Embeddings as data structures 2.0?

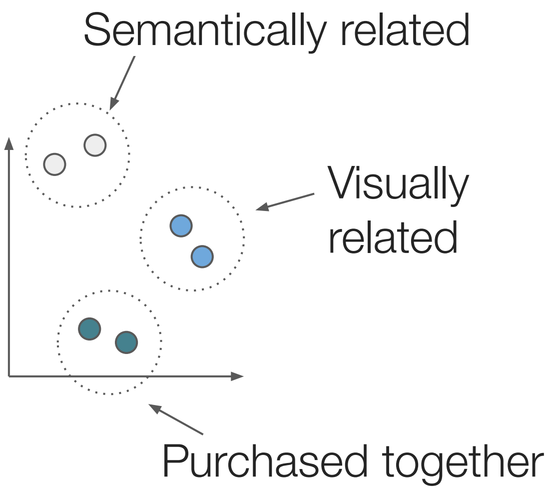
*"**Data structures 2.0** is written in human unfriendly language, such as the floating point values of an embedding. No human is involved in writing this code ... and coding directly in the floating point values is kind of tedious but possible (I tried)." [2]*

$$\underset{N_\theta,\theta}{\operatorname{argmin}}\, L(\mathbf{X}, \mathbf{Y}; N_\theta)$$

Network architecture

Parameters

Data describing the similarity to encode



marqo

Building Software 2.0 with embeddings

Jesse Clark
Marqo

marqo

About

@Marqo - The Embeddings Cloud

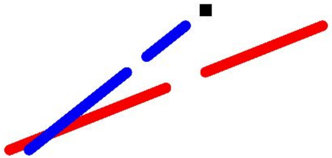
Co-founder/CTO

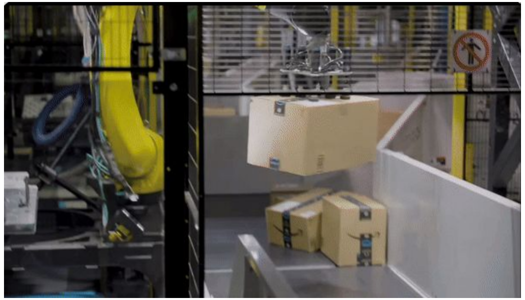
@jn2clark

Physics @ UCL & Stanford

ML and OR @ StitchFix

Deep learning and AI @ Amazon





Building Software 2.0 with embeddings

Outline

- What is Software 2.0?
    - Examples
- What are embeddings?
    - Producing embeddings
    - Training models for embeddings
- Building with embeddings
    - Classification, search and RAG

What is Software 2.0?

*"**Software 2.0** is written in human unfriendly language, such as the weights of a neural network. No human is involved in writing this code ... and coding directly in weights is kind of hard (I tried)." [1]*

[1] https://karpathy.medium.com/software-2-0-a64152b37c35

marqo

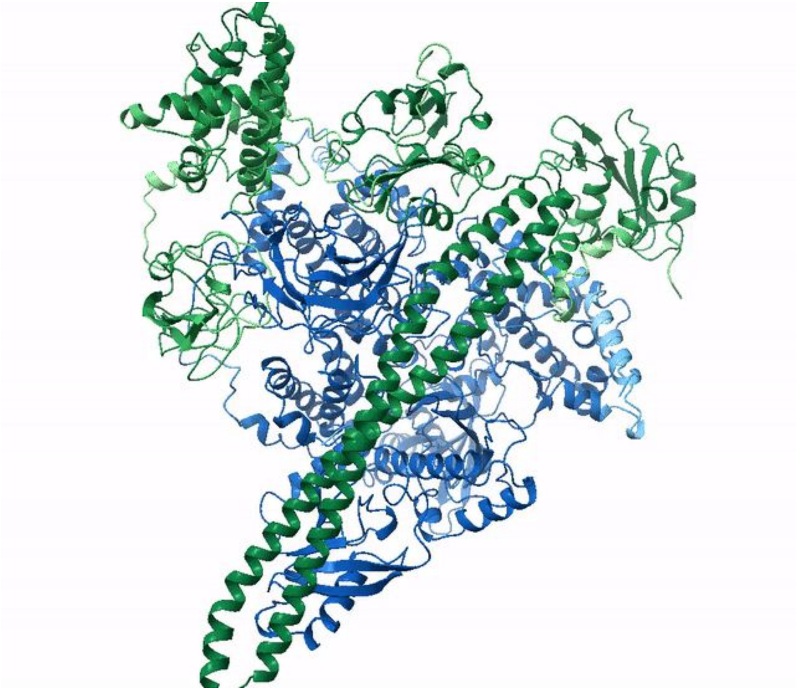
What is Software 2.0? - Examples

Visual Recognition

marqo

What is Software 2.0? - Examples

Structure prediction



What is Software 2.0? - Examples

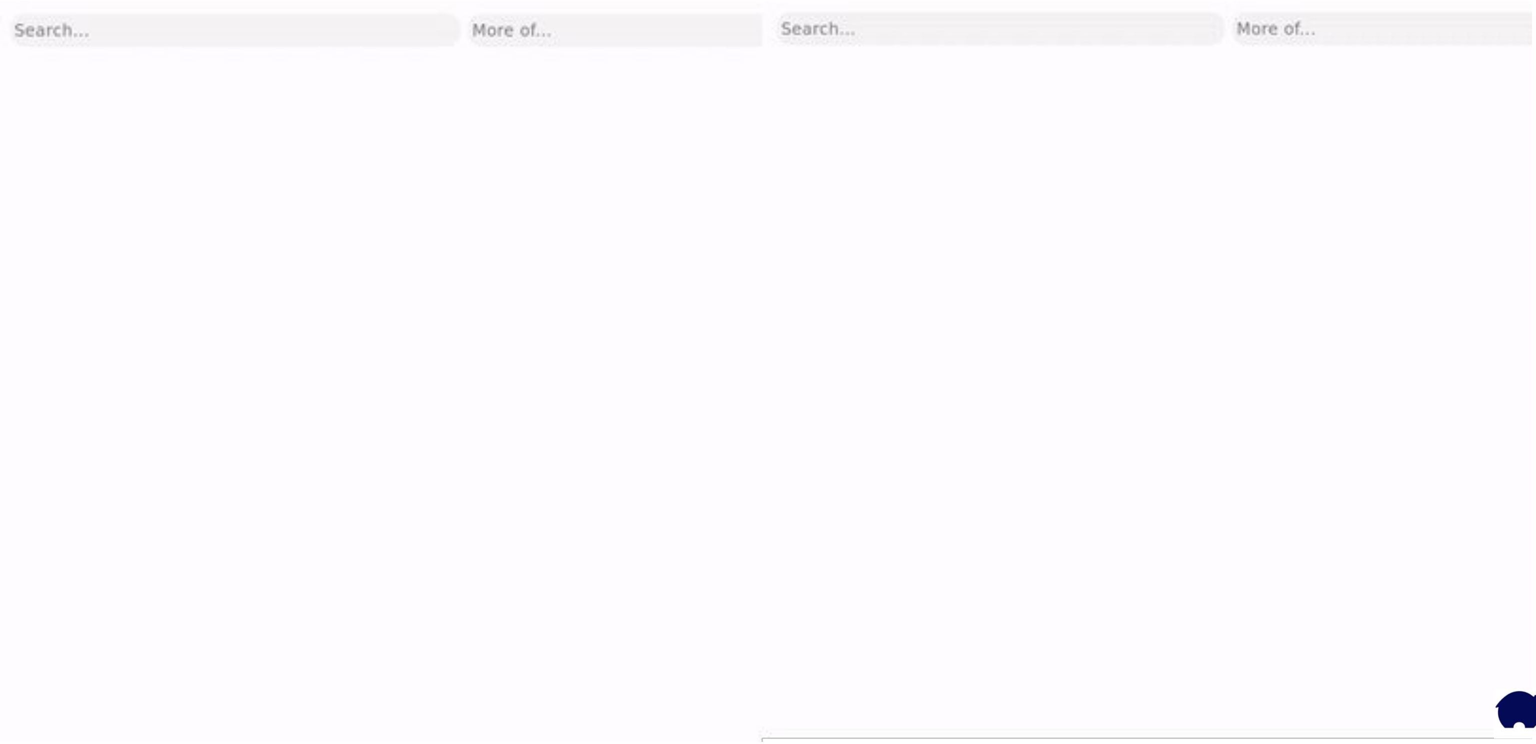
Text summarisation

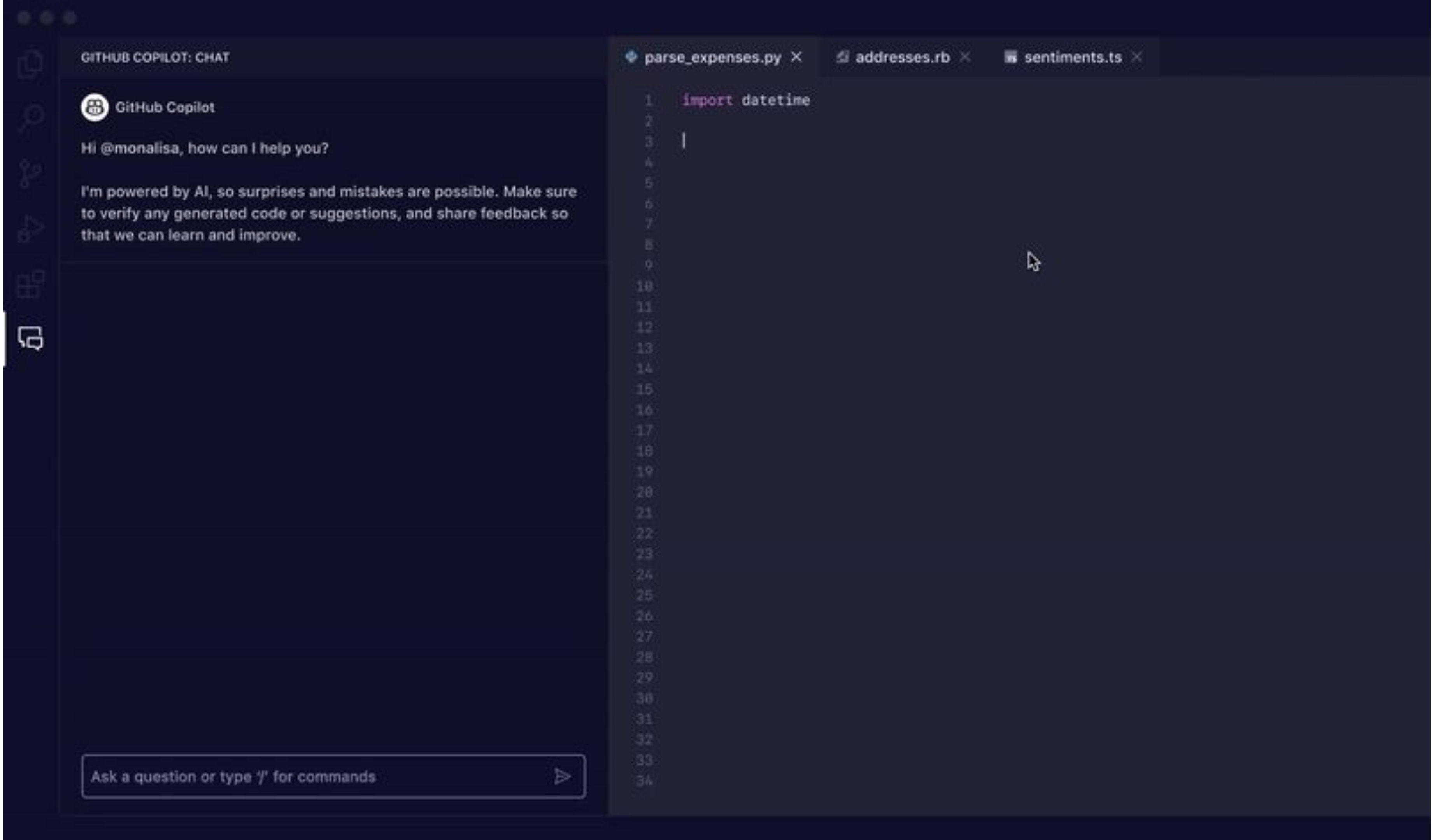
The bottleneck is no longer access to information; now it's our ability to keep up.
AI can be trained on a variety of different types of texts and summary lengths.
A model that can generate long, coherent, and meaningful summaries remains an open research problem.

