



Georg Jung  
Software Architekt

On-device, smarter und präzise

# Semantische Suche mit HyDE und Phi-3

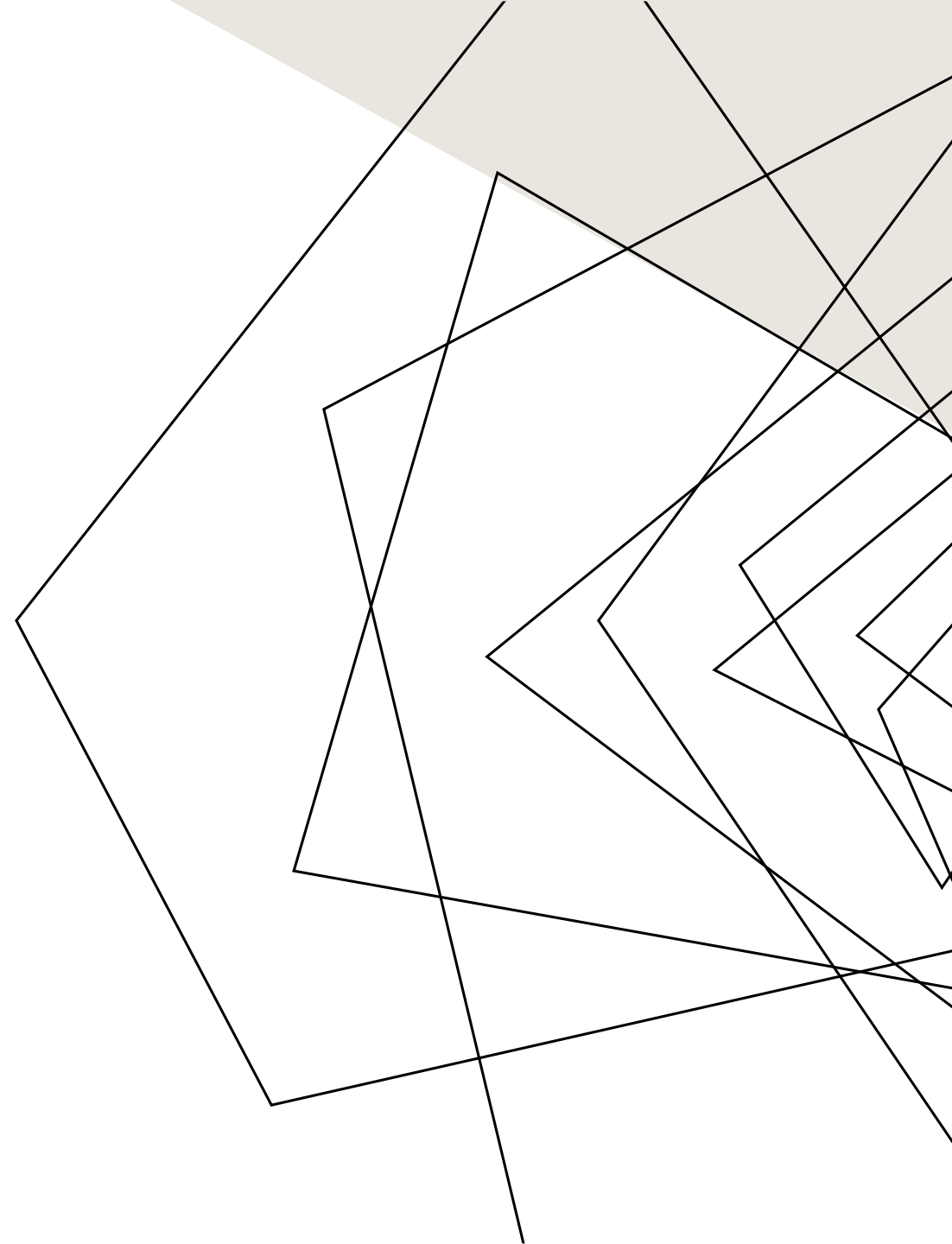
# GEORG JUNG

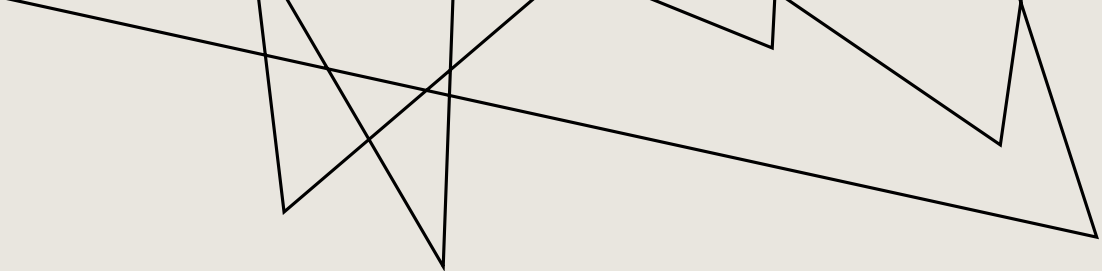
- C# & .NET
- AI in .NET produktiv nutzbar machen
- Open Source
  - FastBertTokenizer
  - FaceAiSharp
  - rund 1,5 Mio. NuGet Downloads

[georg@gjung.com](mailto:georg@gjung.com)

[linkedin.com/in/georg-jung/](https://linkedin.com/in/georg-jung/)

[github.com/georg-jung/](https://github.com/georg-jung/)



- 
- ✓ PoV: Software-Entwickler
  - ✓ HyDE verstehen
  - ✓ Alle Bausteine überblicken
  - ✓ Praktische Vorgehensweise verstehen
  - ✓ On-device: keine Cloud-Dienste nötig
- 
- × AI-Interna & Details
  - × Fertige Produkte

# ERWARTUNGEN

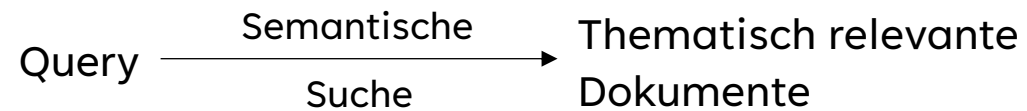
# SEMANTISCHE SUCHE

- Gleiche Bedeutung, statt gleiche Buchstaben
- Arbeitet mit Embeddings
