AI Development 101 with Cloudflare AI

I'm Kristian 👋

@kristianf_ on Twitter

Who am I?

- Developer Advocate at Cloudflare
- Teaching developers with videos/free courses (➡️ links.7.dev)
- Focus on AI development in 2024

Schedule

1. AI Development 101
    1. Models
    2. Embeddings
    3. Vector Databases
2. Live coding

Models

AI models are different algorithms that perform different tasks

---

Example: ChatGPT

ChatGPT is an *interface* to a bunch of low-level GPT models (GPT-3.5, GPT-4, etc.)

Workers AI Models

In Workers AI, we support a bunch of model categories:

- Automatic speech recognition
- Image/text classification
- Text-to-image
- (continued...)

Workers AI Models

- Text embeddings
- Text generation
- Translation

How to use Workers AI

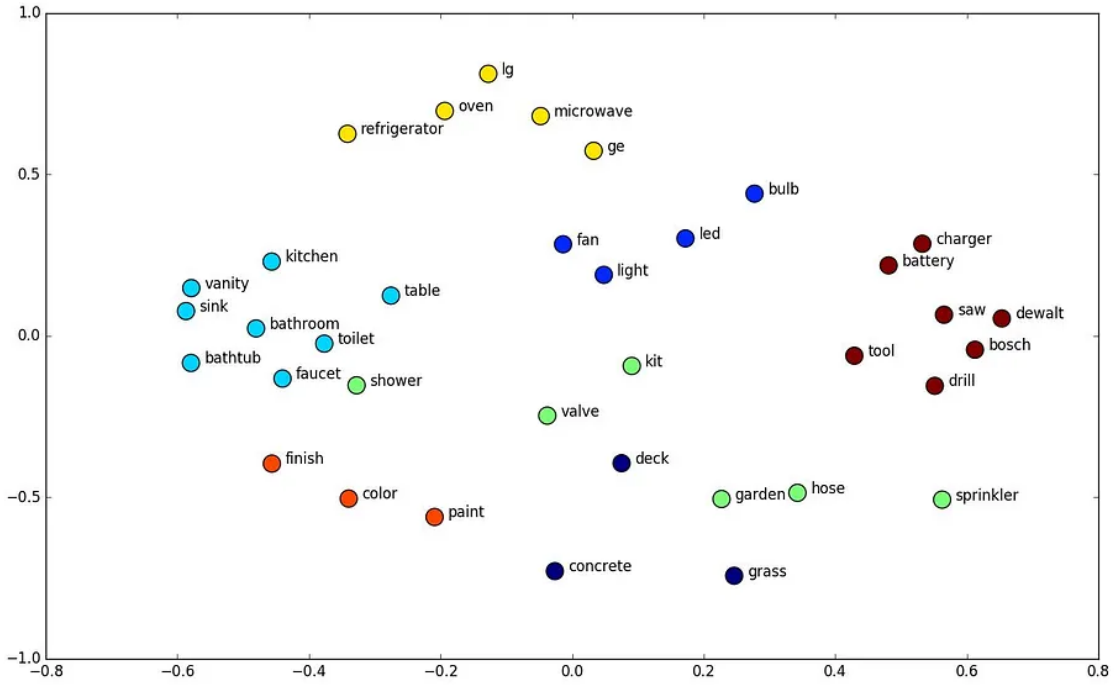
```
const aiClient = new Ai(env.AI)

const messages = [
  { role: 'system', content: 'You are a friendly assistant' },
  { role: 'user', content: 'What is the origin of the phrase Hello, World' }
]

const MODEL = '@cf/meta/llama-2-7b-chat-int8'
const { response } = await aiClient.run(MODEL, { messages })

// response: "The origin of the phrase \"Hello, World\" is not well-
documented, but it is believed..."
```

Embeddings

Embeddings measure how similar strings are.

Workers AI uses the `BAAI` models from HuggingFace.

Embeddings



How to use embeddings

```
const ai = new Ai(env.AI);

const stories = [
  'This is a story about an orange cloud',
  'This is a story about a llama',
  'This is a story about a hugging emoji'
]

const embeddings = await ai.run('@cf/baai/bge-base-en-v1.5', {
  text: stories
});
```

What does an embedding look like?