The last few decades have witnessed a fundamental change in the challenge of taking in new information. The bottleneck is no longer access to information; now it's our ability to keep up. We all have to read more and more to keep up-to-date with our jobs, the news, and social media. We've looked at how AI can improve people's work by helping with this information deluge and one potential answer is to have algorithms automatically summarize longer texts. Training a model that can generate long, coherent, and meaningful summaries remains an open research problem. In fact, generating any kind of longer text is hard for even the most advanced deep learning algorithms. In order to make summarization successful, we introduce two separate improvements: a more contextual word generation model and a new way of training summarization models via reinforcement learning (RL). The combination of the two training methods enables the system to create relevant and highly readable multi-sentence summaries of long text, such as news articles, significantly improving on previous results. Our algorithm can be trained on a variety of different types of texts and summary lengths. In this blog post, we present the main contributions of our model and an overview of the natural language challenges specific to text summarization.

What is Software 2.0? - Examples

Search and recommendations




GITHUB COPILOT: CHAT
GitHub Copilot
Hi @monalisa, how can I help you?
I'm powered by AI, so surprises and mistakes are possible. Make sure to verify any generated code or suggestions, and share feedback so that we can learn and improve.
Ask a question or type '/' for commands
parse_expenses.py
addresses.rb
sentiments.ts
import datetime

What is Software 2.0?

*"**Software 2.0** is written in human unfriendly language, such as the weights of a neural network. No human is involved in writing this code ... and coding directly in weights is kind of hard (I tried)." [1]*

Software 1.0

Rules + data → Answers

```python
def binary_search(arr, target):
    low = 0
    high = len(arr) - 1

    while low <= high:
        mid = (low + high) // 2

        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            low = mid + 1
        else:
            high = mid - 1

    return -1
```

Software 2.0

Answers + data → Rules

$$\underset{N_\theta,\theta}{\operatorname{argmin}}\, L(\mathbf{X}, \mathbf{Y}; N_\theta)$$

Network architecture

Parameters

"Rules"

Data

Desired output ("Answers")

What is Software 2.0?

*"**Software 2.0** is written in human unfriendly language, such as the weights of a neural network. No human is involved in writing this code ... and coding directly in weights is kind of hard (I tried)." [1]*

Software 2.0

Answers + data → Rules

"Find a function that produces the outputs we want from the data we have"

$$\underset{N_\theta,\theta}{\operatorname{argmin}}\, L(\mathbf{X}, \mathbf{Y}; N_\theta)$$

Network architecture

Parameters

Data

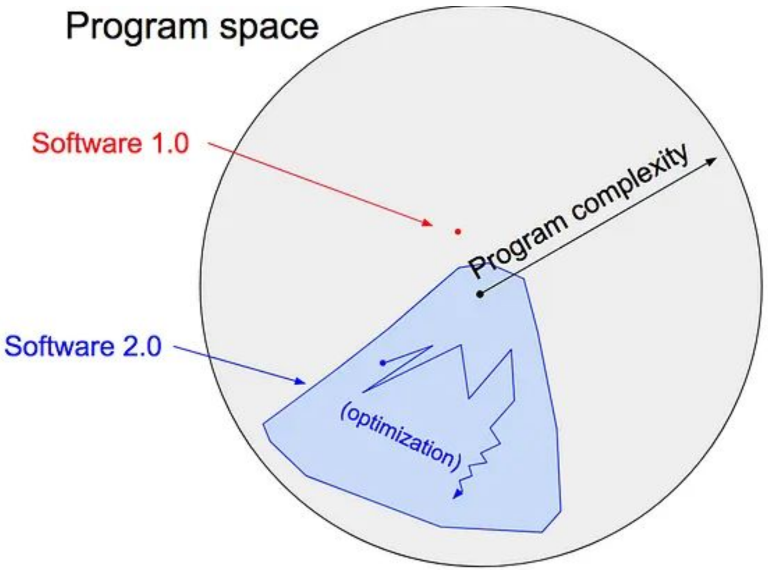
Desired output ("Answers")

"Rules"

marqo

What is Software 2.0?

*"**Software 2.0** is written in human unfriendly language, such as the weights of a neural network. No human is involved in writing this code ... and coding directly in weights is kind of hard (I tried)." [1]*



Software 2.0

Answers + data → Rules

$$\underset{N_\theta,\theta}{\operatorname{argmin}}\, L(\mathbf{X}, \mathbf{Y}; N_\theta)$$

marqo

Why Software 2.0? - Benefits and considerations

Benefits

- Computationally homogeneous
- Constant running time and memory
- Simple to bake into silicon
- It is highly portable
- Direct optimization
- It is much better

Drawbacks

- Harder to understand
- Soft and unexpected failure modes
- Can be difficult to test
- Requires strict testing and monitoring of data
    - "unit tests for data"

What is Software 2.0? - Summary

Software 1.0:

- Human-engineered source code (e.g., .cpp files)
- Compiled into a binary that does useful work

```python
def binary_search(arr, target):
    low = 0
    high = len(arr) - 1

    while low <= high:
        mid = (low + high) // 2

        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            low = mid + 1
        else:
            high = mid - 1

    return -1
```

Software 2.0:

- Source code comprises:
    - Dataset defining behavior
    - Neural net architecture providing code skeleton
- Many details (weights) to be filled in
- Training process compiles dataset into final neural network binary

$$\underset{N_\theta,\theta}{\operatorname{argmin}} L(\mathbf{X}, \mathbf{Y}; N_\theta)$$

Building Software 2.0 with embeddings

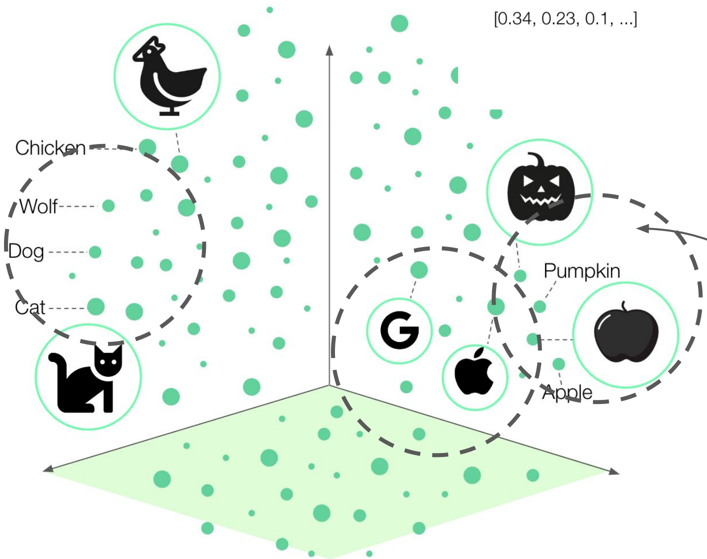
Outline

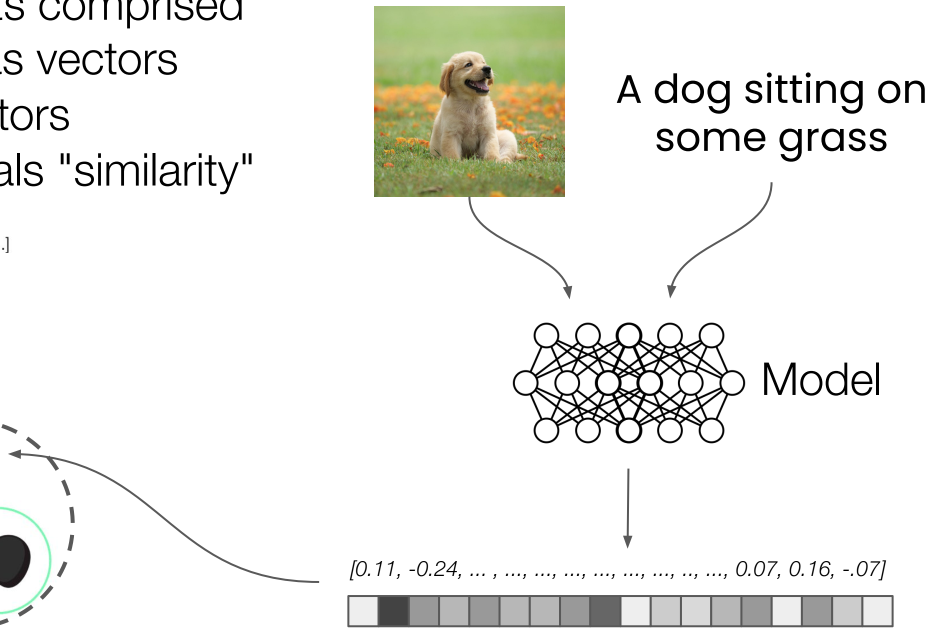
- What is Software 2.0?
  - Examples
- What are embeddings?
  - Producing embeddings
  - Training models for embeddings
- Building with embeddings
  - Classification, search and RAG

marqo

What are embeddings?