- Gefühl: Googlen vs. Strg+F

# SEMANTISCHE SUCHE

- Gleiche Bedeutung, statt gleiche Buchstaben
- Arbeitet mit Embeddings
- Gefühl: Googlen vs. Strg+F

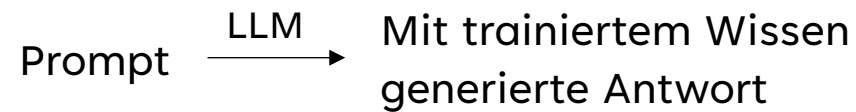
➤ Bibliothekar



# LARGE LANGUAGE MODEL

- ✓ Kreativität
- ✓ Kontextverständnis
- ✓ Mehrere Quellen kombinieren

- × Eigene Daten / firmeninternes Wissen
- × Halluzinationen
- × Nachvollziehbare Quellen



# LARGE LANGUAGE MODEL

- ✓ Kreativität
- ✓ Kontextverständnis
- ✓ Mehrere Quellen kombinieren

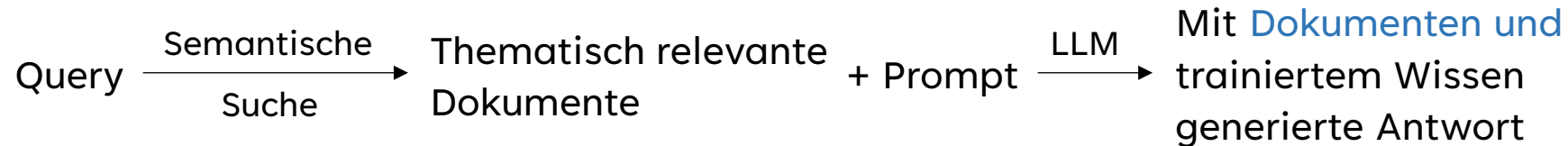
- × Eigene Daten / firmeninternes Wissen
- × Halluzinationen
- × Nachvollziehbare Quellen

➤ „Dorfältester“



# RAG?

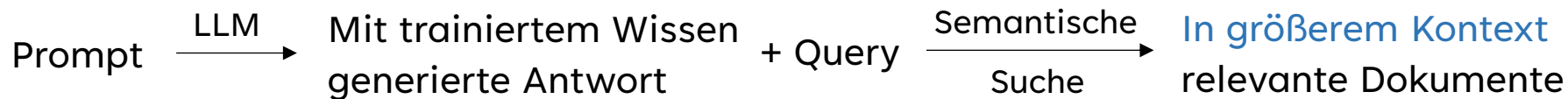
- Retrieval Augmented Generation
- Erst semantisch Suchen, dann Ergebnisse in LLM-Prompt integrieren
- ✓ Präzisere Antworten/weniger Halluzinationen
- ✓ Eigene Informationen beisteuern





