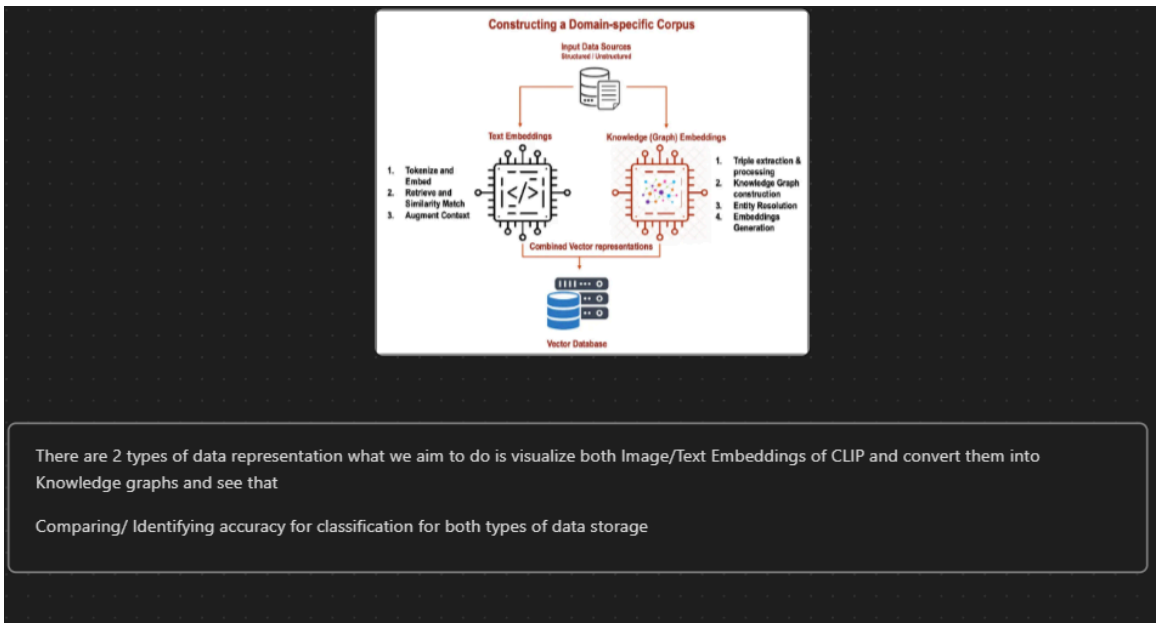
Doc1

Embedding Methods visual and accuracy comparison, visualization on CLIP or other multimodal model architectures

- https://www.kineviz.com/sightxr
- https://freedium.cfd/https://towardsdatascience.com/combine-text-embeddings-and-knowledge-graph-embeddings-in-rag-systems-5e6d7e493925
- https://ohie.org/framework/
- https://networkx.org/
- https://towardsdatascience.com/knowledge-graph-embeddings-101-2cc1ca5db44f
- https://www.youtube.com/watch?v=NXHIj1II5rE



The concept of Text Embeddings and Knowledge Graphs Embeddings

FIGURE 1 | The scene[1].

