cim lingen 2024

Sebastian Gingter

Wieso versteht der Computer mich auf einmal?

Wir lüften das Geheimnis von Embeddings

Sebastian Gingter

Developer Consultant @ Thinktecture AG

- Generative AI in business settings
- Flexible and scalable backends
- All things .NET

- Pragmatic end-to-end architectures
- Developer productivity
- Software quality

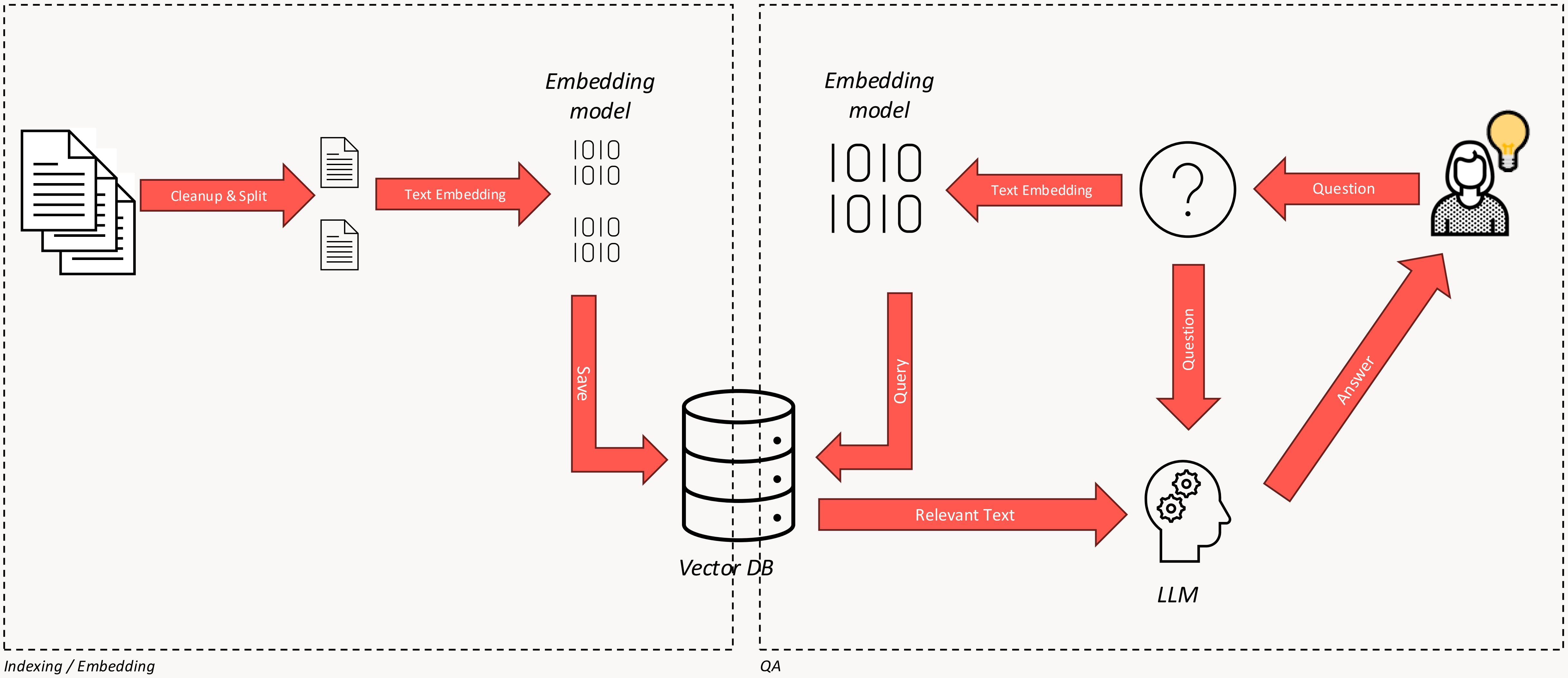
sebastian.gingter@thinktecture.com @phoenixhawk https://www.thinktecture.com