# HyDE!

- Hypothetical Document Embeddings
- „Erfundene“ LLM-Antwort als Embedding nutzen
  - Den Dokumenten strukturell ähnlicher als ein Query -> Bessere Embeddings
- ✓ Original-Quellen finden
- ✓ LLM-Kreativität & -Kontextwissen für Suche nutzen



# RAG vs. HyDE

## RAG

- Individuelle Antwort
- Steht & fällt mit Qualität der semantischen Suche
- Präziser/halluzinationsärmer

## Semantische Suche mit HyDE

- Bessere Ergebnisse durch mehr Kontext aus der LLM-Antwort
- Halluzinationsfrei & Original-Quellen



# Demo: Ziel?

- Simple semantische „Suchmaschine“ auf lokalem Datensatz
  - Hier: .NET-, ASP.NET-, EF Core & Npgsql-Dokumentation
- Ausführung vollständig on-device
- Wir implementieren alle interessanten Schritte

## Semantic Search

Query

Configure n-to-m relation

Search!

Show hypothetical document

### [Many-to-many relationships - EF Core](#)

#### Many-to-many relationships

Many-to-many relationships are used when any number entities of one entity type is associated with any number of entities of the same or another entity type. For example, a `Post` can have many associated `Tags`, and each `Tag` can in turn be associated with any number of `Posts`.

#### Understanding many-to-many relationships

Many-to-many relationships are different from [one-to-many](#) and [one-to-one](#) relationships in that they cannot be represented in a simple way using just a foreign key. Instead, an additional entity type is needed to "join" the two sides of the relationship. This is known as the "join entity type" and maps to a "join table" in a relational database. The

Similarity: 0.86

### [Eager Loading of Related Data - EF Core](#)

#### Eager Loading of Related Data

##### Eager loading

You can use the `Include` method to specify related data to be included in query results. In the following example, the blogs that are returned in the results will have their `Posts` property populated with the related posts.

```
[code-csharp]Main
```