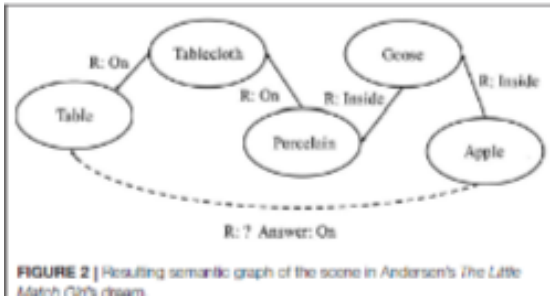
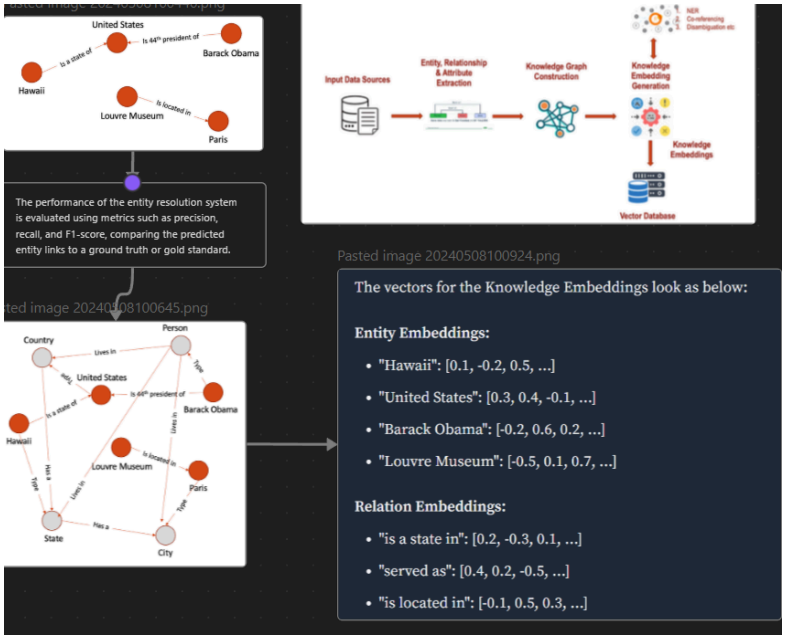
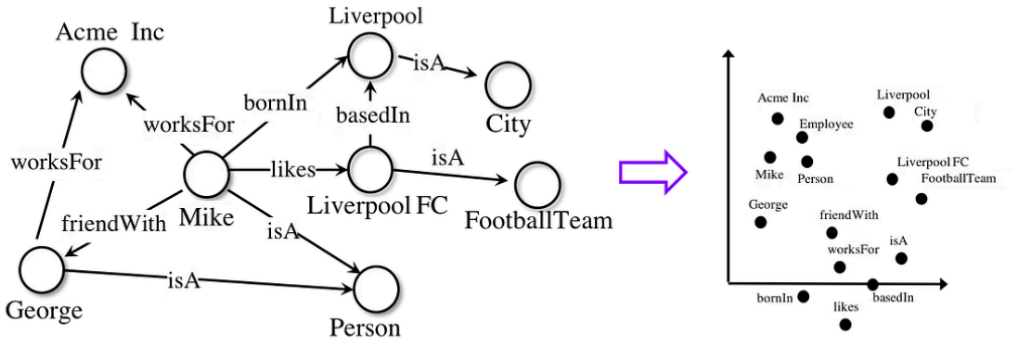