Vectors (Embeddings)
- Encode the meaning of objects comprised of text, images, audio, video as vectors
- Embeddings are (learned) vectors
- Proximity in vector space equals "similarity"



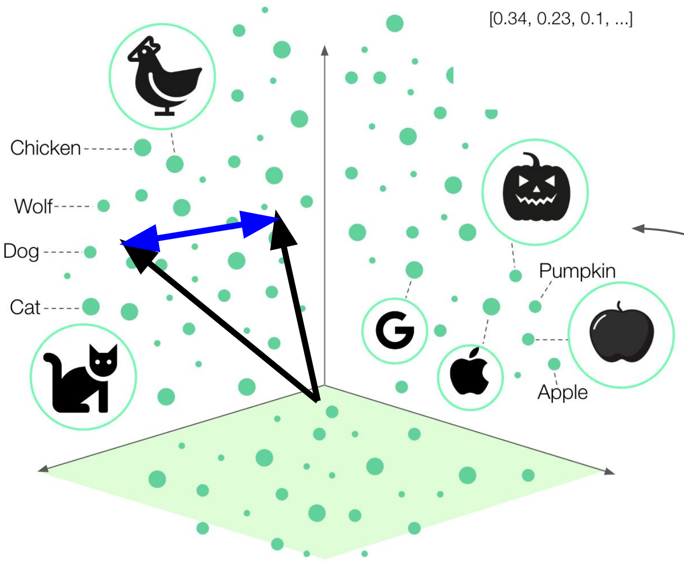


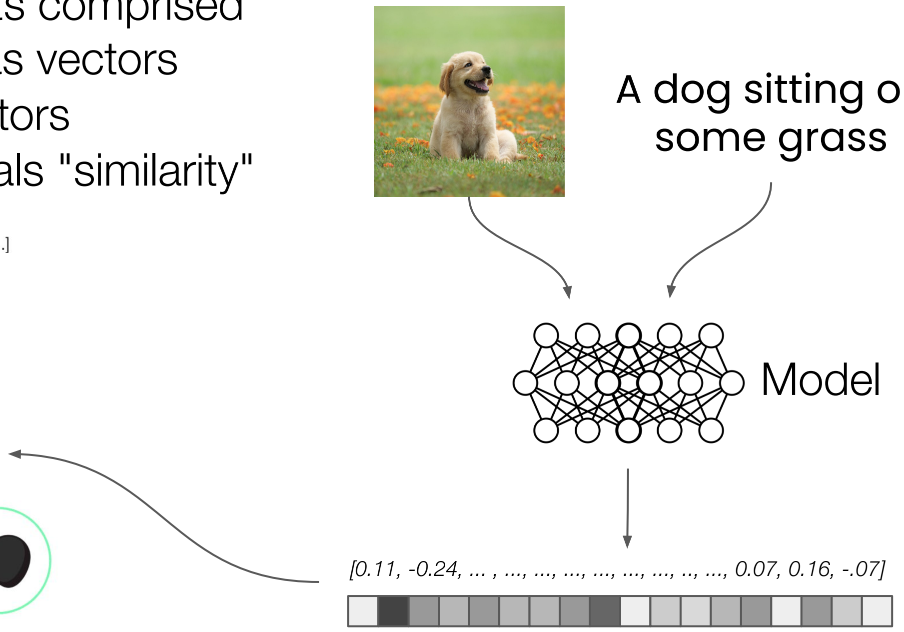
marqo

What are embeddings?

Vectors (Embeddings)

- Encode the meaning of objects comprised of text, images, audio, video as vectors
- Embeddings are (learned) vectors
- Proximity in vector space equals "similarity"

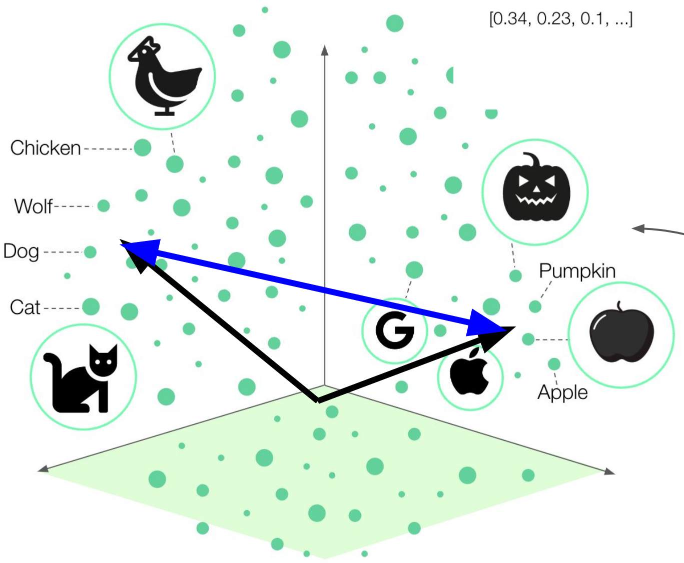


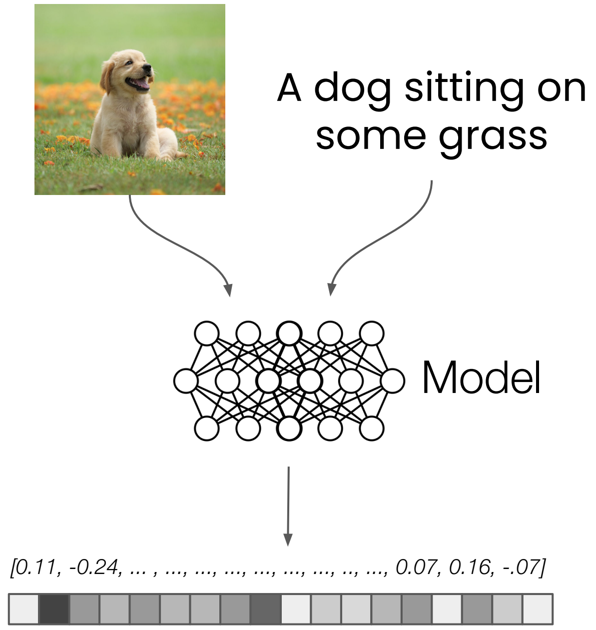


What are embeddings?

Vectors (Embeddings)

- Encode the meaning of objects comprised of text, images, audio, video as vectors
- Embeddings are (learned) vectors
- Proximity in vector space equals "similarity"

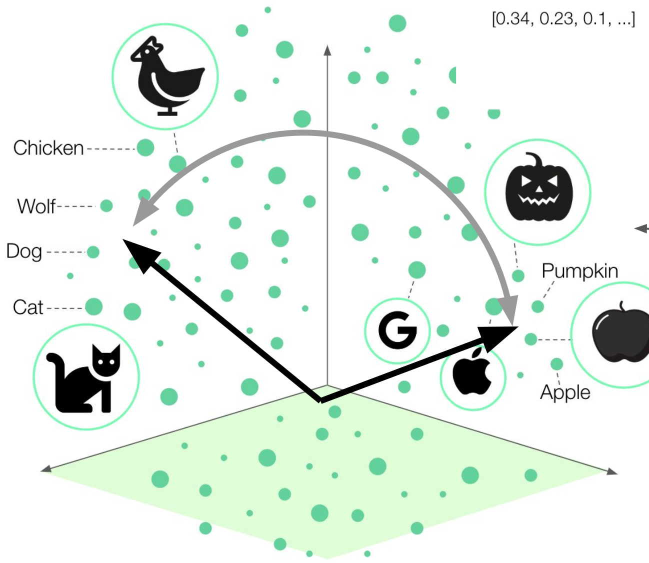


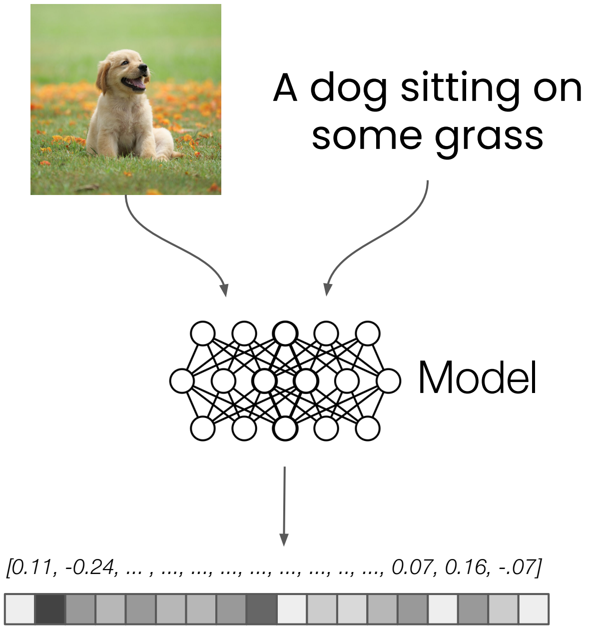


What are embeddings?

Vectors (Embeddings)

- Encode the meaning of objects comprised of text, images, audio, video as vectors
- Embeddings are (learned) vectors
- Proximity in vector space equals "similarity"



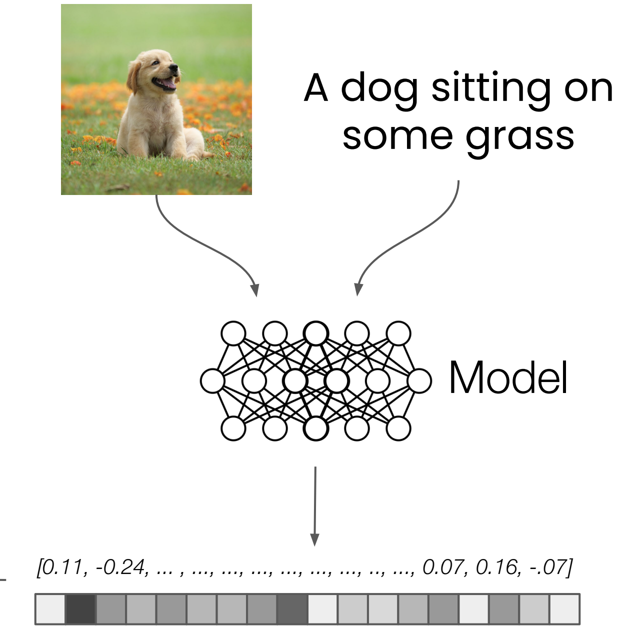


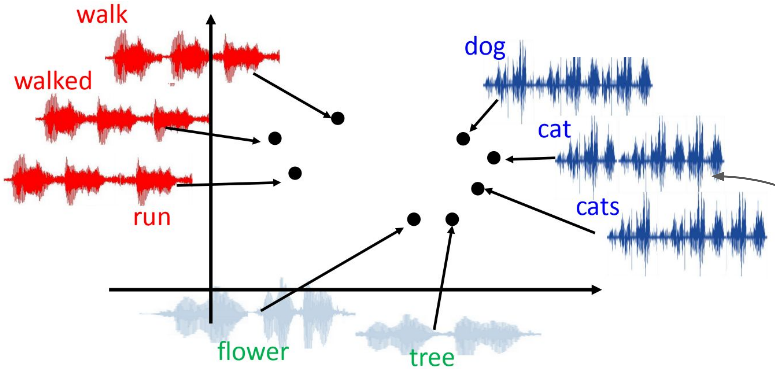
marqo

What are embeddings?