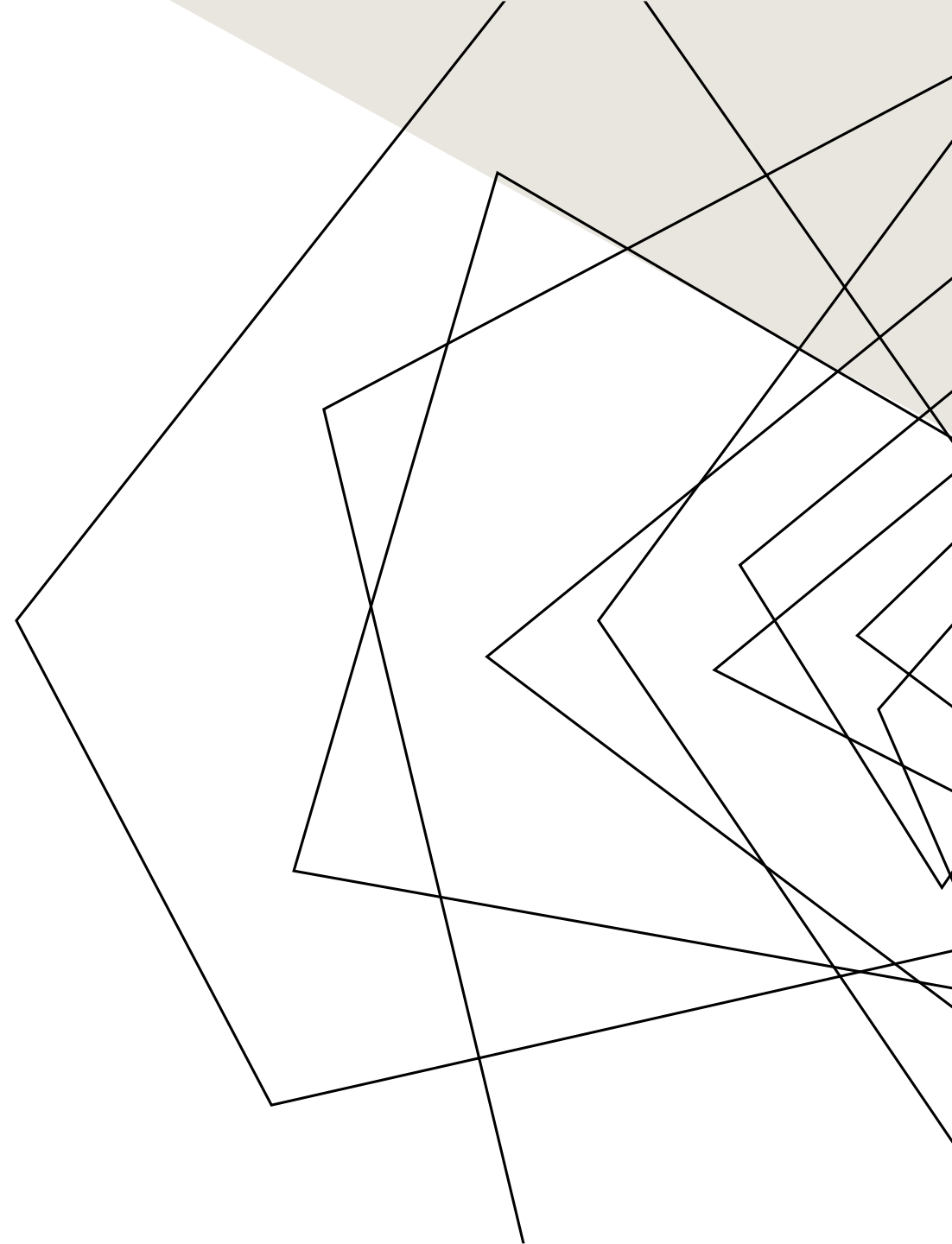
Similarity: 0.84

### [Loading Related Data - EF Core](#)

Loading Related Data

# Projektplan

1. Modellauswahl
2. Semantische Embeddings berechnen
3. Ähnliche Vektoren finden
4. HyDE ergänzen



# Modellauswahl

- Embedding-Modell
  - Vektorrepräsentation der Corpus-Dokumente (hier: Dokumentation) erzeugen
  - Vektorrepräsentation des Search Query erzeugen
- Language Model
  - Hypothetisches Dokument aus Search Query erzeugen
- Trade-Off: Qualität der Ergebnisse vs. benötigte Rechenleistung & (V)RAM

# Modellauswahl: Kriterien

- Ranking/Ergebnisqualität
- Speicherbedarf (RAM/VRAM)
- Kontextgröße
- Embeddingdimensionen
- Format (ONNX?)
- Wo ausführbar (CUDA, DirectML, ...)?
- Modellarchitektur
- Lizenz
- Dateigröße
- Anzahl Modellparameter

# Embedding-Modell: MTEB-Leaderboard

- Massive Text Embedding Benchmark

Rank ▲	Model ▲	Model Size (Million Parameters) ▲	Memory Usage (GB, fp32) ▲	Embedding Dimensions ▲	Max Tokens ▲	Average (56 datasets) ▲
6	<a href="#">dunzhang-stella-en-400M-v5</a>	435	1.62	1024	8192	70.11
7	<a href="#">stella-en-400M-v5</a>	435	1.62	8192	8192	70.11
23	<a href="#">jina-embeddings-v3</a>	572	2.13	1024	8194	65.51
24	<a href="#">gte-large-en-v1.5</a>	434	1.62	1024	8192	65.39
26	<a href="#">cde-small-v1</a>	143	0.53	768	512	65
29	<a href="#">mxbai-embed-large-v1</a>	335	1.25	1024	512	64.68
30	<a href="#">UAE-Large-V1</a>	335	1.25	1024	512	64.64
34	<a href="#">multilingual-e5-large-instruct</a>	560	2.09	1024	514	64.41
36	<a href="#">GIST-large-Embedding-v0</a>	335	1.25	1024	512	64.34
37	<a href="#">bge-large-en-v1.5</a>	335	1.25	1024	512	64.23

# bge-small-en-v1.5

- Nur Englisch
- Fokus hauptsächlich Effizienz
- In verschiedenen „Intelligenz“ vs. Effizienz Trade-offs verfügbar
- Klassische BERT-Architektur