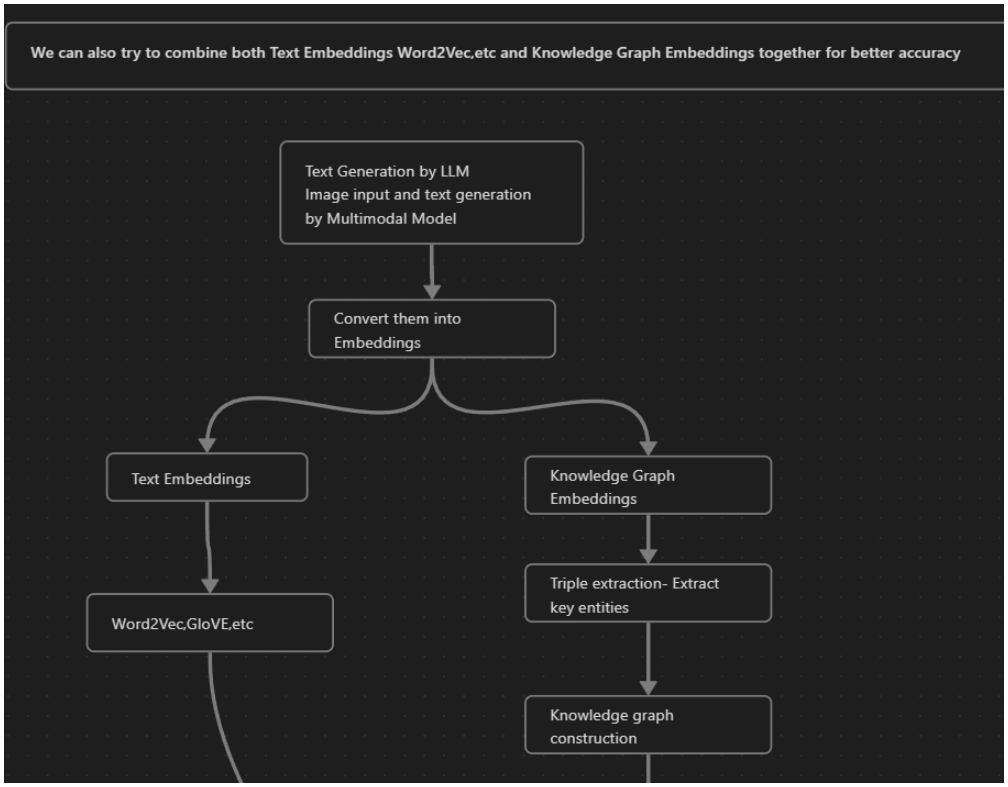
FIGURE 2 | Resulting semantic graph of the scene in Anderson's *The Little Match Girl's dream*.