Agenda

- Introduction
- Embeddings, Embedding Models
  - and theory 😱
- Conclusion

Introduction

Use-cse Retrieval-Augmented Generation (RAG)
Indexing & (Semantic) search



Semantic Search

- Classic search: lexical
  - Compares words, parts of words and variants
  - Classic SQL: WHERE ‘content’ LIKE ‘%searchterm%’
  - We can search only for things where we know that its somewhere in the text

- New: Semantic search
  - Compares for the same contextual meaning
    - “Das Rudel rollt das runde Gerät auf dem Rasen herum”
    - “The pack enjoys rolling a round thing on the green grass”
    - “Die Hunde spielen auf der Wiese mit dem Ball”
    - “The dogs play with the ball on the meadow”

Semantic Search

- How to grasp “semantics”?
- Computers only calculate on numbers
  - Computing is “applied mathematics”
- AI also only calculates on numbers
- How to convert text to numbers?
  - ASCII 😅

Basics

First: Tokens

- “Chatbots are, if used correctly, a useful tool.”
- “Chatbots_are,_if_used_correctly,_a_useful_tool.”
- [“Chat”, “bots”, “_are”, “,”, “_if”, “_used”, “_correctly”, “,”, “_a”, “_useful”, “_tool”, “.”]
- [14065, 91601, 553, 11, 538, 2061, 20323, 11, 261, 8316, 4584, 13]

https://platform.openai.com/tokenizer

Semantic Search

- Tokens are a numeric representation of text
- But we need a numeric representation of **meaning**

  ➔ “Embeddings”

Embeddings

Embedding (math.)

- Topologic: Value of a high dimensional space is “embedded” into a lower dimensional space
- Natural / human language is very complex (high dimensional)
    - Task: Map high complexity to lower complexity / dimensions
- Injective function
- Similar to hash, or a lossy compression

Embeddings

- Embedding model (specialized ML model) converting text into a numeric representation of its meaning

- Representation is a vector in an n-dimensional space
  - n floating point values
  - OpenAI
    - “text-embedding-ada-002” uses 1536 dimensions
    - “text-embedding-3-small” 512 and 1536
    - “text-embedding-3-large” 256, 1024 and 3072
  - Huggingface 🤗 models have a very wide range of dimensions

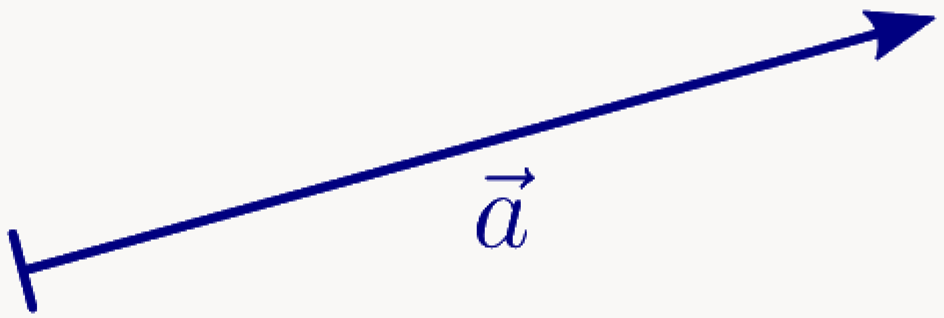
https://huggingface.co/spaces/mteb/leaderboard & https://openai.com/blog/new-embedding-models-and-api-updates

Embeddings

- Embedding models are unique
- Each dimension has a different meaning, individual to the model
- Vectors from different models are incompatible with each other
- Some embedding models are multi-language, but not all
- In an LLM, also the first step after tokenizing is to embed the input into a lower dimensional space

What is a vector?