# Language Model: Open LLM Leaderboard oder Chatbot Arena

T	Model	Average	Hub License	#Params (B)
	Goekdeniz-Guelmez/Josiefied-Qwen2.5-7B-Instruct-abliterated-v2	27.76	apache-2.0	7
	BAAI/Infinity-Instruct-3M-0625-Yi-1.5-9B	27.74	apache-2.0	8
	cognitivecomputations/dolphin-2.9.1-yi-1.5-34b	27.73	apache-2.0	34
	01-ai/Yi-1.5-9B-Chat	27.71	apache-2.0	8
	jpacifico/Chocolatine-3B-Instruct-DPO-Revised	27.63	mit	3
	Gryphe/Pantheon-RP-Puze-1.6.2-22b-Small	27.58	other	22
	nbeerbower/Mistral-Small-Gutenberg-Doppel-22B	27.58	other	22
	cloudyu/Mixtral 34Bx2 MoE 60B	27.42	apache-2.0	60
	microsoft/Phi-3.5-mini-instruct	27.4	mit	3
	NAPS-ai/naps-llama-3.1-8b-instruct-v0.4	27.31	apache-2.0	8

[huggingface.co/spaces/open-llm-leaderboard/open\\_llm\\_leaderboard](https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard)

Rank* (UB)	Rank (StyleCtrl)	Model	Arena Score	95% CI	Votes	Organization	License	Knowledge Cutoff
1	1	ChatGPT-4o-latest (2024-09-03)	1339	+4/-4	28488	OpenAI	Proprietary	2023/10
1	1	o1-preview	1335	+4/-5	17562	OpenAI	Proprietary	2023/10
3	3	o1-mini	1313	+4/-4	17919	OpenAI	Proprietary	2023/10
3	3	Gemini-1.5-Pro-002	1305	+5/-4	11430	Google	Proprietary	Unknown
4	3	Gemini-1.5-Pro-Exp-0827	1299	+4/-3	32437	Google	Proprietary	2023/11
6	8	Grok-2-08-13	1291	+3/-3	35661	xAI	Proprietary	2024/3
6	9	Yi-Lightning	1287	+5/-3	13262	01 AI	Proprietary	Unknown
7	5	GPT-4o-2024-05-13	1285	+3/-2	99251	OpenAI	Proprietary	2023/10
9	15	GLM-4-Plus	1274	+5/-5	13674	Zhipu AI	Proprietary	Unknown
9	17	GPT-4o-mini-2024-07-18	1274	+4/-3	38831	OpenAI	Proprietary	2023/10
9	13	Gemini-1.5-Flash-Exp-0827	1269	+3/-4	25555	Google	Proprietary	2023/11
9	20	Gemini-1.5-Flash-002	1269	+8/-5	8957	Google	Proprietary	Unknown