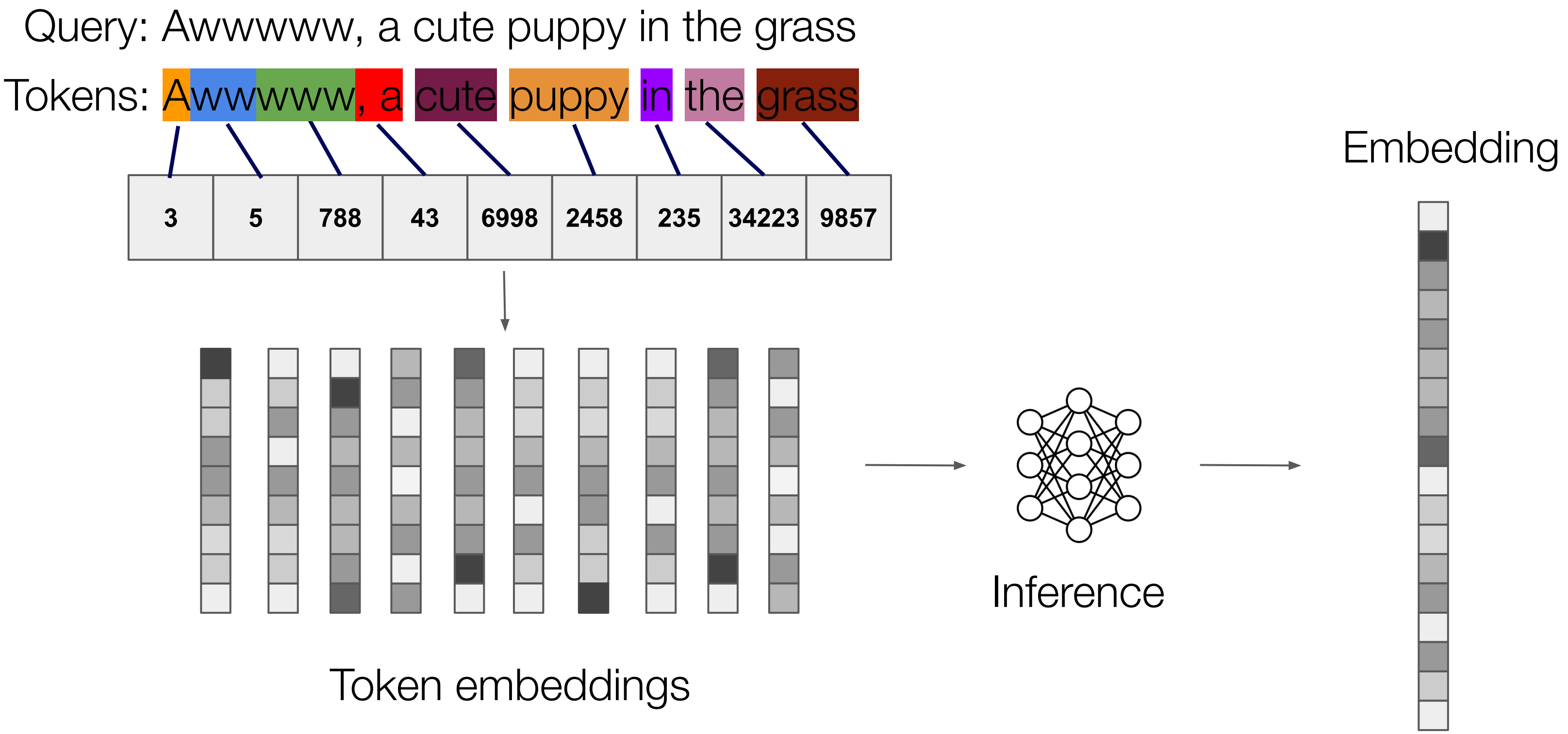
Vectors (Embeddings)

- Encode the meaning of objects comprised of text, images, audio, video as vectors
- Embeddings are (learned) vectors
- Proximity in vector space equals "similarity"



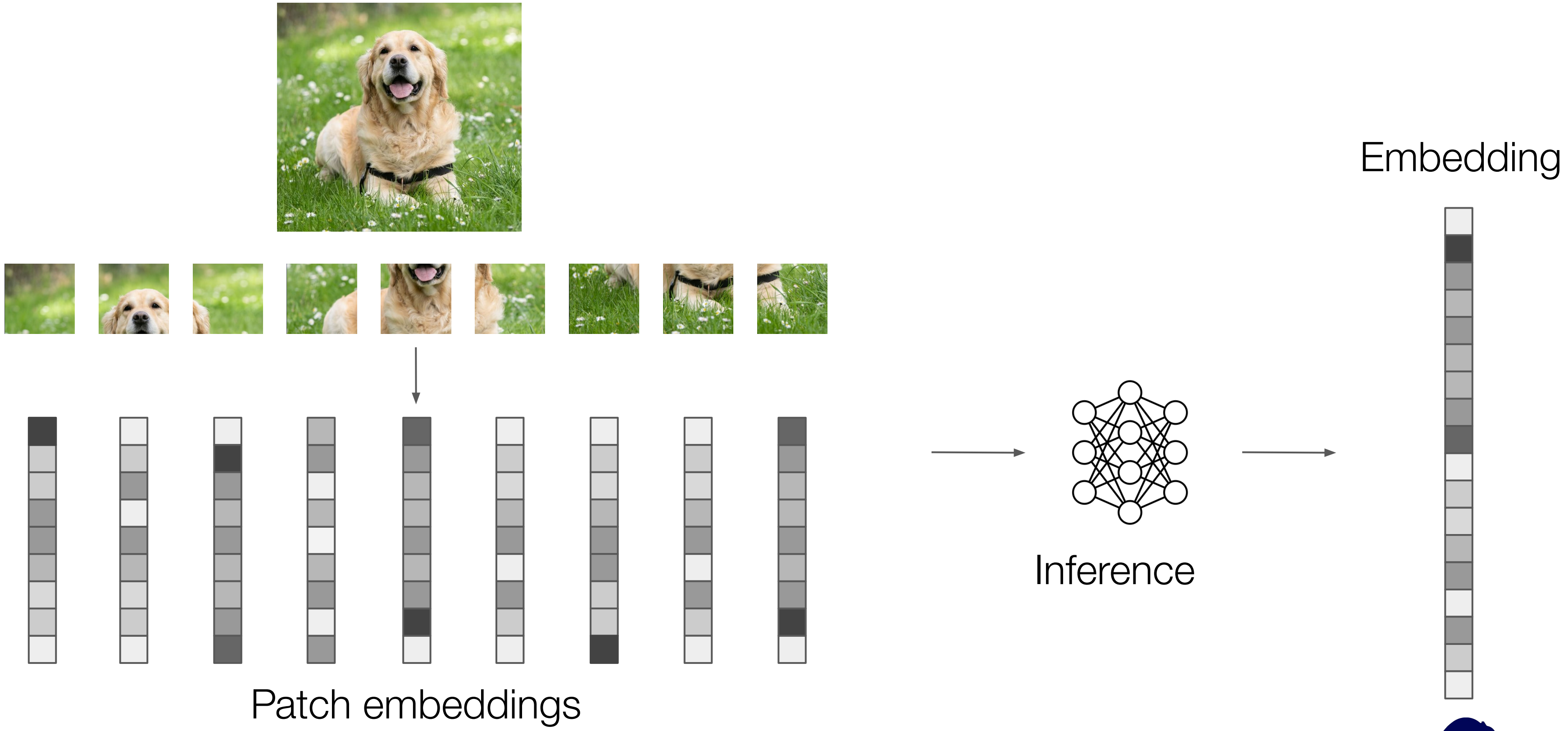


What are embeddings? How models understand text

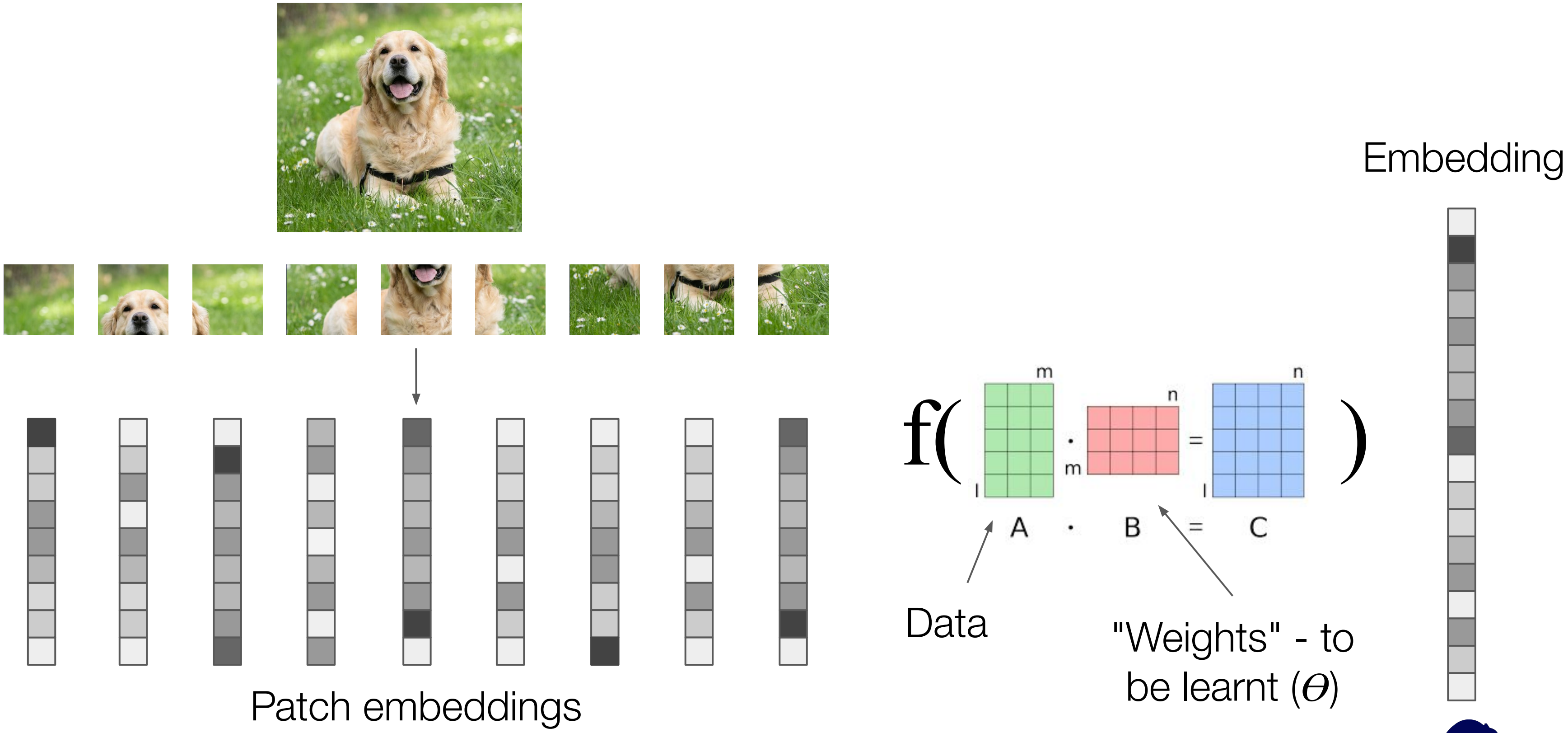


marqo

What are embeddings? How models understand images



What are embeddings? How models understand images



marqo

Embeddings as data structures 2.0?