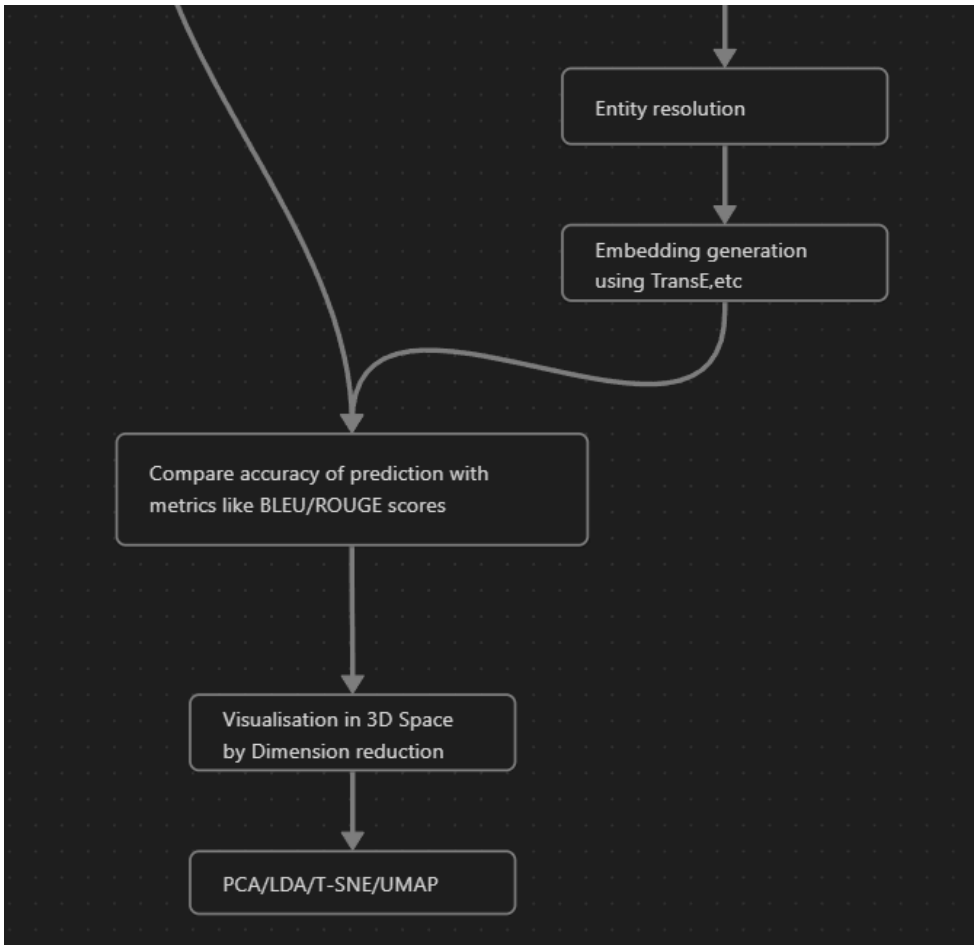


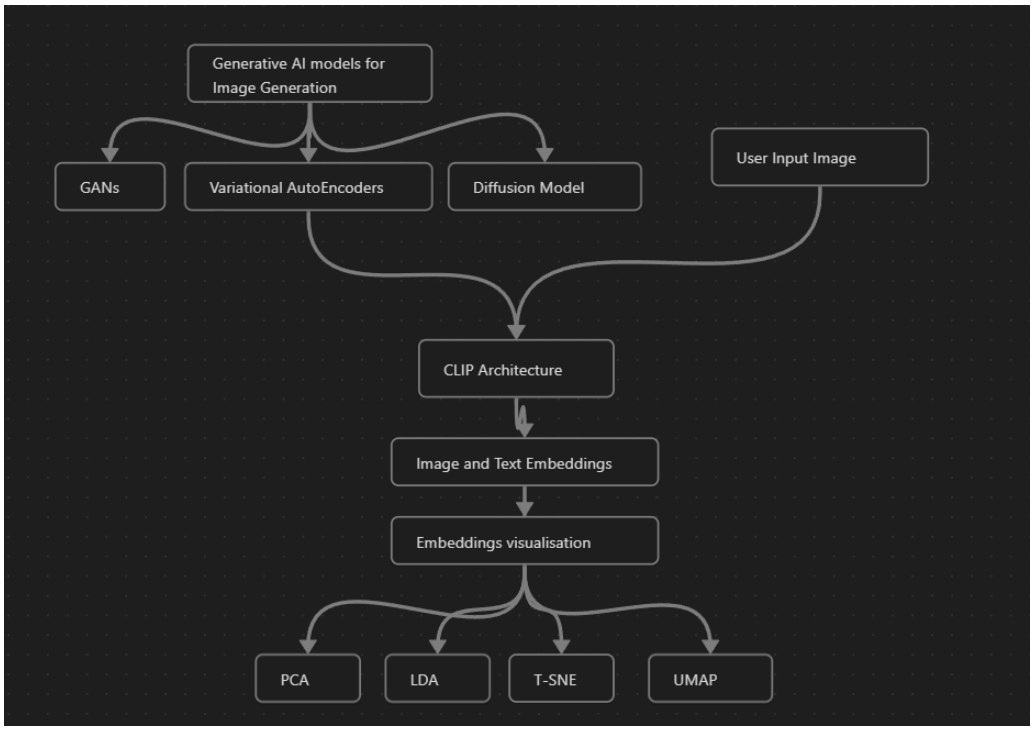


The figure above shows embedding representations of nodes and relations projected in a 2D vectorial space. Image source: https://docs.ampligraph.org/en/1.1.0/. Ampligraph is a free open source python library for KGE.