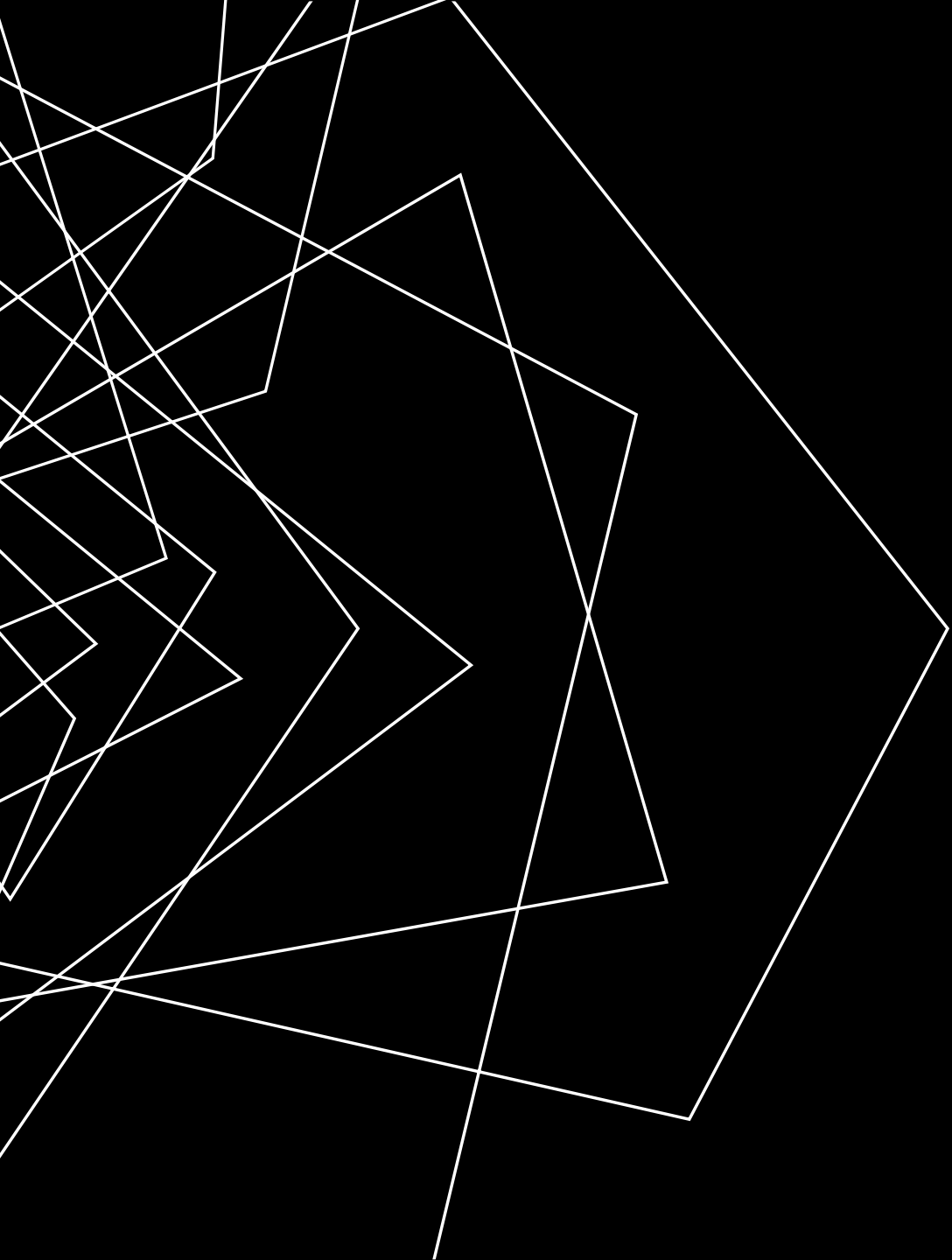
[lmarena.ai](https://lmarena.ai)

# Phi-3

- Gefühl: „Kostenloses lokales Mini-ChatGPT“
- Durchaus intelligent, weniger wissend
- Auf normalen Notebooks problemlos nutzbar
- Von Microsoft, unter MIT-Lizenz

# Projektplan

1. Modellauswahl ✓
2. Semantische Embeddings berechnen
3. Ähnliche Vektoren finden
4. HyDE ergänzen



DEMO

# Projektplan

1. Modellauswahl ✓
2. Semantische Embeddings berechnen ✓
3. Ähnliche Vektoren finden ✓
4. HyDE ergänzen

Bisher: knapp 75 Lines of Code

# Projektplan

1. Modellauswahl ✓
2. Semantische Embeddings berechnen ✓
3. Ähnliche Vektoren finden ✓
4. HyDE ergänzen ✓

Insgesamt: unter 140 Lines of Code

An abstract geometric pattern consisting of numerous white lines of varying lengths and orientations, creating a complex, overlapping web of shapes. The pattern is concentrated on the left side of the image, with lines extending towards the center.

# DEMO

[github.com/georg-jung/HyDE-with-Phi3-talk](https://github.com/georg-jung/HyDE-with-Phi3-talk)

# Perspektive & Herausforderung: Kombination der Bausteine

- Semantische Suche mit HyDE?
- Semantische Suche mit HyDE, danach RAG?
- Semantische Suche mit HyDE mit RAG?
- Semantische Suche mit HyDE mit RAG, danach RAG?
- Mehrere Iterationen?



# Perspektive & Herausforderung: Kombination der Bausteine

<b>Semantische Suche mit HyDE</b>	
<b>Semantische Suche mit HyDE, danach RAG</b>	
<b>Semantische Suche mit HyDE mit RAG, danach RAG</b>	
<b>Mehrere Iterationen?</b>	

# Perspektive: Zukünftige Modelle

- Es werden stetig sowohl „schlauere“ als auch effizientere Modelle veröffentlicht.
  - insbesondere auch frei nutzbare
- bspw. bge-large war vor gut einem Jahr MTEB-Leader, jetzt Platz 38