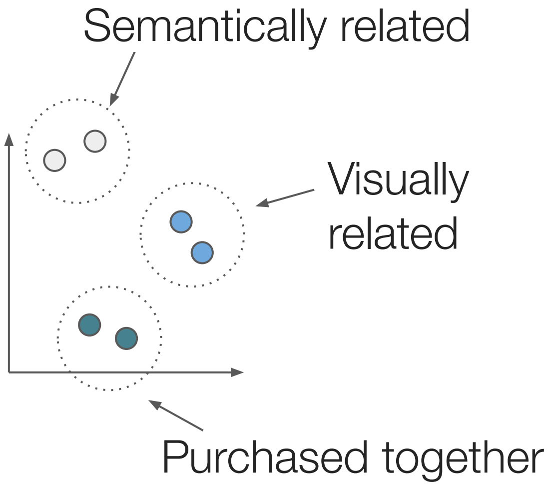
*"**Data structures 2.0** is written in human unfriendly language, such as the floating point values of an embedding. No human is involved in writing this code ... and coding directly in the floating point values is kind of tedious but possible (I tried)." [2]*

$$\underset{N_\theta,\theta}{\operatorname{argmin}}\, L(\mathbf{X}, \mathbf{Y}; N_\theta)$$

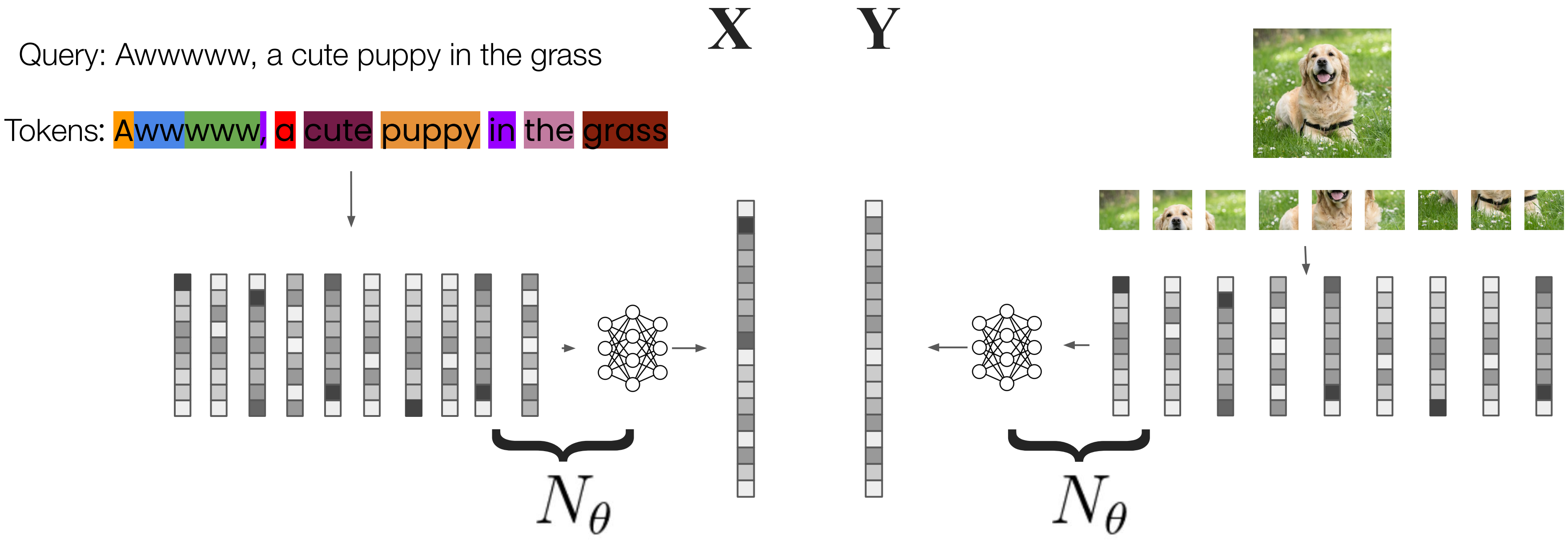
Network architecture

Parameters

Data describing the similarity to encode



Training embedding models $\underset{N_\theta,\theta}{\operatorname{argmin}} L(\mathbf{X}, \mathbf{Y}; N_\theta)$



Make these embeddings as close as possible

$$\operatorname{argmin} L$$

marqo

Marqo FashionCLIP

State-of-the-art embeddings model for fashion search and recommendations

https://www.marqo.ai/blog/search-model-for-fashion

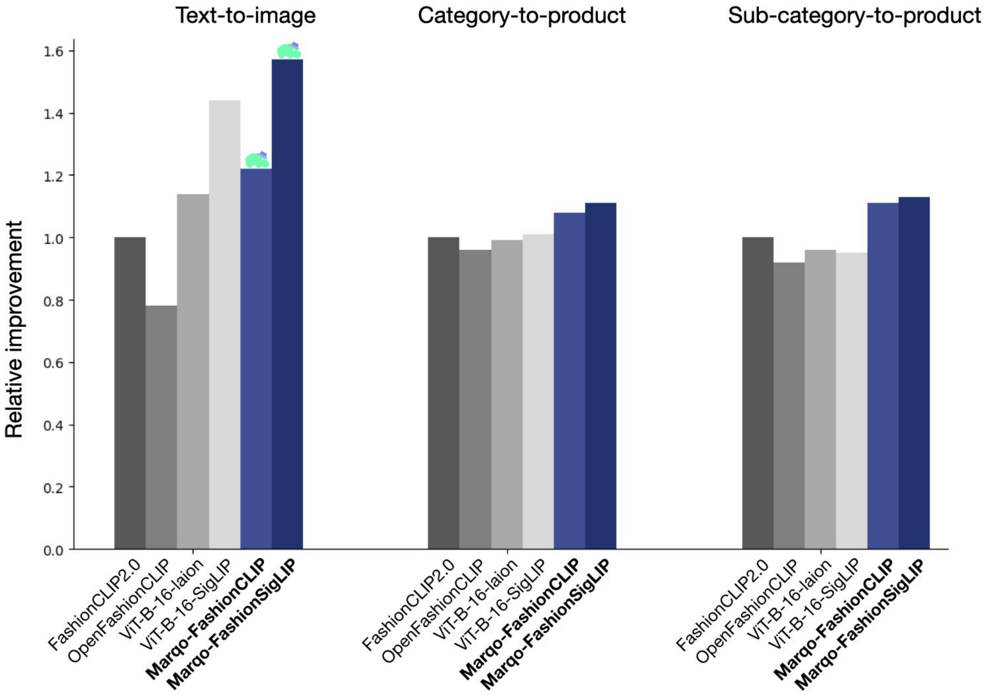
$$L(\text{image, text}) +$$
$$L(\text{image, colors}) +$$
$$L(\text{image, materials}) +$$
$$L(\text{image, style}) +$$
$$L(\text{image, keywords})$$

Midriff Waist Shaper Dress
Colors: blue, purple, pink
Materials: cotton, linen
Style: casual, spring/summer
Keywords: Floral midi dress, blue floral dress

marqo

Marqo FashionCLIP

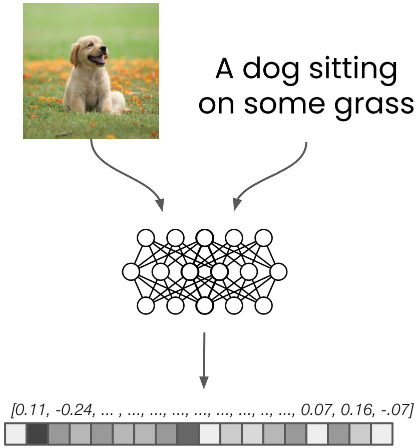
State-of-the-art embeddings model for fashion search and recommendations



marqo

Summary of software 2.0 and embeddings

- Software 2.0
  - learns the functions from data instead of being handcrafted
- Embeddings
  - learn the representation of data instead of being handcrafted
- Embeddings are learned vectors
- Embeddings provide a consistent format for diverse unstructured data



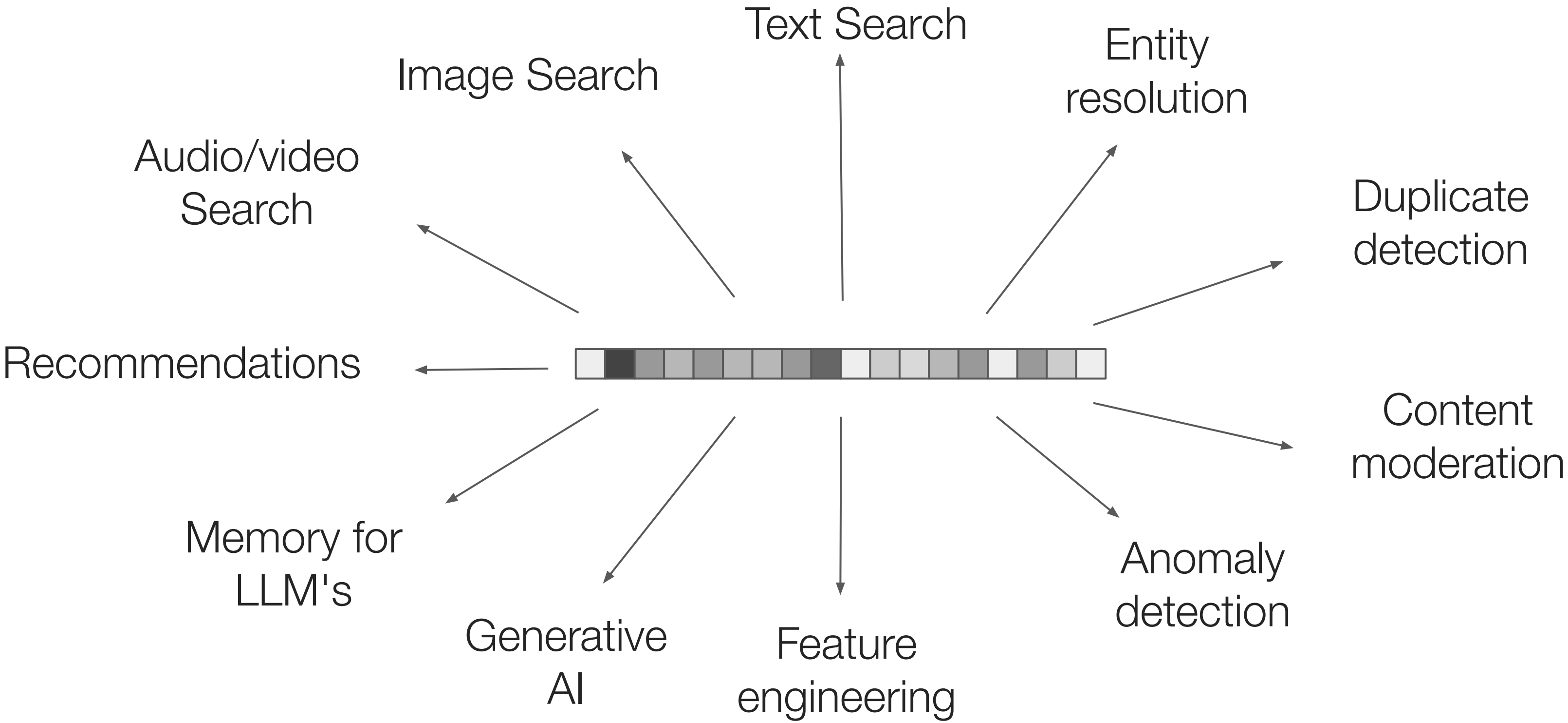
$$\underset{N_\theta,\theta}{\operatorname{argmin}}\, L(\mathbf{X}, \mathbf{Y}; N_\theta)$$

