4 Days Workshop on

Artificial Intellegence

Text Embeddings

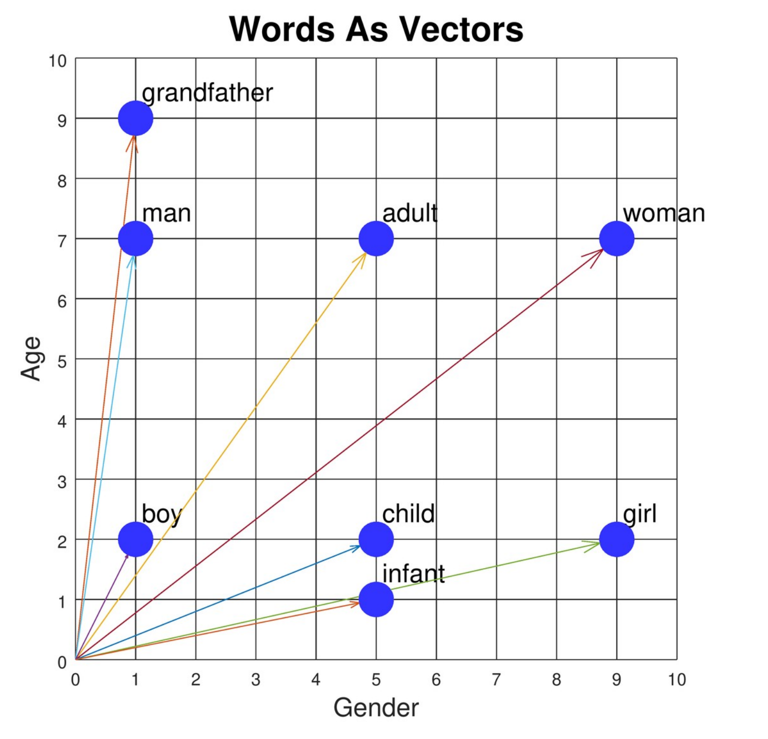
Representing text into numbers in better way.

What are embeddings ?

- Traditional vectors don’t capture meaning, context or word order.
- We want numbers that represent semantic meaning, not just position.
- Embeddings turn text into dense vectors while capturing meaning behind word.
- Word with similar meaning have similar embeddings.

Word2Vec and GloVe

- It is the first context based word embedding.
- Word2Vec learns word meaning from surrounding words.
- GloVe Combines word co-occurrence with neural embeddings and improves quality by looking at entire corpus statistics
- BUT 1 vector per word, Apple has same vector for both fruit and company.



ELMo – Embeddings Of Language Models

- First contexutal embedding.
- Uses deep bi directional LSTM (Improved version of RNN)
- Bank gets different embedding in “river bank” and “money bank”
- Slow and not great for sentence level task.

BERT – Deep Transformer Embeddings

- Uses Transformers to get deep, bidirectional context
- Outputs embeddings for each words.
- Pretrained on massive corpora (e.g., Wikipedia)
- Great for downstream tasks (search, Q&A, etc.)
- But it’s a bit heavy to run and isn’t designed to return a single vector for a sentence, which we often need.

Sentence Transformers

- Built on top of BERT.
- Returns a single vector for a full sentence
- Trained for similarity/search directly
- Perfect for semantic search, clustering, recommendations

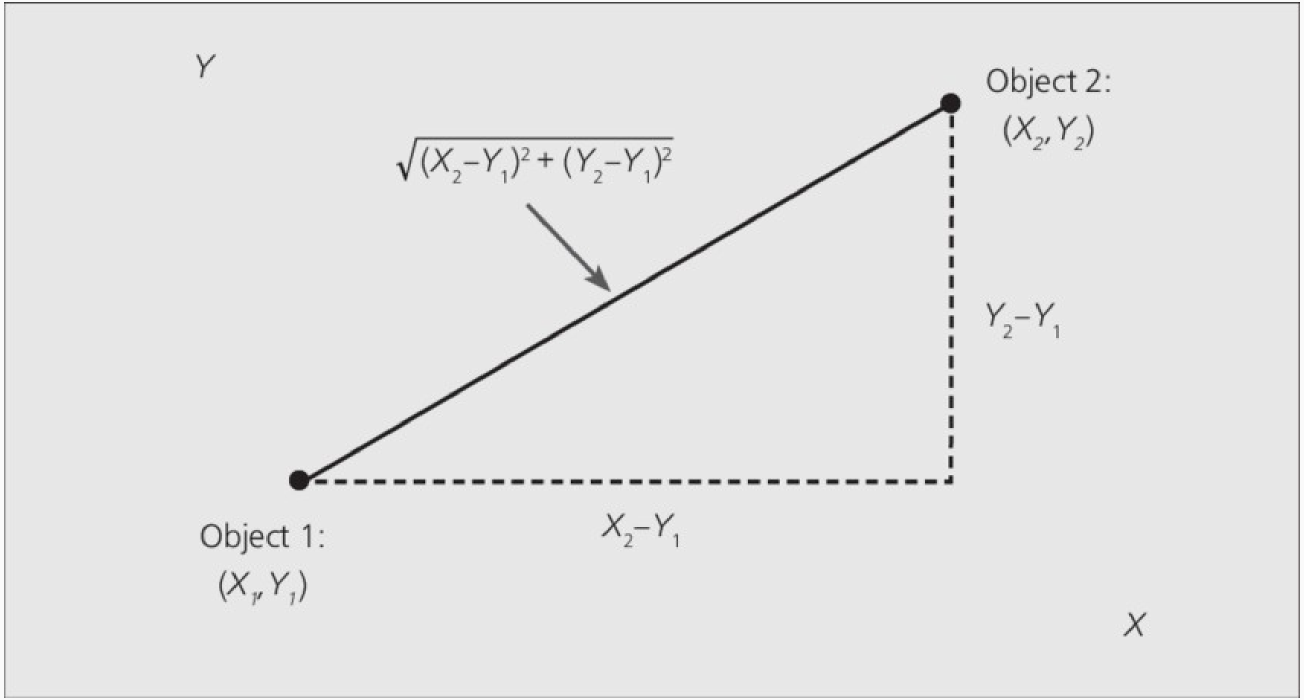
Morden Embeddings & Beyond

- Trained on huge datasets with powerful transformer models.
- Capture deep semantic relationships.
- Work across languages, topics, tasks.
- Used in ChatGPT, semantic search, LLM agents.
- Are extremely rich, general-purpose.
- BUT usually paid APIs, not open-source.
- E.g. OpenAI, Cohere, or Google Embeddings.

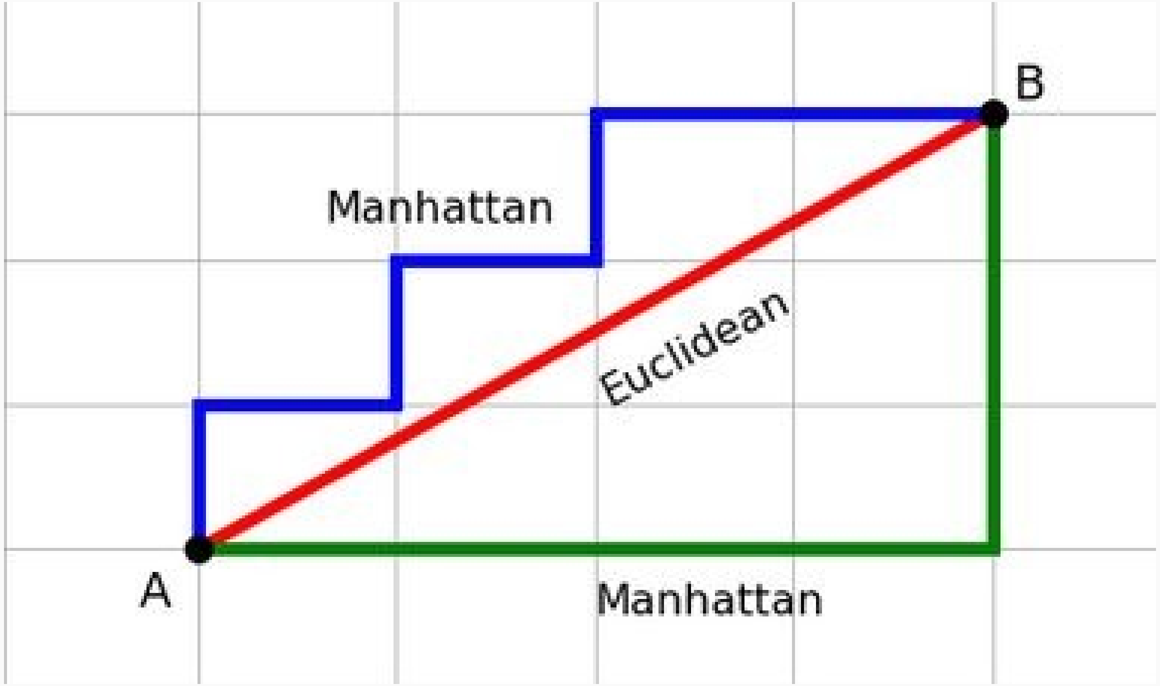
Distance Metrics

Different ways of measuring similarity in vectors

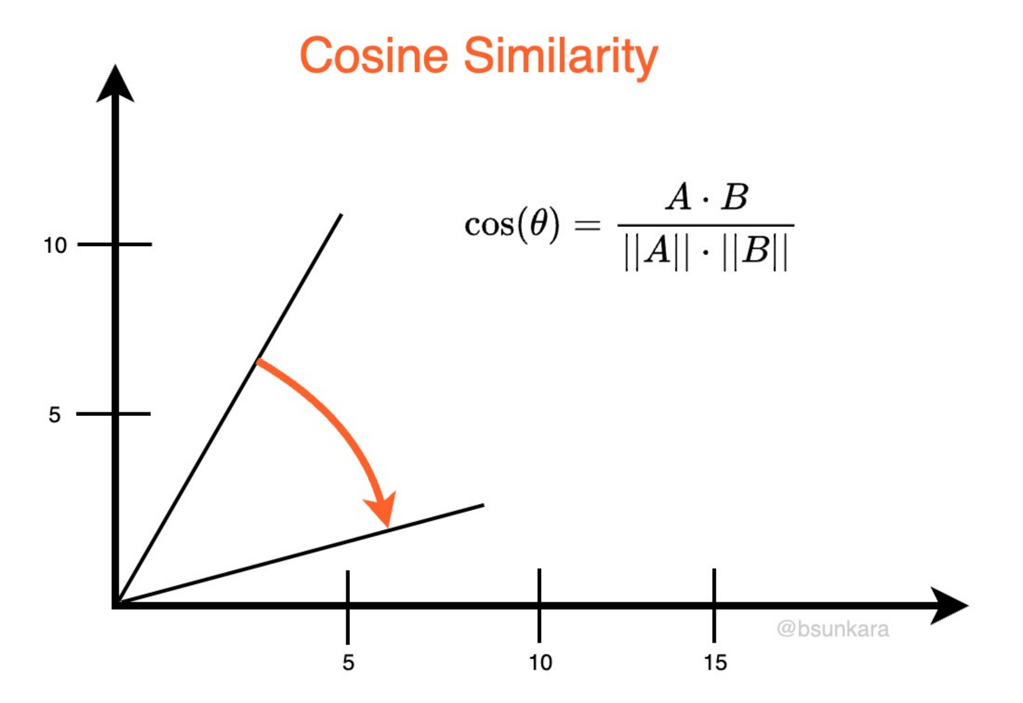
Euclidean Distance



Manhattan Distance



Cosine Similarity (Normalized Dot Product)



$$\cos(\theta) = \frac{A \cdot B}{||A|| \cdot ||B||}$$