An embedding (in Workers AI) has:

- *shape*: a description of the vector size

- *data*: the vector representing the data

```
"shape": [1, 768],
"data": [
  [0.0319, 0.0060, 0.0259, ...]
]
```

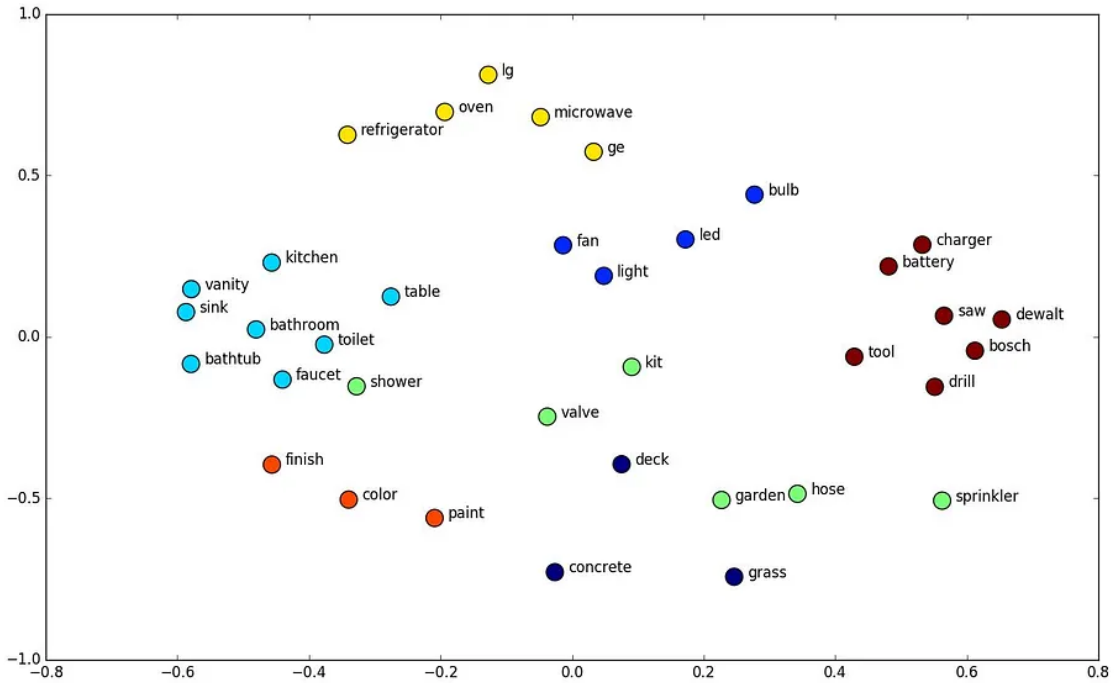
Embeddings are really just...

Vectors - a list of numerical coordinates indicating position along an axis.

In many applications in AI/machine learning, a coordinate is a *float* between 0 and 1.

Vectors indicate similarity

The closer that the vectors of two pieces of data are to each other, *the more similar they are*.



So, how do we know how similar these vectors are?

Vector databases

Vector databases

Vector databases store embeddings and allow quick retrieval/processing of similarity

---

Vector databases drive retrieval

It also allows us to store IDs or other identifying factors to relate to other data stores

```
{
  id: '1',
  values: [0.051, 0.003, 0.357],
  metadata: {}
}
```

```
const vectorIds = vectors.map(v => v.id)
const dbQuery = await database.query(
  `select * from items where id in ${vectorIds}`
)

// Items that correspond to the vector records
console.log(dbQuery.results)
```

Example of everything put together

Retrieval Augmented Generation

Retrieval Augmented Generation (RAG)

1. Take data and store it in a relational database (Cloudflare D1)
2. Generate vectors for those database records (Vectorize)
3. (continued...)

Retrieval Augmented Generation (RAG)

3. When a new query comes in (Workers AI)
    1. Generate vectors for the query (Workers AI embedding model)
    2. Look up similar, known vectors to the query using a vector database (Vectorize)
    3. Get a list of relevant records in the database (Cloudflare D1)
4. Augment the query with relevant information

Querying without RAG

"Q: Who won the 2023 NBA Championship?"

"A: I don't have access to that information, as my cut-off date for information is July 2022."

Inject data

1. Add a piece of data to DB
2. Inject it as **context** into the query

Querying with RAG

```
Context: The Denver Nuggets won the 2023 NBA Championship after defeating
the Miami Heat in four of five games.
Q: Who won the 2023 NBA Championship?"
```

"A: The Denver Nuggets won the 2023 NBA Championship."

Live coding

Thanks!

Link to source code: https://github.com/kristianfreeman/that-conf-rag-2023