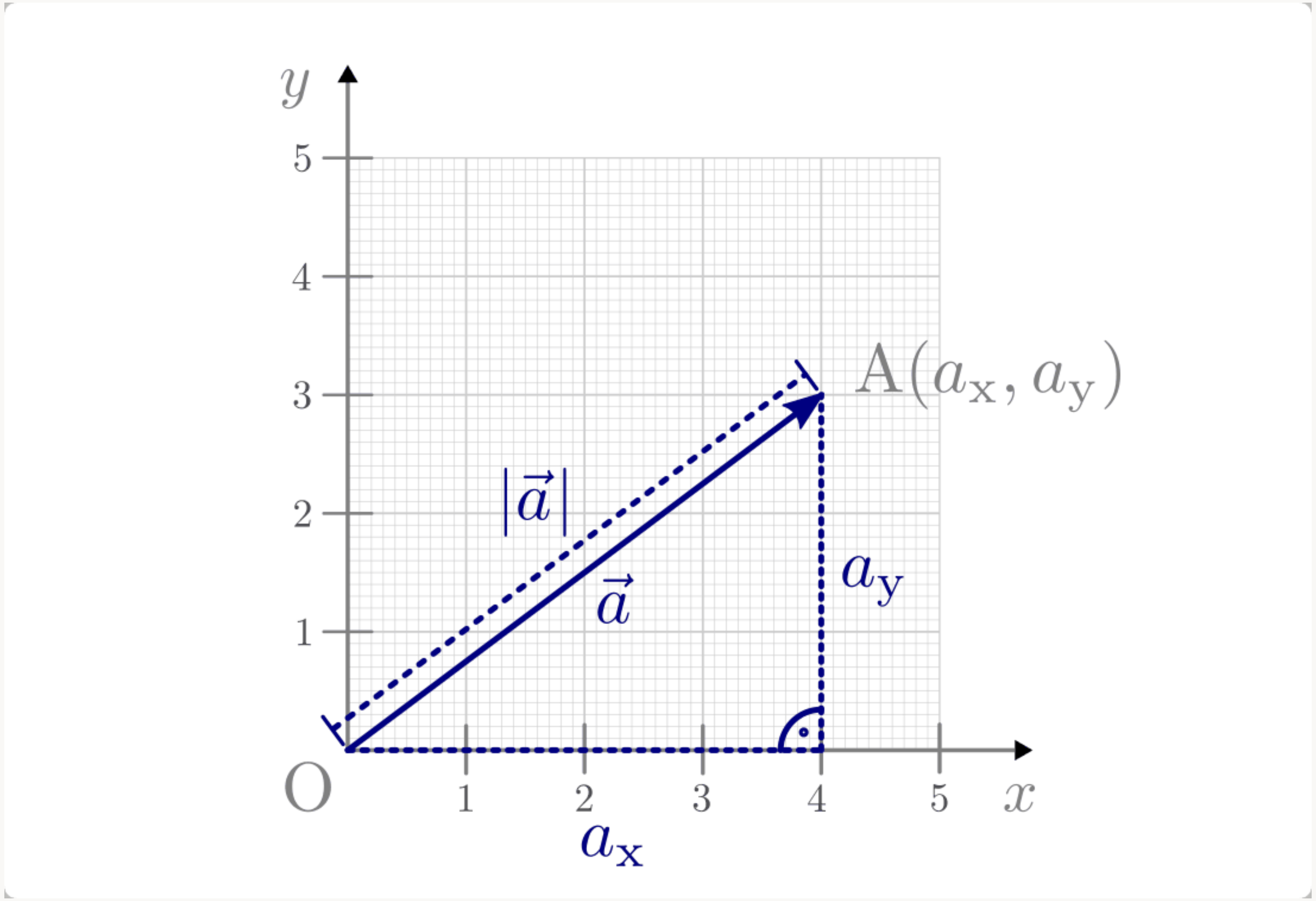
- Mathematical quantity with a direction and length
- $\vec{a} = \binom{a_x}{a_y}$



