180^\circ$
Creato con Excalidraw

Graphical representation of triangle postulate in the spherical, euclidean, and hyperbolic spaces — Image by the author

In the context of GRL, non-Euclidean approaches seem to be very promising. The main idea behind these approaches is to map the graph data representation in a continuous embedding space, which is able to resemble the underlying graph structure of the input (e.g., hyperbolic spaces represent a continuous version of trees). For instance, a particular implementation of semantic networks known as Knowledge Graphs (KGs) exhibits hierarchical and logical patterns that can be preserved with high fidelity through hyperbolic embeddings. Further information on KGs are available in the following article and the related series:

Knowledge Graphs at a glance

Incorporate human knowledge into intelligent systems, exploiting a semantic graph perspective

towardsdatascience.com

Positional vs Structural Embeddings

GRL techniques aim at learning low-dimensional representations that preserve the structure of the input graph. Techniques such as matrix factorization or random walk tend to preserve the global structure, reconstructing the edges in the graph and maintaining distances such as the shortest paths in the original network. The goal of the methods focused on the global structure is to learn the so-called *positional embeddings.*

Other techniques, including Graph Neural Networks (GNNs), intend to capture the local graph structure: central nodes with similar neighbors in a graph should have similar embeddings, in spite of the distance of the nodes in the original network. The goal of these structure-aware methods is to learn the so-called *structural embedding.*

Transductive vs Inductive Learning

A specific way to classify embedding methods is related to their generalization capability for unseen data. A transductive setting assumes that all graph nodes are observed during the training process: the requirement of this setting is a fixed graph, which does not consider new instances in the form of new nodes, edges, or subgraph structures. Transductive methods allow us to infer new information between the nodes analyzed during the training process. For instance, given partially labeled nodes, we can classify unlabeled nodes. Or, in other cases, we can predict new edges between the graph nodes observed during the training.

In the inductive setting, we expect that the model is capable to generalize to nodes and edges that were not seen during the training process. Therefore, inductive learning can be particularly suitable for dynamic and temporally evolving graphs. Node features take a crucial role in inductive graph representation learning methods. Indeed, unlike the transductive approaches, these features can be employed to learn embedding with parametric mappings. The learning goal is achieved by optimizing such parametric mappings instead of directly optimizing the embeddings. This implies that the learning mappings can be applied to any node, even those that were not seen during the training process.

Unsupervised vs Supervised Tasks

In *unsupervised* tasks, the graph structure is the only available information. Therefore, the goal of the embedding learning procedure is to preserve the graph structure, optimizing a reconstruction loss. This loss function measures how well the learned embeddings can approximate the original graph.

In supervised tasks, additional information on node or graph labels is provided. In these cases, the embedding learning procedure aims at addressing a downstream task, including the prediction of node, edge, or graph attributes. Therefore, the model is optimized in order to reach this specific purpose. As I have already reported in the following article, in the case of GNNs, you can add a prediction layer to the network architecture in order to reach your learning goal, from node classification to link prediction.

What's Next

In the next article of this series, I will provide details on GraphEDM, a general framework that encapsulates unsupervised and supervised methods, graph regularization, and GNNs.

For all the articles on the GraphEDM framework, you can use the following link: https://towardsdatascience.com/tagged/graphedm-series.

For further readings on Graph Representation Learning, you can follow my series at the following link: https://towardsdatascience.com/tagged/grl-series.

If you like my articles, you can support me using this link https://medium.com/@giuseppefutia/membership and becoming a Medium member.