Rough Flow of project



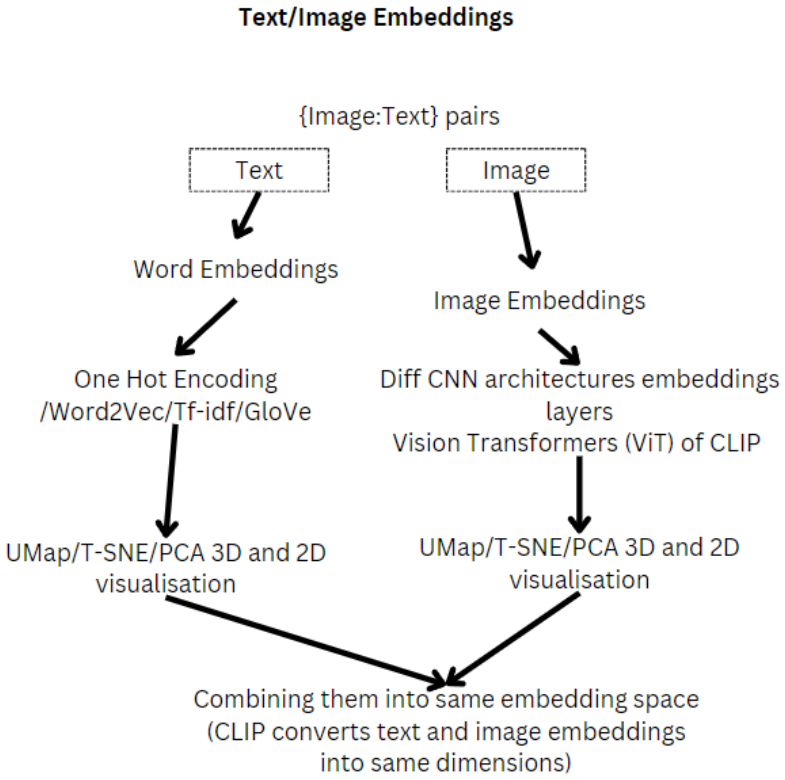

Entity resolution
Embedding generation using TransE,etc
Compare accuracy of prediction with metrics like BLEU/ROUGE scores
Visualisation in 3D Space by Dimension reduction
PCA/LDA/T-SNE/UMAP


Generative AI models for Image Generation
GANs
Variational AutoEncoders
Diffusion Model
User Input Image
CLIP Architecture
Image and Text Embeddings
Embeddings visualisation
PCA
LDA
T-SNE
UMAP

Text-Image Embeddings & Knowledge Graph Embeddings

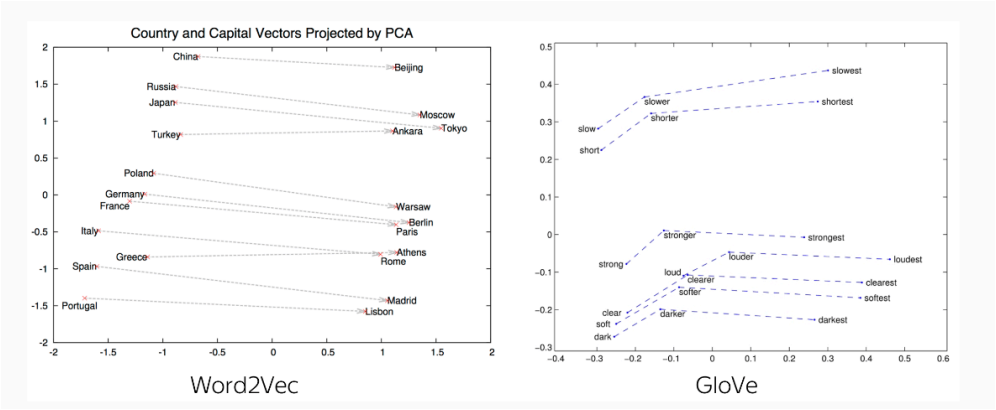
The project will be divided into 2 parts:

1. Text-Image Vector Embeddings
2. Knowledge Graph Embeddings (KGEs)

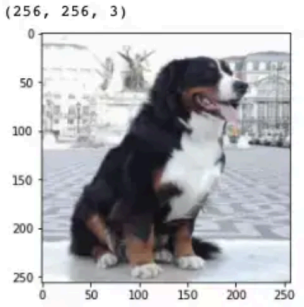


Prerequisites (for Part 1):

- **Understanding Word and Image embeddings + Basics of NLP**
  https://lena-voita.github.io/nlp_course/word_embeddings.html -Word Embeddings
  1. Understand Context Windows,
  2. Word2Vec ka Log loss function,
  3. Embedding dimensions.
  4. Read analysis and interpretability section
  5. Linear Structure