Building Software 2.0 with embeddings

Outline

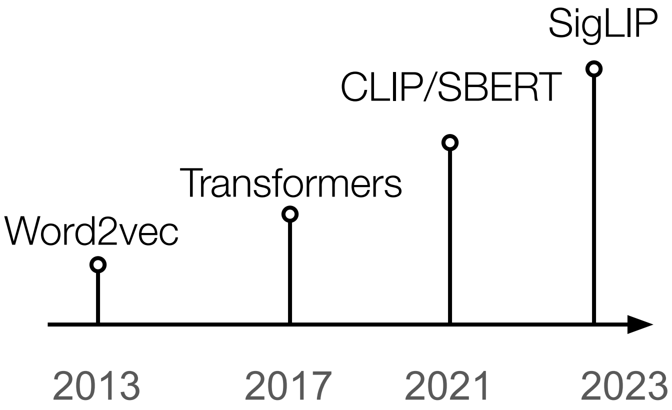
- What is Software 2.0?
  - Examples
- What are embeddings?
  - Producing embeddings
  - Training models for embeddings
- Building with embeddings
  - Classification, search and RAG

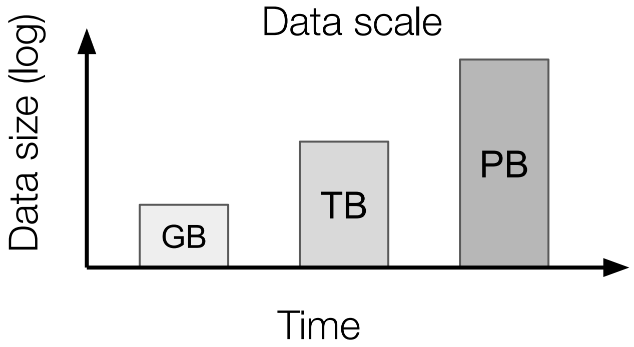
Building with Embeddings

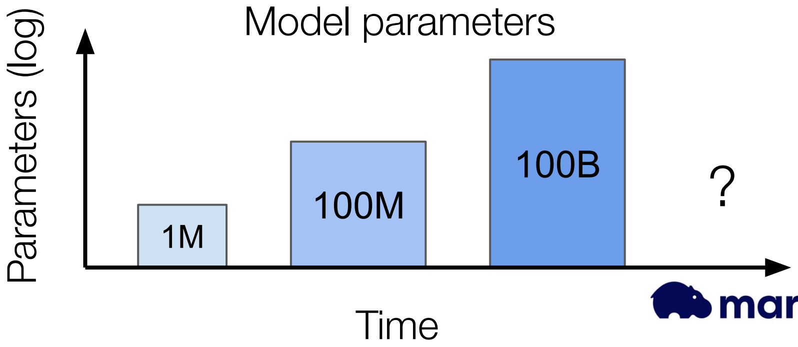


Building with Embeddings - why now?

- Embedding quality has dramatically improved
- Driven by:
    - data quality and quantity
    - training methods
    - hardware improvements
    - architecture improvements