Graphedm Series   Grl Series   Graph Neural Networks   Graph Learning

Editors Pick

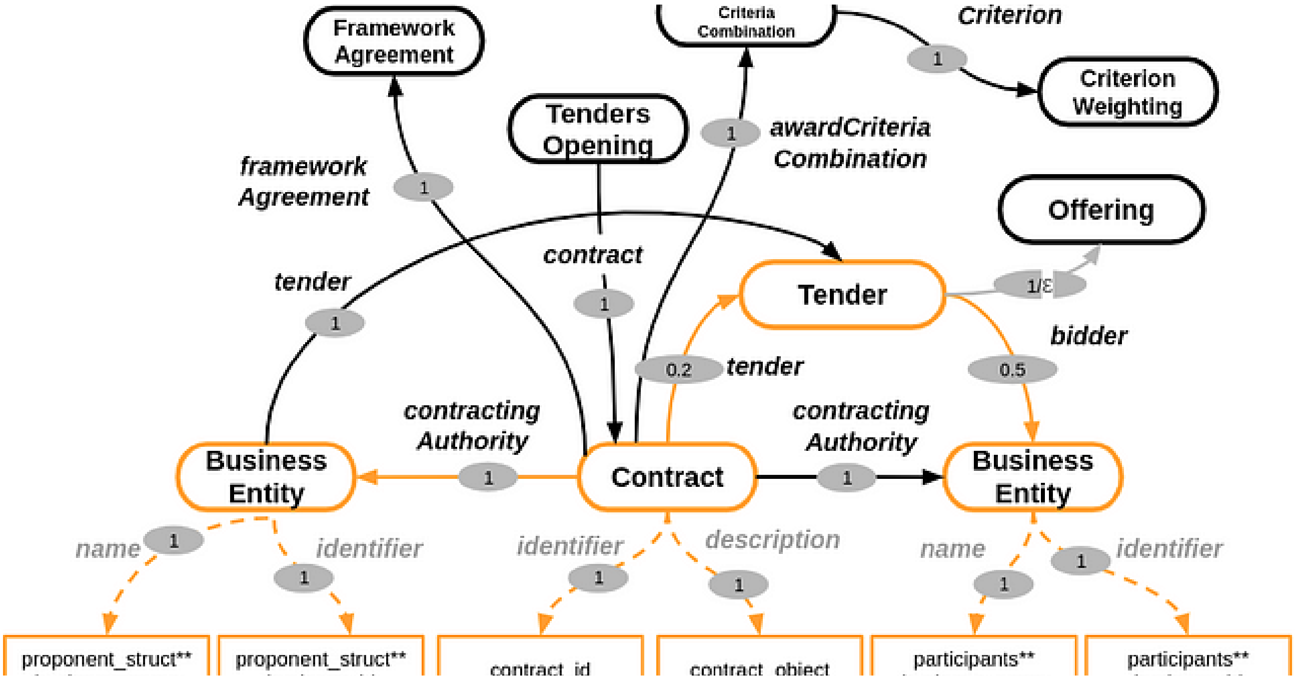
Follow

Written by Giuseppe Futia

768 Followers · Writer for Towards Data Science

Senior Data Scientist at GraphAware | Ph.D. at Politecnico di Torino. Passionate about Knowledge Graphs, Semantic Modeling, and Graph Neural Networks.