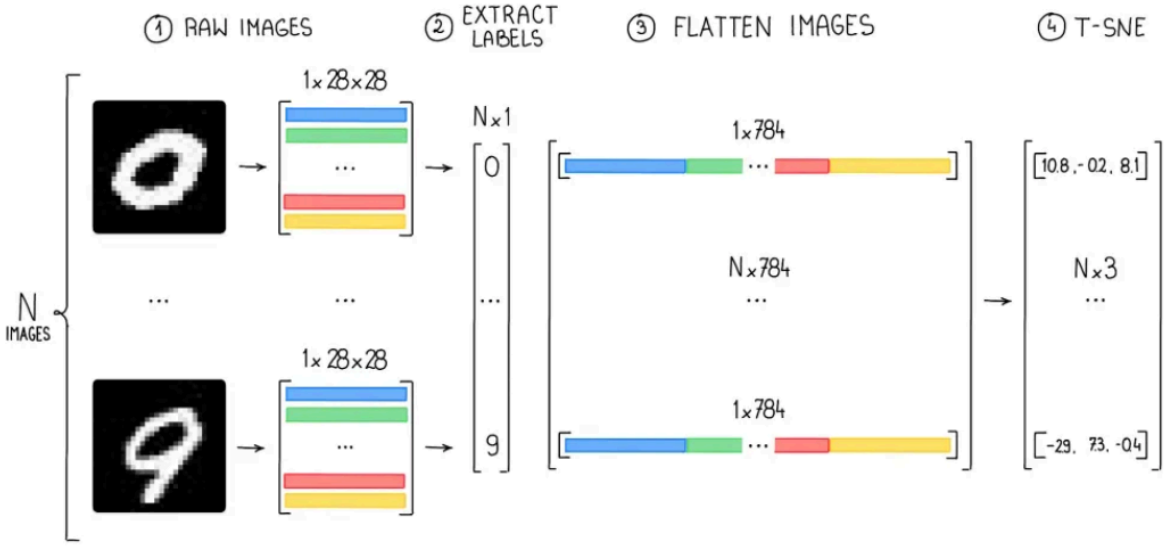


- **Image Embeddings of CLIP model and CNNs:** https://www.activeloop.ai/resources/generate-image-embeddings-using-a-pre-trained-cnn-and-store-them-in-hub/




```
[0.13786289 2.39912081 0.96784693 0.1817003  0.87487513 1.32822526
 0.57953429 2.90846252 0.24717608 1.58063555]
```

Each CNN model has different embedding layers for Image, we can use them for visualization.



This is any CNN model embedding

- What is an Image Embedding?
- Leveraging Embeddings and Clustering Techniques in Computer Vision

This RoboFlow blog discusses about CLIP and Large Multimodal Model's Image embeddings