https://mathinsight.org/vector_introduction

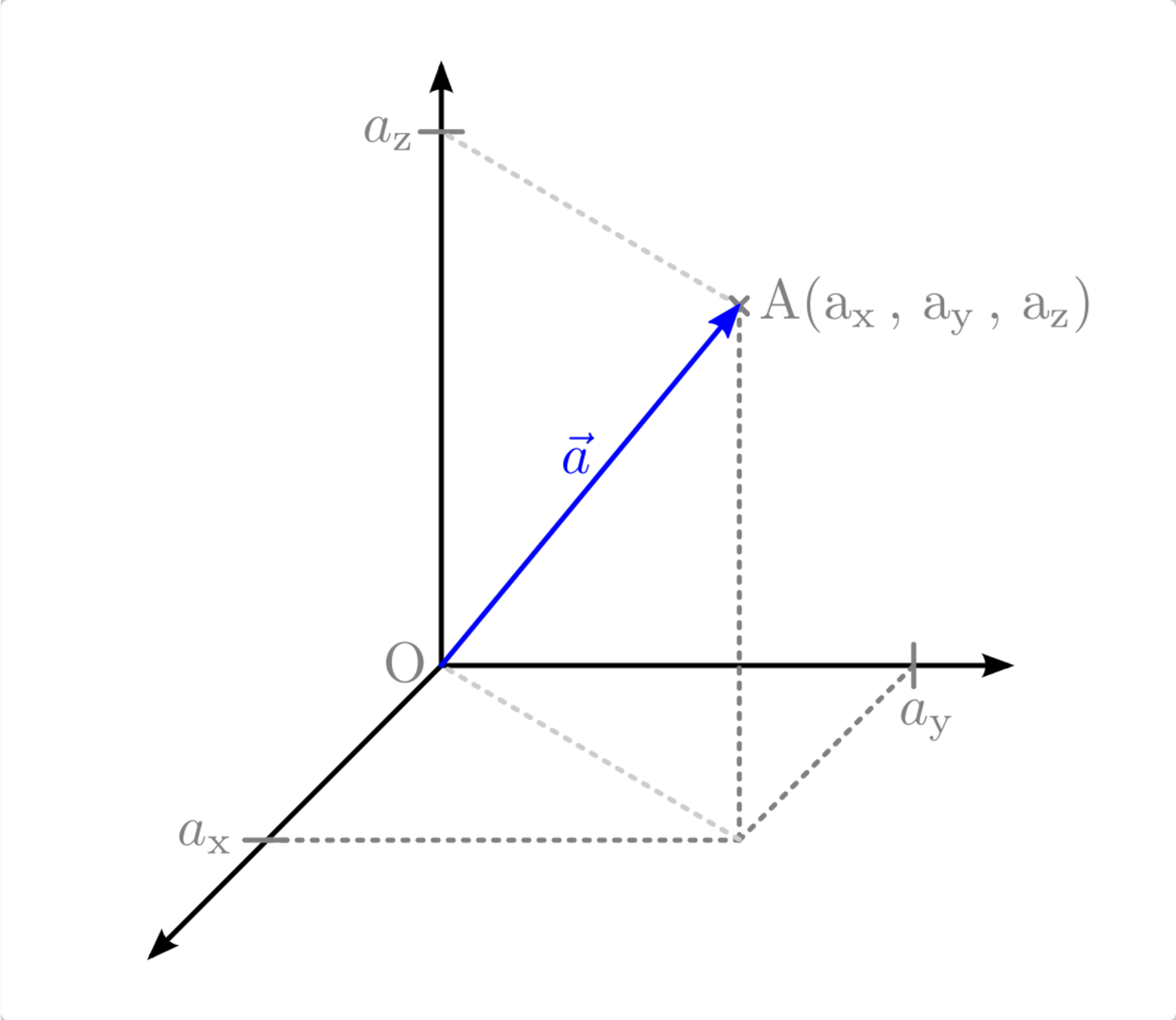
Vectors in 2D

$$\vec{a} = \begin{pmatrix} a_x \\ a_y \end{pmatrix}$$



Vectors in 3D

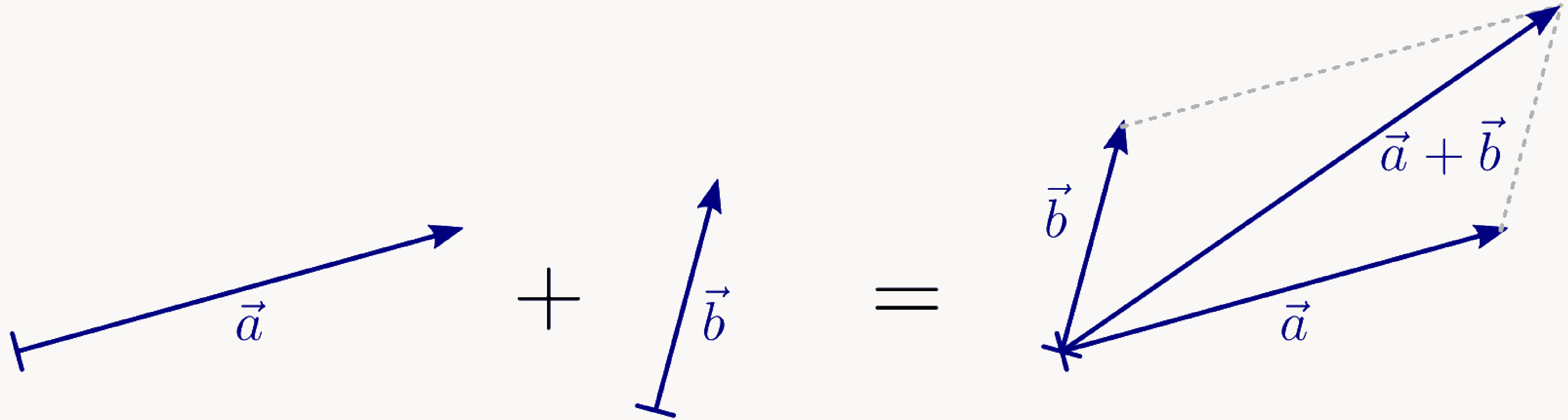
$$\vec{a} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}$$



Vectors in multidimensional space

$$\vec{a} = \begin{pmatrix} a_u \\ a_v \\ a_w \\ a_x \\ a_y \\ a_z \end{pmatrix}$$