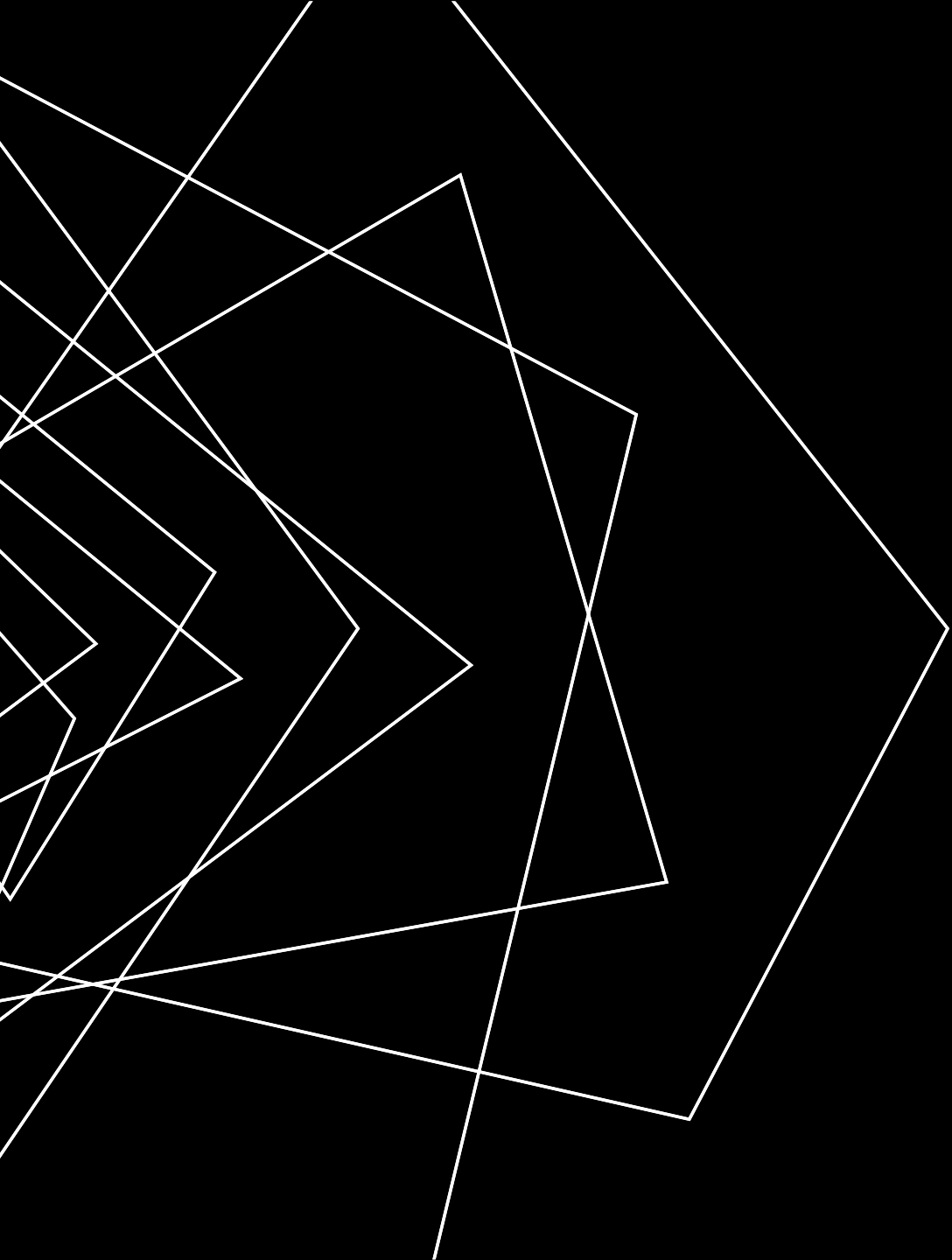
# Perspektive: Neue Software und APIs

- [SemanticKernel](#)
- [SmartComponents](#)
- [onnxruntime-genai](#)

Recipient	Address	
First name	Line 1	
Last name	Line 2	
Phone number	City	State
	Zip	Country
<div>Submit Smart Paste</div>		

# Perspektive: Neue Hardware

- Neue CPU-Generationen kommen mit integrierten NPUs
  - Mehr (i)GPUs werden mit DirectML nutzbar
- OnnxRuntime



# VIELEN DANK

Georg Jung

[georg@gjung.com](mailto:georg@gjung.com)

[github.com/georg-jung/](https://github.com/georg-jung/)

[linkedin.com/in/georg-jung/](https://linkedin.com/in/georg-jung/)