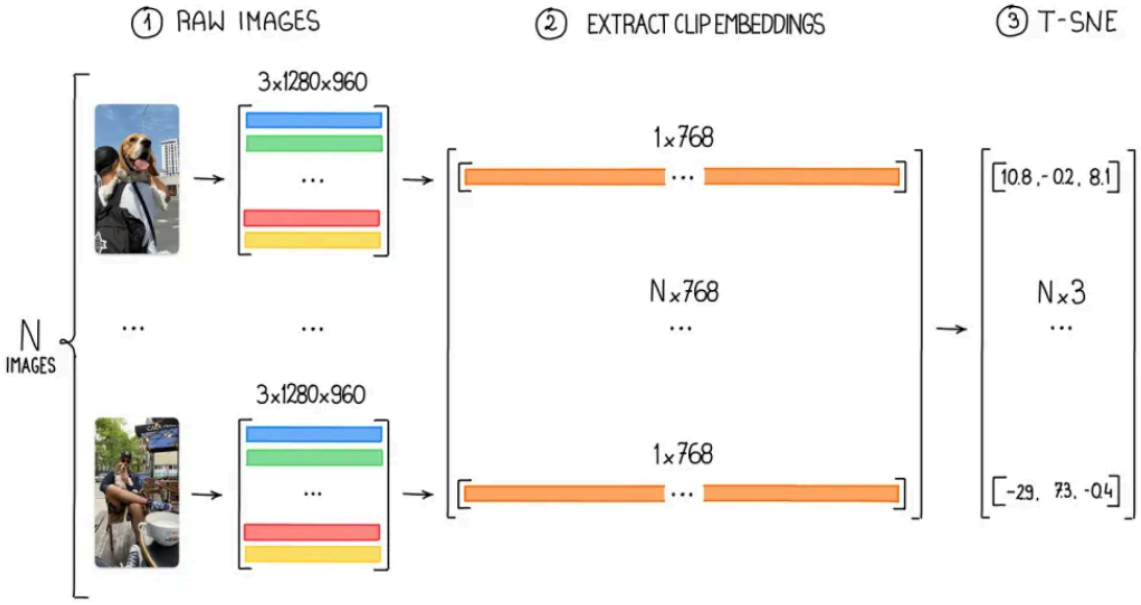


Diagram depicting the dimensions of the tensors at each stage of processing when using CLIP embeddings.

CLIP embeddings address this issue by providing a more abstract and compact representation of images, effectively encoding high-level visual and semantic information. By using CLIP embeddings, we can efficiently work with high-resolution images while preserving their essential visual and semantic characteristics for various computer vision tasks

An image embedding will encode this information. We could then compare the image embedding to a text embedding like "fruit" to see how similar the concept of "fruit" is to the contents of the image. We could take two prompts, such as "fruit" and "vegetable", and see how similar each one is. The most similar prompt is considered the most representative of the image.

You can save embeddings in a special database called a vector database (used to store embedding) to run efficient searches between image and text embeddings. This efficient searching enables you to build a "semantic" image search engine.

- **Dimensionality Reduction Techniques**
  https://programminghistorian.org/en/lessons/clustering-visualizing-word-embeddings - Read Dimension Reduction section only
  PCA: https://perma.cc/XSG8-NLU7 , https://www.datacamp.com/tutorial/pca-analysis-r
  T-SNE- https://www.datacamp.com/tutorial/introduction-t-sne
  UMap- https://pair-code.github.io/understanding-umap/