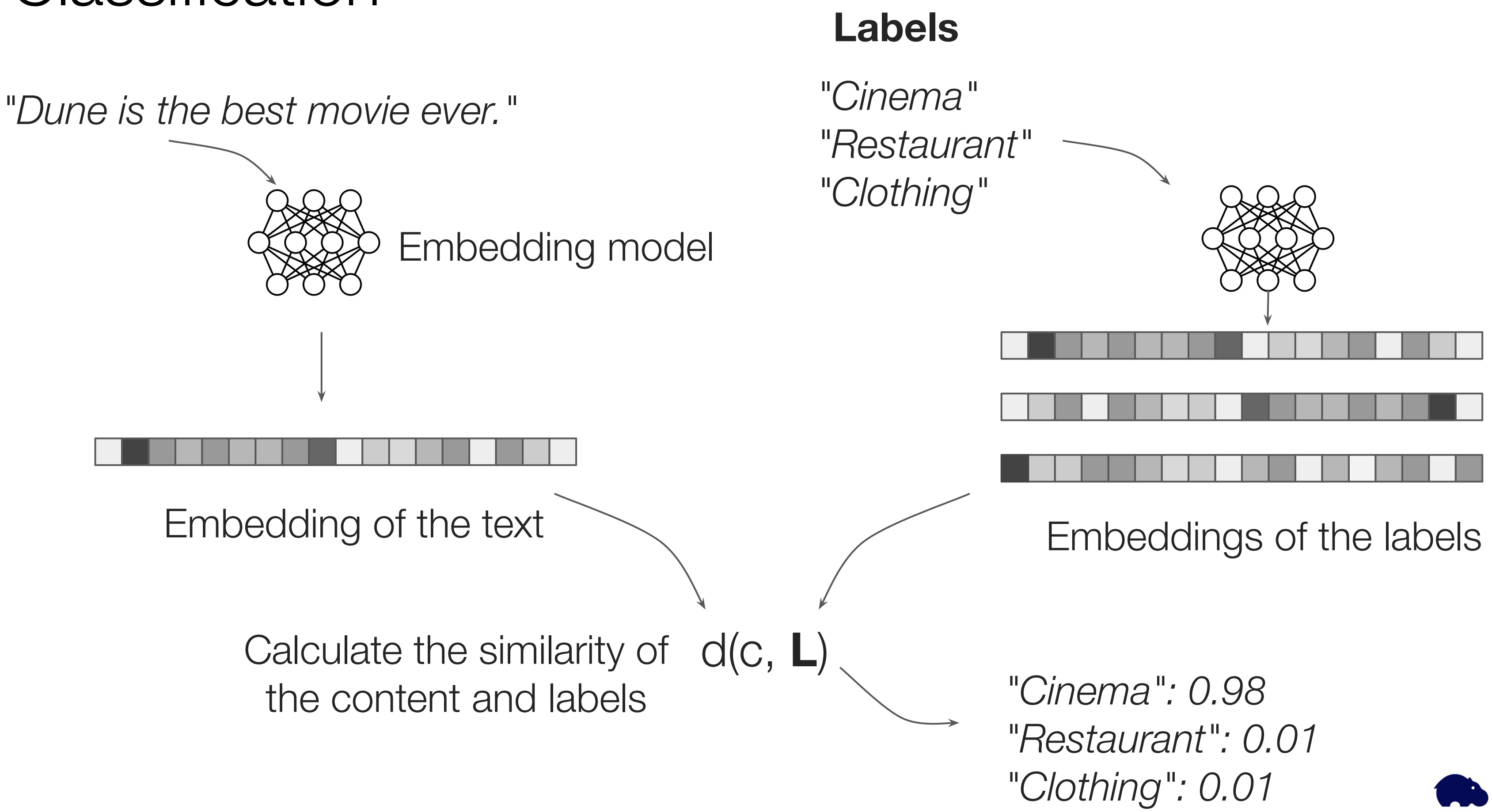




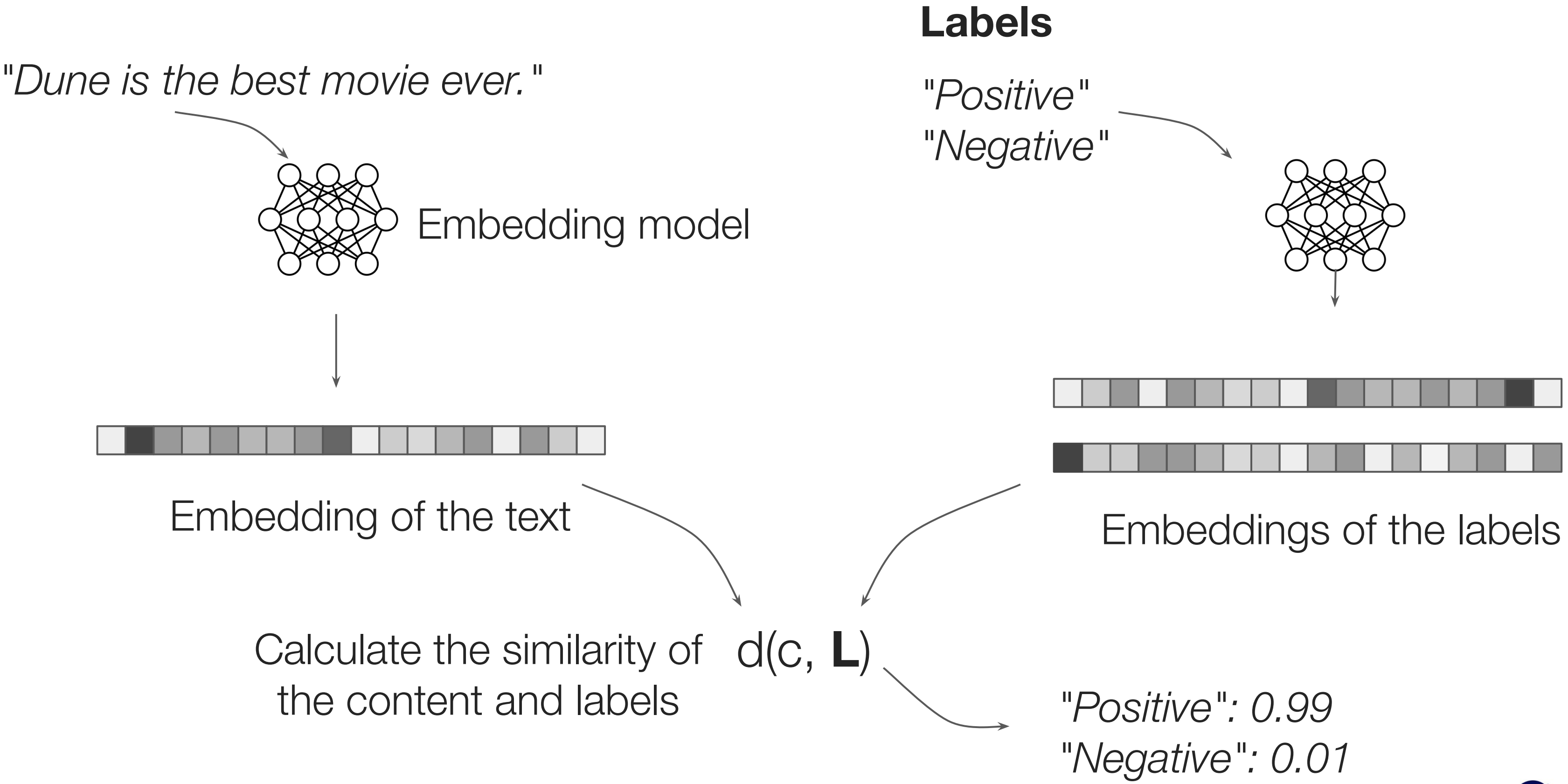


marqo

Classification

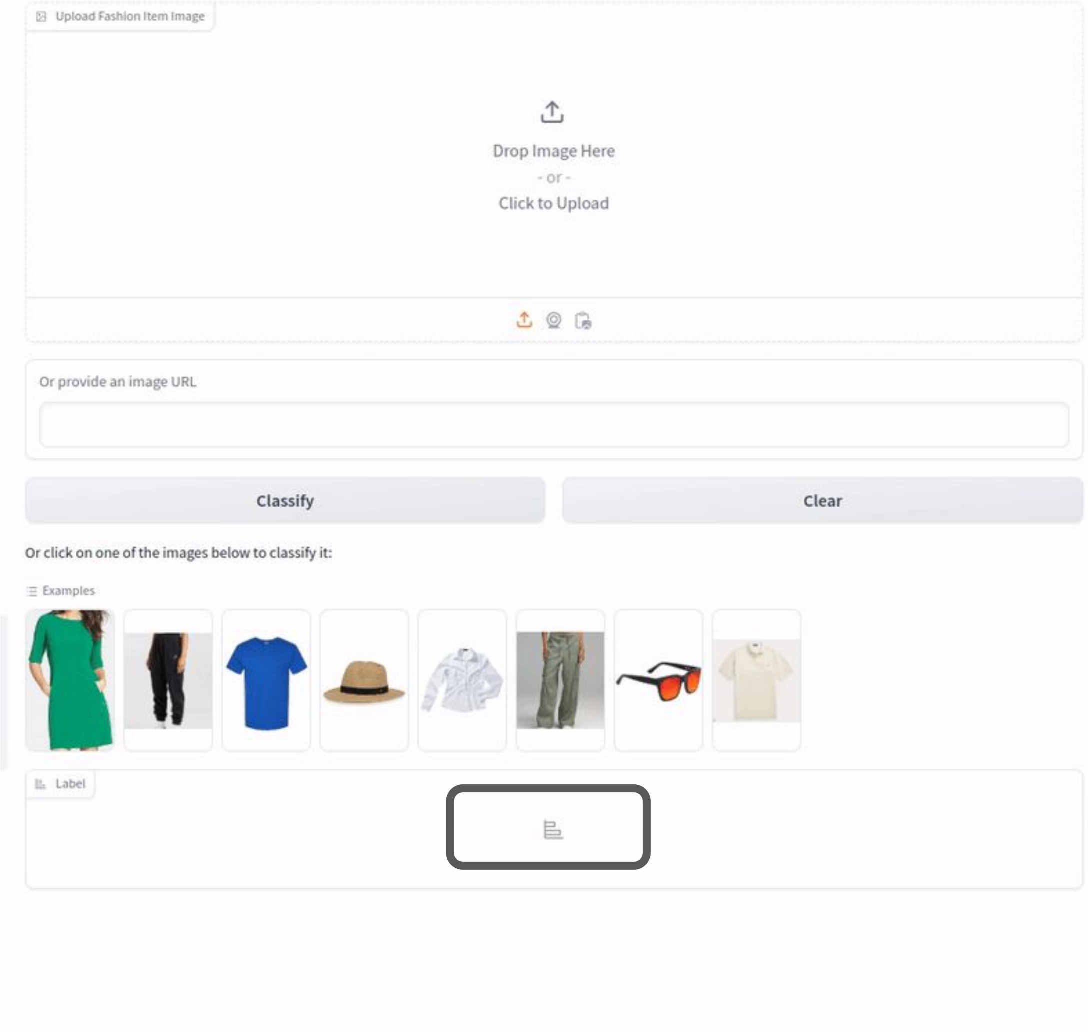


Classification



marqo

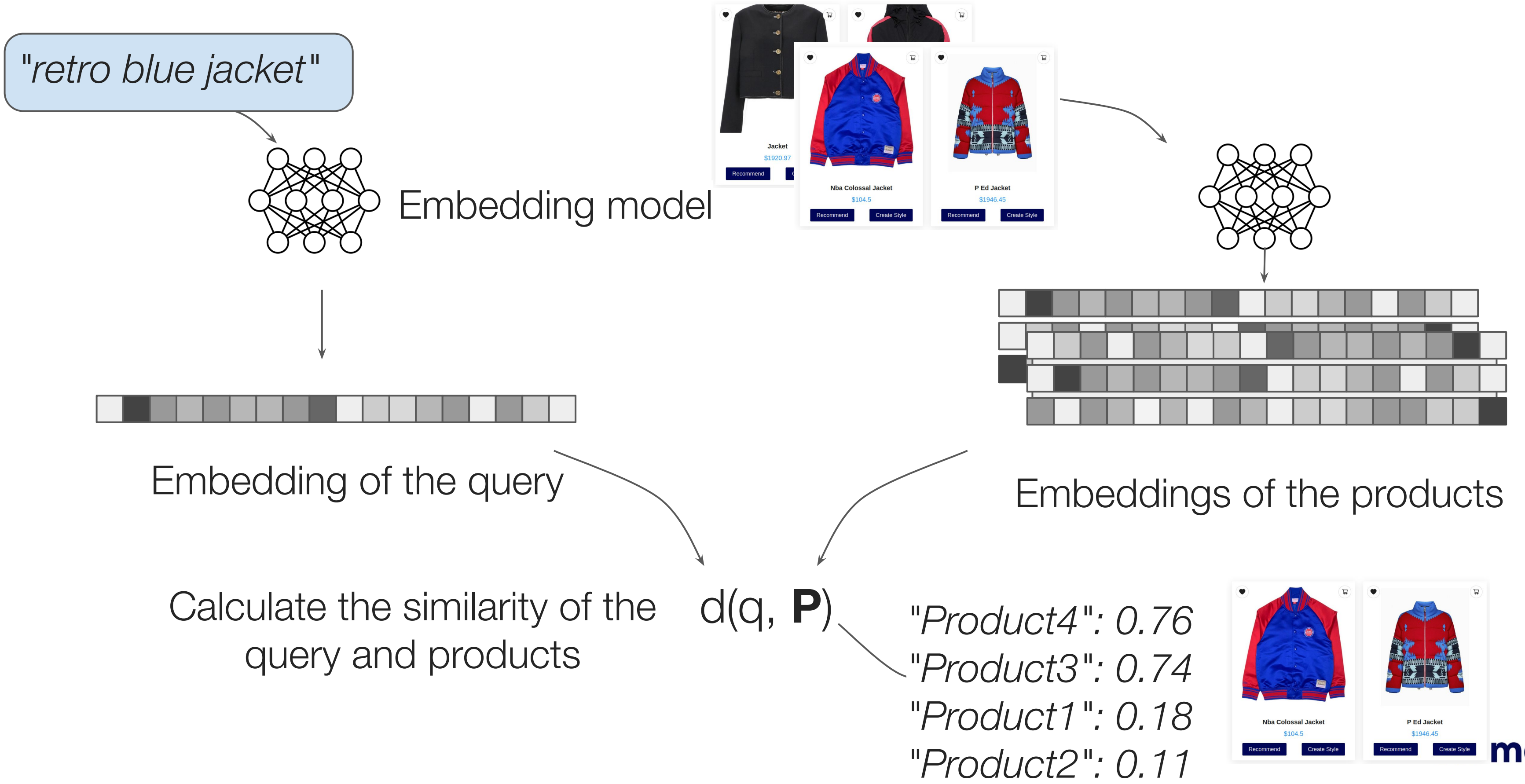
```python
# Define the query and classification labels
query = ["Dune is the best movie ever."]
labels = ["Cinema", "Food", "Pet food"]

# Step 1: Get the embedding for the query
query_embedding = get_embedding(query) # type: ignore

# Step 2: Get the embeddings for each classification label
label_embeddings = get_embedding(labels) # type: ignore

# Step 3: Calculate cosine similarity between the query embedding and each label embedding
# This returns a similarity score for each label
similarities = cosine_similarity(query_embedding, label_embeddings)

# Step 4: Find the label with the highest similarity score
max_similarity_index = np.argmax(similarities)
predicted_label = labels[max_similarity_index]

# Step 5: Print the result
print(f"The predicted label for the query is: {predicted_label}")
```



https://huggingface.co/spaces/Marqo/Marqo-FashionSigLIP-Classification

marqo

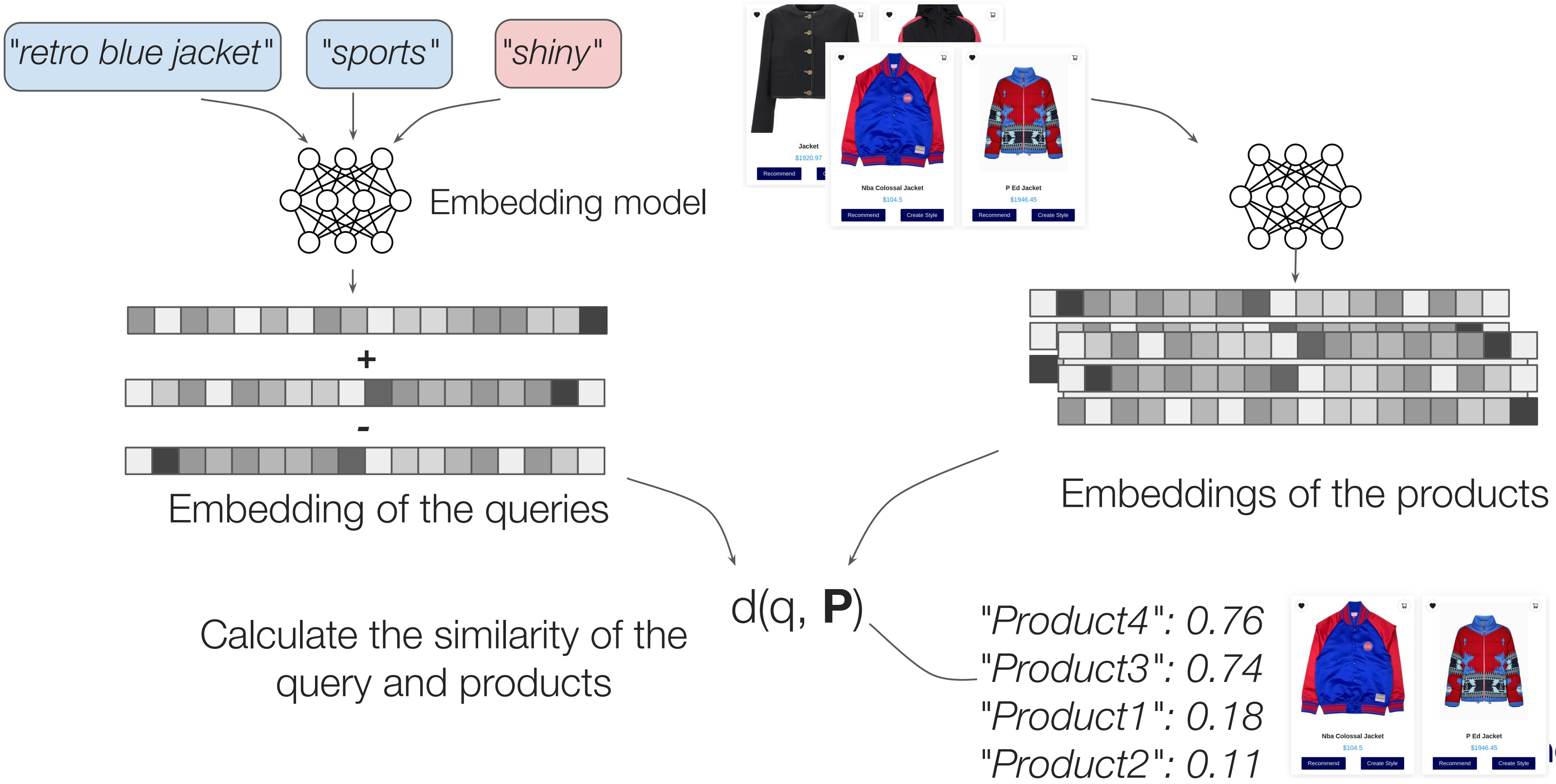
Search



marqo


Search for this...
More of this...
Less of this...
Filters
Score Modifiers
Search
Search as you type
Show live recommendations