More from Giuseppe Futia and Towards Data Science



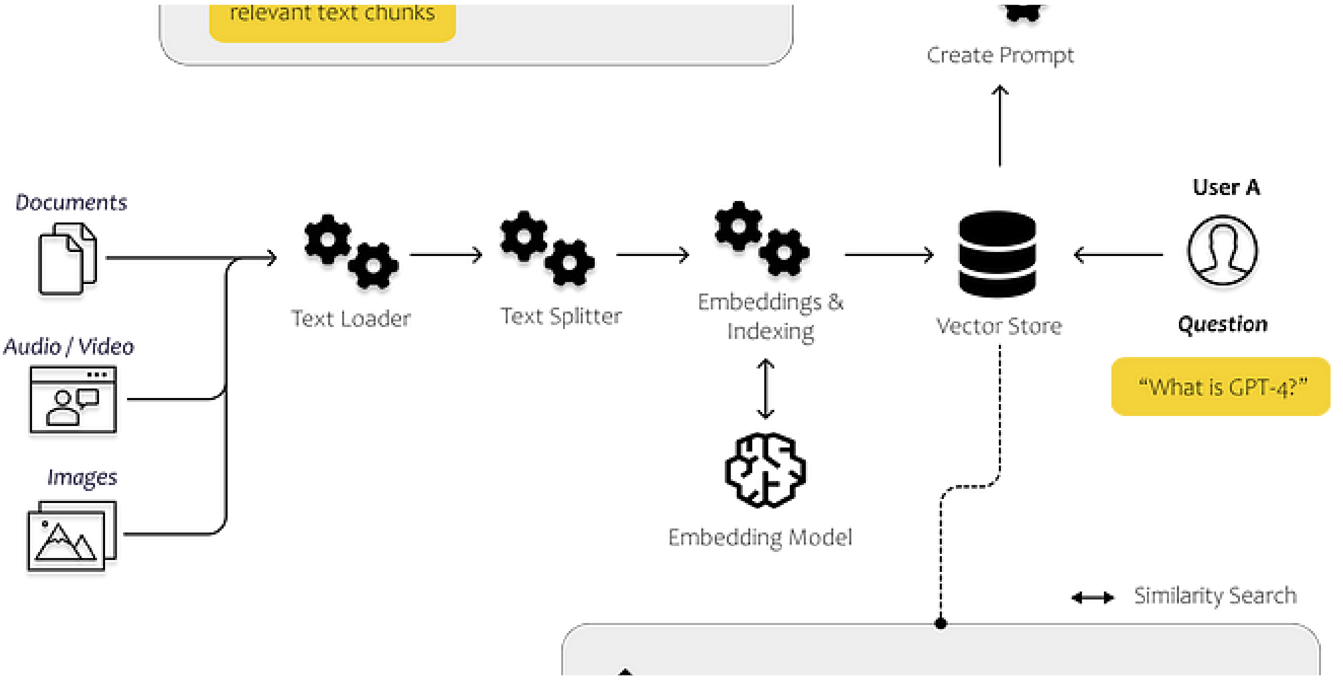
Giuseppe Futia in Towards Data Science

Semantic Models for Constructing Knowledge Graphs

Capturing the Semantics of Data Sources with Graph-Based Structures

✦ · 6 min read · Dec 16, 2020

69



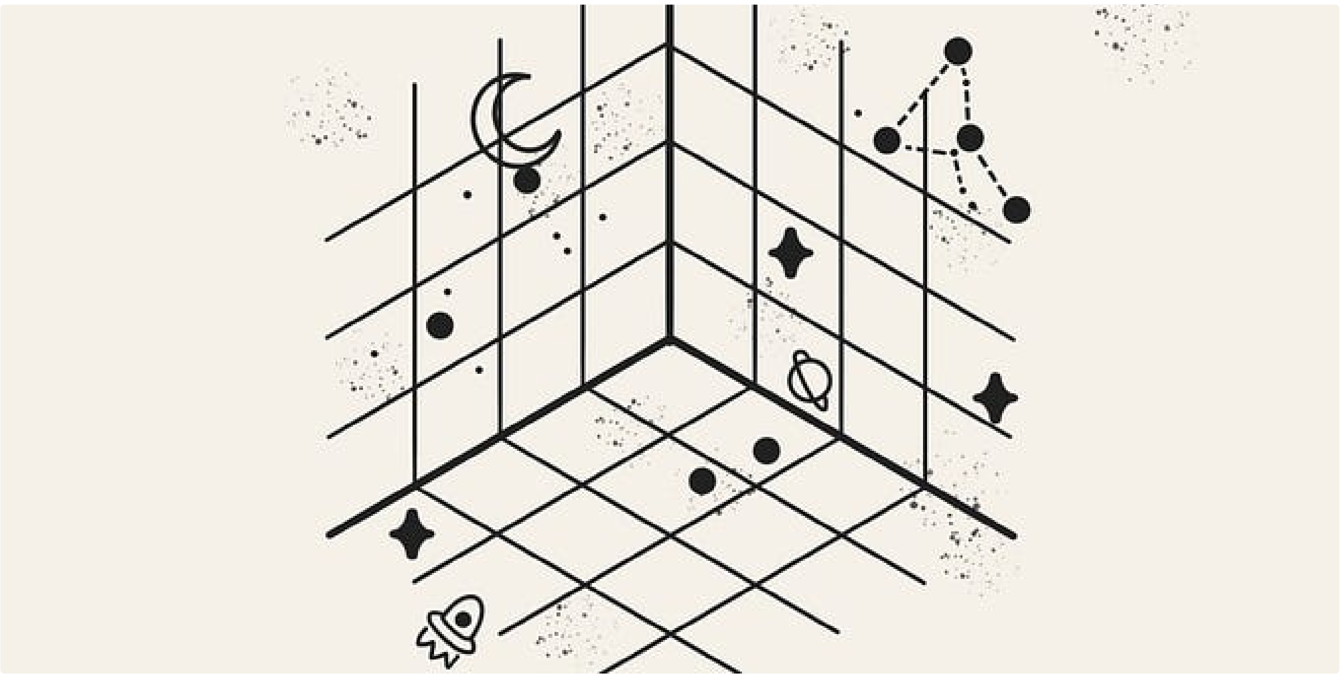
Dominik Polzer in Towards Data Science

All You Need to Know to Build Your First LLM App

A step-by-step tutorial to document loaders, embeddings, vector stores and prompt templates

✦ · 26 min read · Jun 22

3.4K 31



Leonie Monigatti in Towards Data Science

Explaining Vector Databases in 3 Levels of Difficulty

From noob to expert: Demystifying vector databases across different backgrounds

✦ · 8 min read · Jul 4

1.8K 18

ARISTOTLE
ATHENS
PLATO
INFLUENCED
INFLUENCED
BORN IN
IS LOCATED IN
IS A
PARTHENON
SOCRATES
IS A
IS A
PHILOSOPHER
IS A

Giuseppe Futia in Towards Data Science

Knowledge Graphs at a glance

Incorporate human knowledge into intelligent systems, exploiting a semantic graph perspective

✦ · 7 min read · Sep 27, 2020

182 1

See all from Giuseppe Futia

See all from Towards Data Science