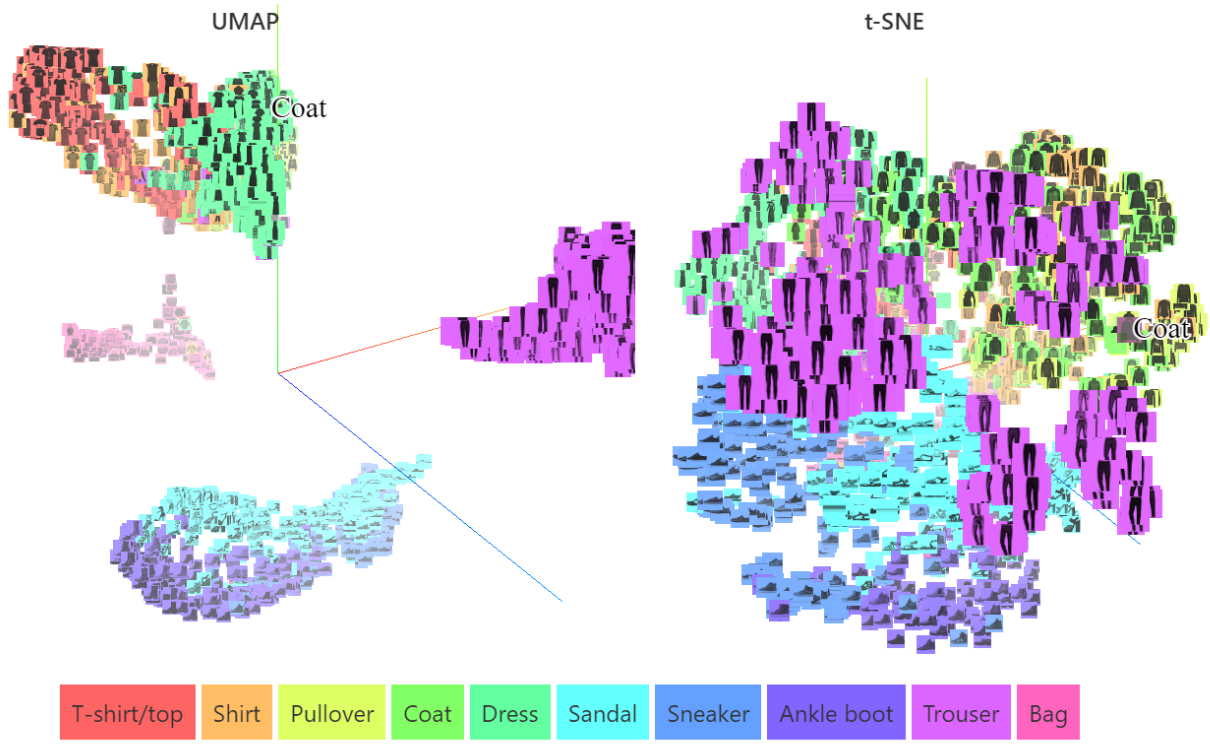


Figure 2: *Dimensionality reduction applied to the Fashion MNIST dataset. 28x28 images of clothing items in 10 categories are encoded as 784-dimensional vectors and then projected to 3 using UMAP and t-SNE.*

The Plan- Part 1

1- Rag Based Word/Image embeddings
2- Knowledge Graphs Based RAG

https://github.com/rahulnyk/knowledge_graph

Ollama-NetworkX

https://github.com/rahulnyk/knowledge_graph?tab=readme-ov-file

https://www.nlplanet.org/course-practical-nlp/02-practical-nlp-first-tasks/16-knowledge-graph-from-text

To build a knowledge graph from text, we typically need to perform two steps:

1. Extract entities, a.k.a. **Named Entity Recognition (NER)**, which are going to be the nodes of the knowledge graph.
2. Extract relations between the entities, a.k.a. **Relation Classification (RC)**, which are going to be the edges of the knowledge graph.

Relation extraction: https://paperswithcode.com/task/relation-extraction

https://nlpprogress.com/english/relationship_extraction.html

Model that we will be using for Relation extraction will be REBEL

Why will we be using rebel and a brief intro about Relation extraction:

REBEL (Relation Extraction By End-to-end Learning) is a transformer-based model designed for relation extraction. Relation extraction is a natural language processing (NLP) task that involves identifying relationships between entities within a text.

The "large" in REBEL-large indicates that this is a larger, more complex version of the model, which typically means it has more parameters and thus can capture more nuanced patterns in the data.

Why is REBEL Used?

REBEL is used primarily for the task of relation extraction due to several reasons:

1. **End-to-End Learning**: Unlike traditional methods that might require multiple stages of processing, REBEL is designed to handle relation extraction in a single end-to-end framework. This simplifies the pipeline and can improve efficiency and performance.
2. **Transformer Architecture**: Being based on the transformer architecture, REBEL can leverage the advantages of transformers such as parallel processing and attention mechanisms. This allows it to handle large amounts of text and capture complex relationships between entities.
3. **Large-Scale Pretraining**: Models like REBEL-large benefit from pretraining on vast amounts of data, which helps them understand language patterns and relationships more effectively. This pretraining can significantly boost performance on downstream tasks like relation extraction.

https://medium.com/@ido.benshaul/having-fun-with-clip-features-part-i-29dff92bbbcd

Getting started with RAG KGE for LLMs:

GraphRAG: LLM-Derived Knowledge Graphs for RAG

1- Choose sample corpus jaise any article