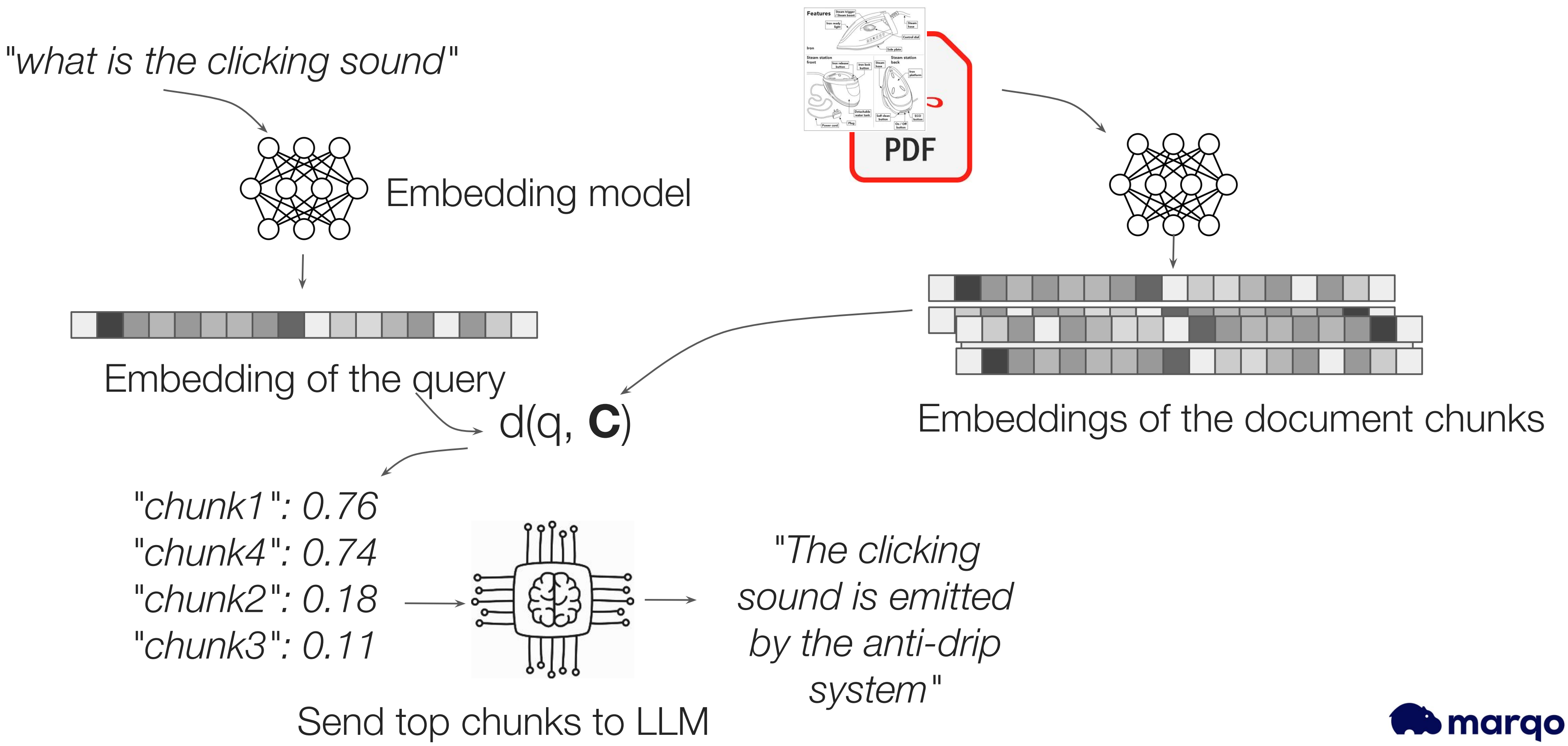
Search



"retro blue jacket"
"sports"
"shiny"
Embedding model
+
-
Embedding of the queries
Jacket
$1920.97
Recommend
Nba Colossal Jacket
$104.5
Recommend
Create Style
P Ed Jacket
$1946.45
Recommend
Create Style
Embeddings of the products
d(q, **P**)
Calculate the similarity of the query and products
"Product4": 0.76
"Product3": 0.74
"Product1": 0.18
"Product2": 0.11
Nba Colossal Jacket
$104.5
Recommend
Create Style
P Ed Jacket
$1946.45
Recommend
Create Style

```python
import marqo
from marqo import Client

# initilize the client
mq = Client("https://api.marqo.ai", api_key=api_key)

# create an index
mq.create_index(index_name,
        model="open_clip/ViT-B-32/laion2b_s34b_b79k")

# embed content
mq.index(index_name).add_documents(content)

# perform a search query on the index
results = mq.index(index_name).search(q="retro blue jacket")
```

Retrieval augmented generation (RAG)

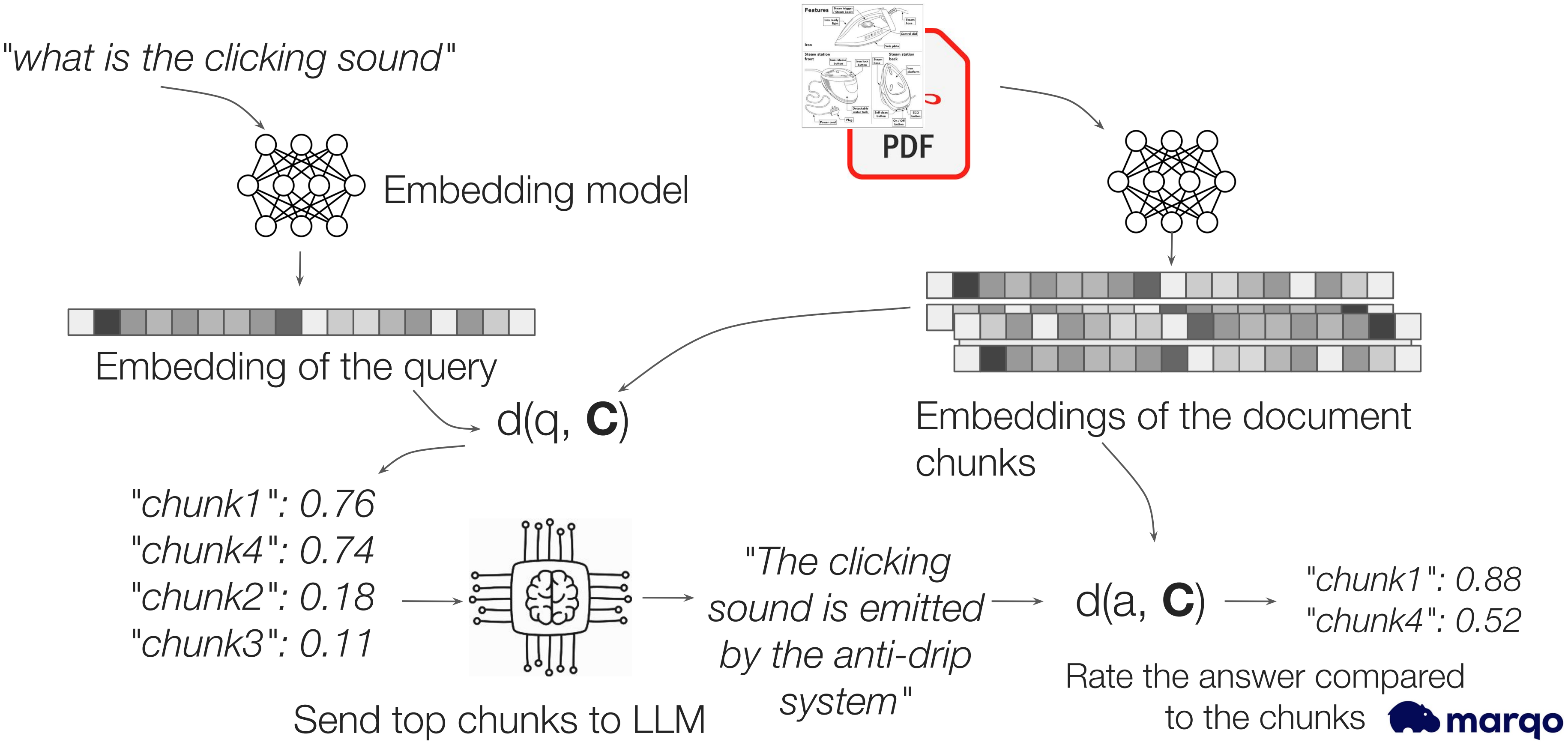


marqo

Please enter some text to start searching...

marqo

Retrieval augmented generation (RAG)



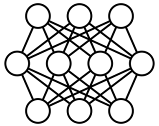
Building with Embeddings

- Embedding based systems share a lot of similarities
- Embedding generation (inference) and comparing (scoring) make up the core
- Embedding generation handled by deep learning models
- The comparison (retrieval) of embeddings is handled by a vector database

Train

$$\underset{N_\theta,\theta}{\operatorname{argmin}} L(\mathbf{X}, \mathbf{Y}; N_\theta)$$

Generate



Retrieve

d(q, **C**)

Marqo Embeddings Cloud

Train

Multimodal fine-tuning, optimised for user interaction data.

```
model_details =
marqtune_client.train(
    method="weighted_clip",

model="open_clip/ViT-L-14/laion2b_
s32b_b82k",
    data={
        "id": "your_data_uuid",
        "bucket_path":
"path_in_bucket"
    },
    params={
        "param1": "value1",
        "param2": "value2"
    }
)
```

Generate

Marqo's open source inference engine provides fast image ingestion, inference and search.

```
mq.create_index("my-index",
model="my-custom-CLIP")

mq.index("my-index").add_documents
(
    [{
        "title": "The Travels of
Marco Polo",
        "image": "my-image-url"
    }],
    tensor_fields=["title",
"image"]
)
```

Retrieve

Marqo provides a fully-managed Vespa vectorDB engine, deployed with Marqo inference nodes for low latency search and ingestion.

```
results =
mq.index("my-index").search(
    "Marqo Polo"
)
```

Building Software 2.0 with embeddings

Outline

- What is Software 2.0?
  - Examples
- What are embeddings?
  - Producing embeddings
  - Training models for embeddings
- Building with embeddings
  - Classification, search and RAG

marqo

Conclusion

- Software 2.0:
  - Uses optimization to learn functions from data
  - Allows for increasingly complex programs
- Embeddings:
  - The data structures 2.0
  - Uses optimization to learn representations from data
  - Interface with unstructured data
  - Allow for composability
- Embedding models share a lot of similarities with each other
- Embedding based systems share a lot of similarities with each other

$$\underset{N_\theta,\theta}{\operatorname{argmin}} L(\mathbf{X}, \mathbf{Y}; N_\theta)$$

Thank you!

jesse@marqo.ai

Marqo
https://www.marqo.ai/

Open source
https://github.com/marqo-ai/marqo

Classification demo
https://huggingface.co/spaces/Marqo/Marqo-FashionSigLIP-Classification

FashionCLIP
https://huggingface.co/Marqo/marqo-fashionSigLIP

Blog
https://www.marqo.ai/blog

marqo

marqo