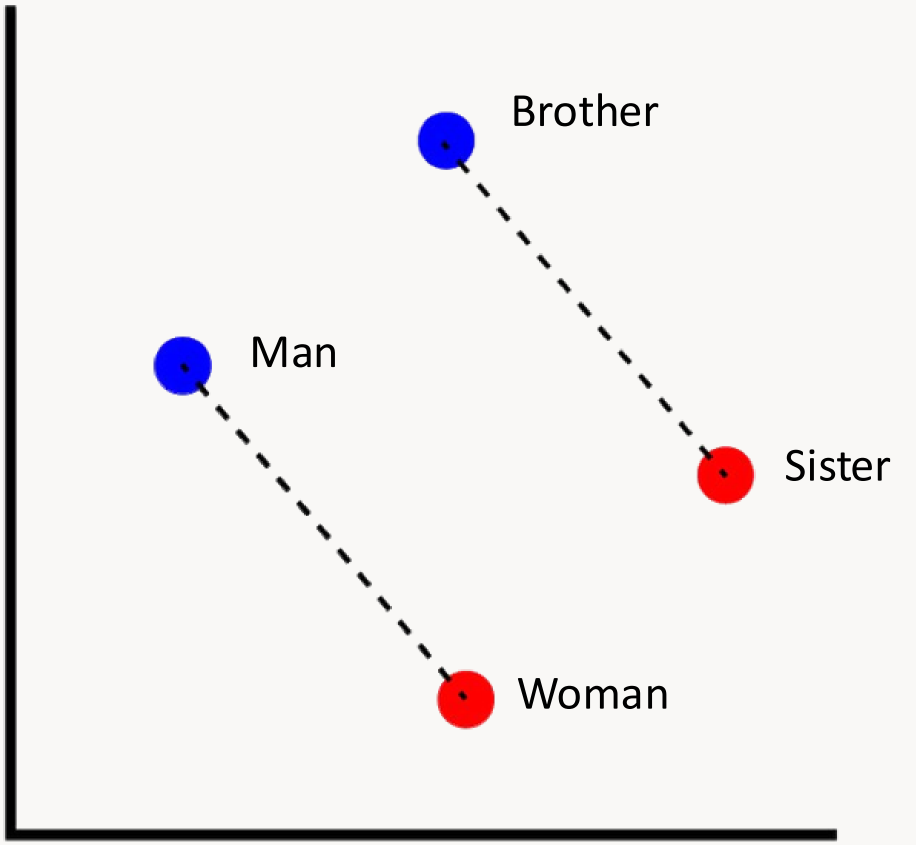
Calculation with vectors



Word2Vec

Mikolov et al., Google, 2013

$$Brother - Man + Woman \approx Sister$$

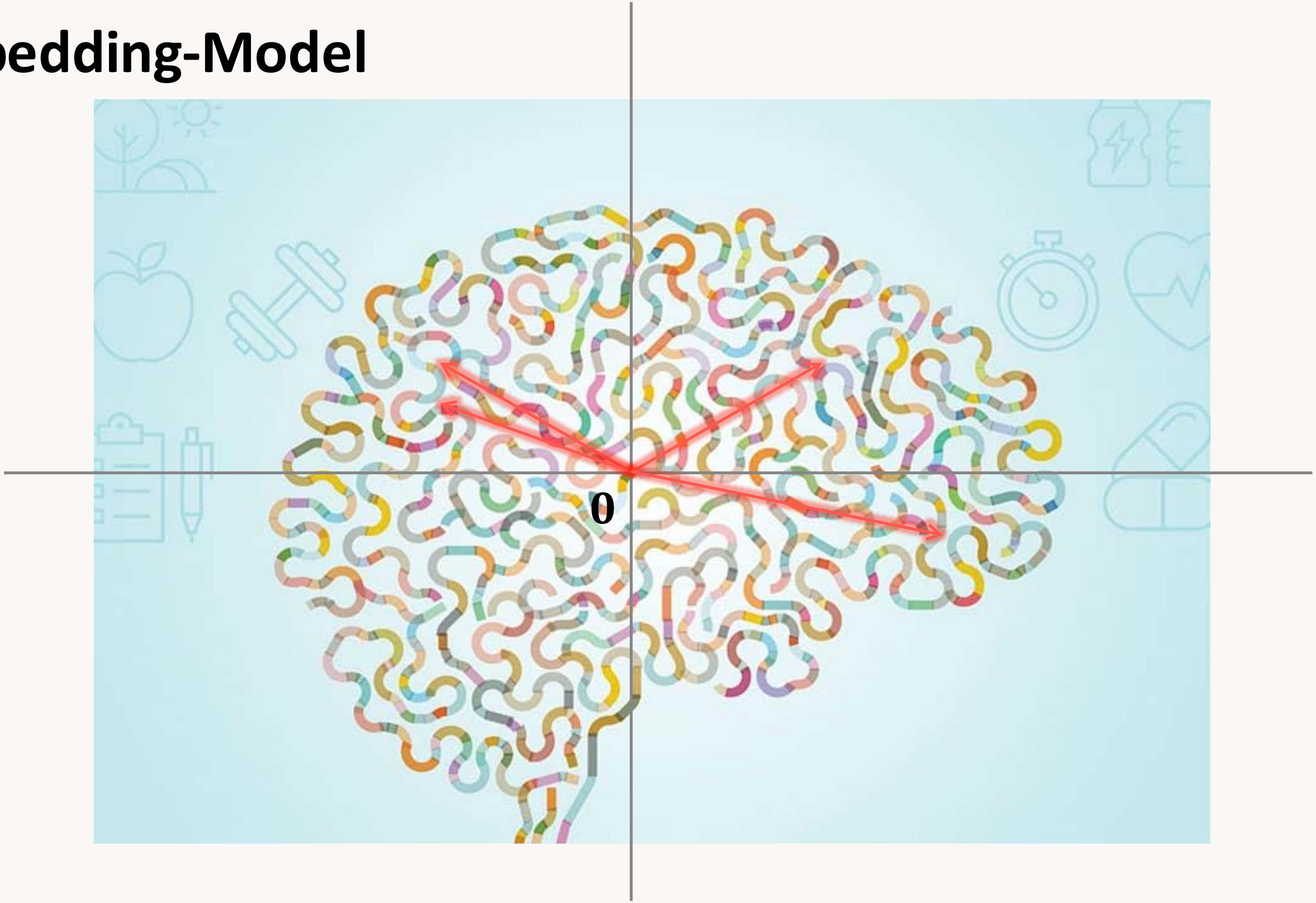


https://arxiv.org/abs/1301.3781

Embedding-Model

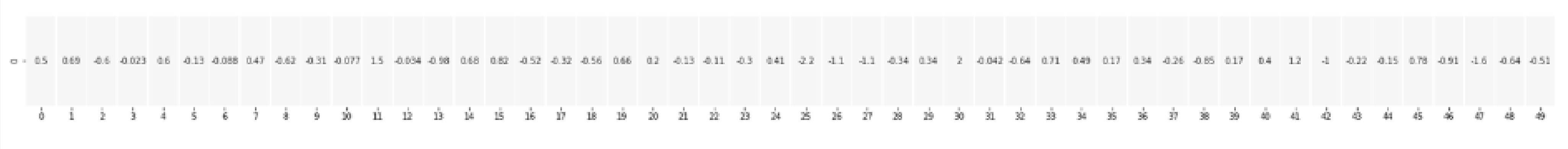
- Task: Create a vector from an input
    - Extract meaning / semantics
- Embedding models usually are very shallow & fast
  Word2Vec is only two layers
- Similar to the first step of an LLM (after tokenization)
    - Convert tokens to semantic values for input layer
- This comparison is very simplified, but one could say:
    - The embedding model ‘maps’ the meaning into the model’s ‘brain’

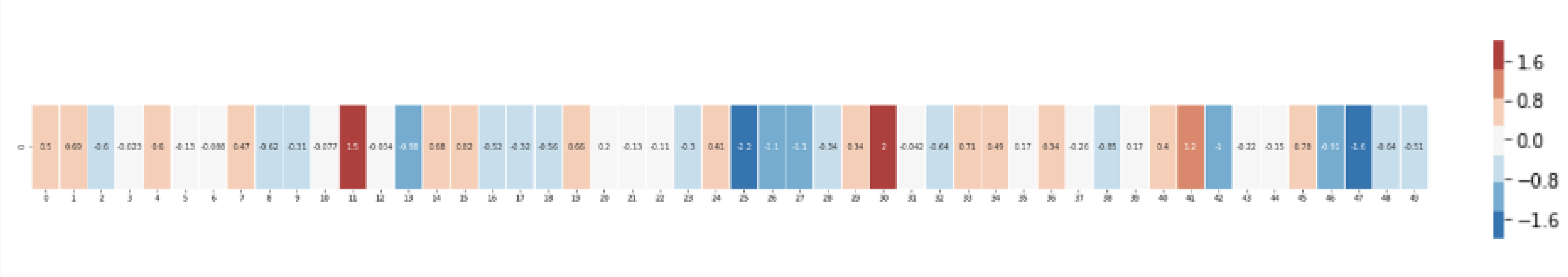
Embedding-Model



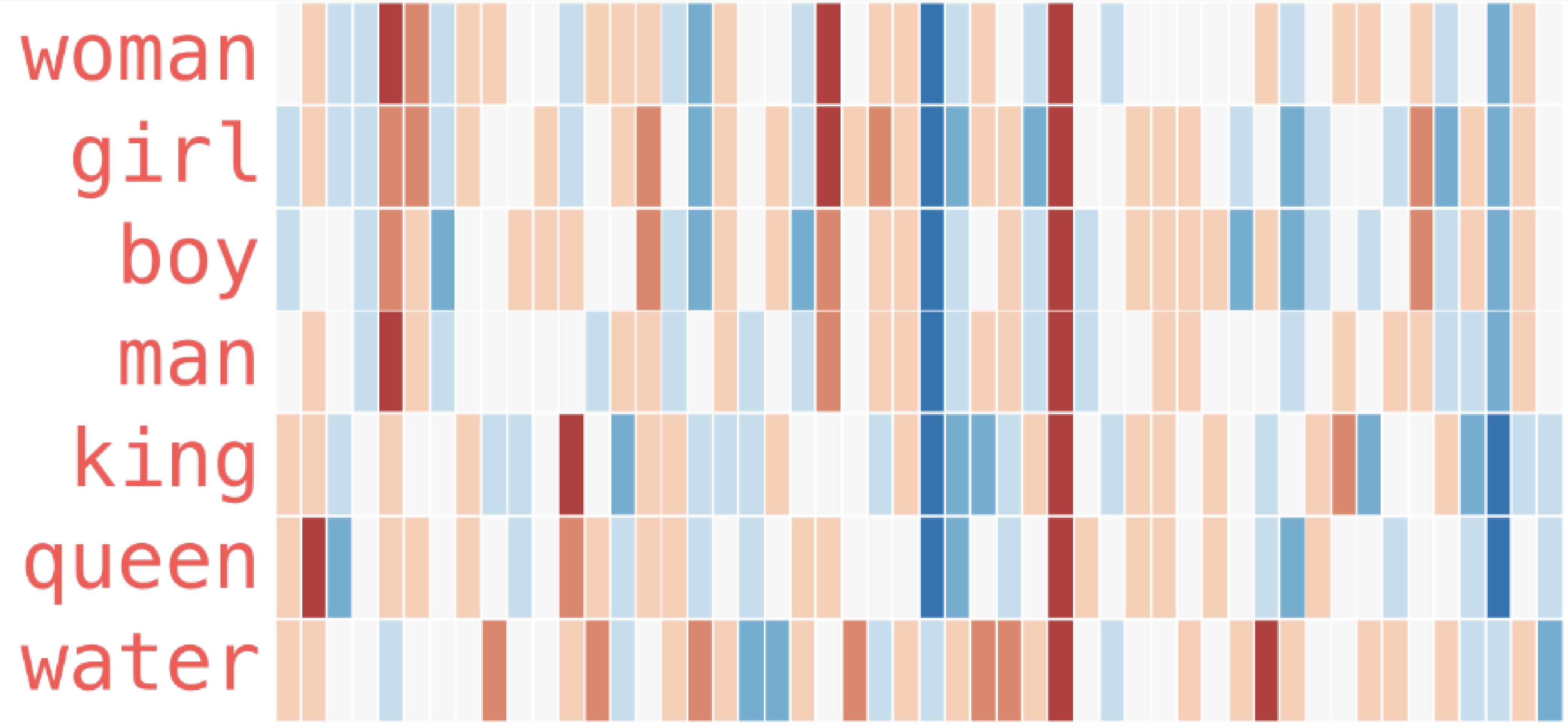
Embedding-Model

[0.50451 , 0.68607 , -0.59517 , -0.022801, 0.60046 , -0.13498 , -0.08813 , 0.47377 , -0.61798 , -0.31012 , -0.076666, 1.493 , -0.034189, -0.98173 , 0.68229 , 0.81722 , -0.51874 , -0.31503 , -0.55809 , 0.66421 , 0.1961 , -0.13495 , -0.11476 , -0.30344 , 0.41177 , -2.223 , -1.0756 , -1.0783 , -0.34354 , 0.33505 , 1.9927 , -0.04234 , -0.64319 , 0.71125 , 0.49159 , 0.16754 , 0.34344 , -0.25663 , -0.8523 , 0.1661 , 0.40102 , 1.1685 , -1.0137 , -0.21585 , -0.15155 , 0.78321 , -0.91241 , -1.6106 , -0.64426 , -0.51042]





Embedding-Model



http://jalammar.github.io/illustrated-word2vec/

Demo

Embeddings

Sentence Transformers, local embedding model

Recap Embeddings

- Embedding model: “Analog to digital converter for text”

- Embeds the high-dimensional natural language meaning into a lower dimensional-space (the model’s ‘brain’)

- No magic, just applied mathematics
- Math. representation: Vector of n dimensions
- Technical representation: array of floating point numbers

Important

- Select your Embedding Model carefully for your use case

- e.g.
  - intfloat/multilingual-e5-large-instruct ~ 50%
  - T-Systems-onsite/german-roberta-sentence-transformer-v2 < 70 %
  - danielheinz/e5-base-sts-en-de > 80% hit rate

- Maybe fine-tuning of the embedding model might be an option

- As of now: Treat embedding models as exchangeable commodities!

Also important

- Embedding models are “small” in comparison to LLMs
  - but still large
  - danielheinz/e5-base-sts-en-de is about 2 GB
- The inference engine for embeddings is large
  - Sentence transformers has dep. on NVIDIA Cuda, 2.4 GB+
- Docker container with pre-loaded model is 5+ GB

Conclusion

Conclusion

- Embedding models vary strongly in quality
    - Have a plan to change the model / recalculate embeddings
- Normalize your vectors to make search fast(er)
- Cluster your collections when search becomes slow or needs too much memory

Conclusion

- Use cases
  - semantic search
  - semantic routing
  - tool selection
  - prompt hacking detection
  - …

Ich danke Euch
und wir danken unseren cim Sponsoren!

Slides: