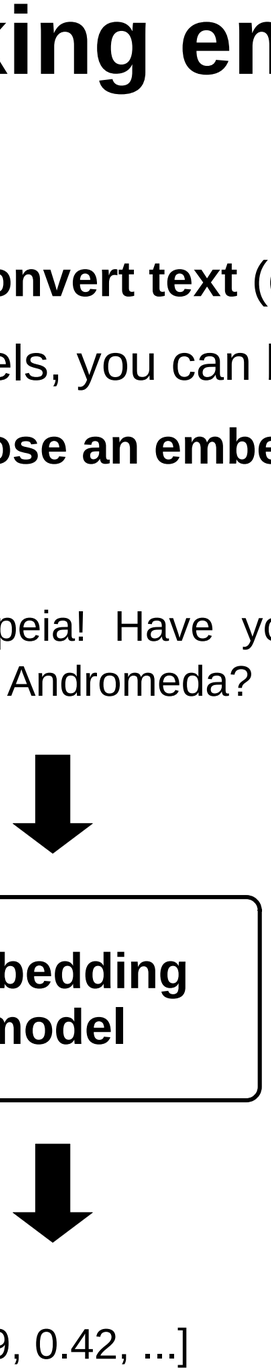
Benchmarking embedding models

Introduction

Benchmarking embedding models

- Embedding models **convert text** (or other modalities) **to a dense vector**.
- With embedding models, you can **build RAG and recommendation systems**.
- But **how do you choose an embedding model**?

Hi Cassiopeia! Have you talked with Andromeda?



[0.69, 0.42, ...]

Benchmarking embedding models

- How do you choose an embedding model?
- You look at **benchmarks**!
- MTEB ranks models based on their performance on **different benchmarks**.

Rank (Bor…	Model	Zero-shot	Memory U…	Number of P…	Embedding D…	Max Tokens	Mean (T…	Mean (TaskT…	Bitext …	Classification	Clustering
1	gemini-embedding-001	99%	Unknown	Unknown	3072	2048	68.37	59.59	79.28	71.82	54.59
2	Qwen3-Embedding-8B	99%	28866	7B	4096	32768	**70.58**	**61.69**	**80.89**	**74.00**	**57.65**
3	Qwen3-Embedding-4B	99%	15341	4B	2560	32768	69.45	60.86	79.36	72.33	57.15
4	Qwen3-Embedding-0.6B	99%	2272	595M	1024	32768	64.34	56.01	72.23	66.83	52.33
5	gte-Qwen2-7B-instruct	⚠ NA	29040	7B	3584	32768	62.51	55.93	73.92	61.55	52.77
6	Linq-Embed-Mistral	99%	13563	7B	4096	32768	61.47	54.14	70.34	62.24	50.60
7	multilingual-e5-large-instruct	99%	1068	560M	1024	514	63.22	55.08	80.13	64.94	50.75
8	embeddinggemma-300m	99%	578	307M	768	2048	61.15	54.31	64.40	60.90	51.17
9	SFR-Embedding-Mistral	96%	13563	7B	4096	32768	60.90	53.92	70.00	60.02	51.84
10	GritLM-7B	99%	13813	7B	4096	32768	60.92	53.74	70.53	61.83	49.75
11	text-multilingual-embedding-002	99%	Unknown	Unknown	768	2048	62.16	54.25	70.73	64.64	47.84

The MTEB leaderboard on 20/09/2025.

Benchmarking embedding models

- Should we **trust** these public benchmarks?
- You can also create **your own benchmark** with your private data.
- The dataset must be **clean, diverse, and in your language** (or multilingual).

Rank (Bor…	Model	Zero-shot	Memory U…	Number of P…	Embedding D…	Max Tokens	Mean (T…	Mean (TaskT…	Bitext …	Classification	Clustering
1	gemini-embedding-001	99%	Unknown	Unknown	3072	2048	68.37	59.59	79.28	71.82	54.59
2	Qwen3-Embedding-8B	99%	28866	7B	4096	32768	**70.58**	**61.69**	**80.89**	**74.00**	**57.65**
3	Qwen3-Embedding-4B	99%	15341	4B	2560	32768	69.45	60.86	79.36	72.33	57.15
4	Qwen3-Embedding-0.6B	99%	2272	595M	1024	32768	64.34	56.01	72.23	66.83	52.33
5	gte-Qwen2-7B-instruct	⚠ NA	29040	7B	3584	32768	62.51	55.93	73.92	61.55	52.77
6	Linq-Embed-Mistral	99%	13563	7B	4096	32768	61.47	54.14	70.34	62.24	50.60
7	multilingual-e5-large-instruct	99%	1068	560M	1024	514	63.22	55.08	80.13	64.94	50.75
8	embeddinggemma-300m	99%	578	307M	768	2048	61.15	54.31	64.40	60.90	51.17
9	SFR-Embedding-Mistral	96%	13563	7B	4096	32768	60.90	53.92	70.00	60.02	51.84
10	GritLM-7B	99%	13813	7B	4096	32768	60.92	53.74	70.53	61.83	49.75
11	text-multilingual-embedding-002	99%	Unknown	Unknown	768	2048	62.16	54.25	70.73	64.64	47.84

The MTEB leaderboard on 20/09/2025.

Benchmarking embedding models

In this course, you will:

- Create a **golden dataset**.
- Run **open source** and **proprietary models**.
- Compute **metrics** to grade the models.
- Perform **statistical tests** to prove if a model is better than another.
- **Automate** some steps in the pipeline.
- **Generate tables** to compare the models.

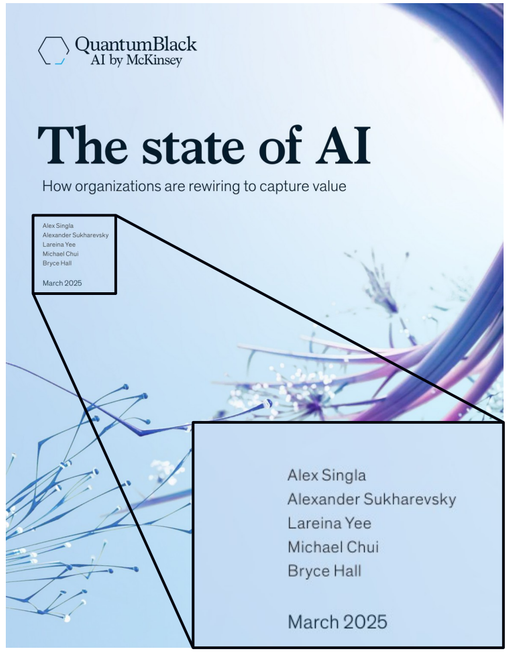
#	Model	mrr	recall@1	recall@5	ndcg@5
a	OpenAI 3-Small	0.780	0.683	0.901	0.804
b	OpenAI 3-Large	0.778	0.681	0.901	0.802
c	Google Gemini-001	0.810 ab	0.725 b	0.927 ab	0.834 ab

Comparing Gemini embedding to OpenAI's embedding models.

Extract text from PDF files

Extract text from PDF files

- Extracting text from PDF files is challenging.
- PDF files can be scanned, have complex layouts, and contain images, tables, etc
- In Python, we can use libraries like PyMuPDF, PyPDF2, and pdfplumber.
- These libraries are ineffective if you want to preserve the structure of the document.



PyMuPDF

The state of AI

March 2025

Alex Singla

Alexander Sukharevsky

Lareina Yee

Michael Chui

Bryce Hall

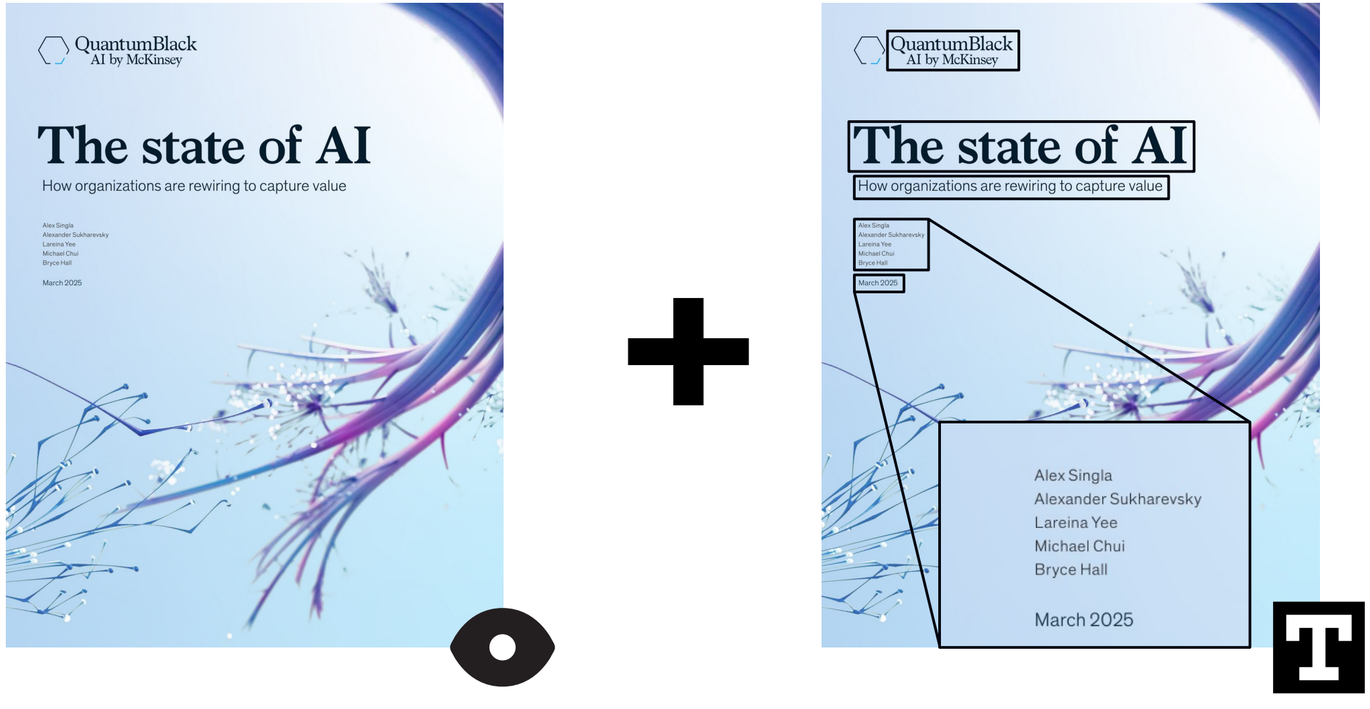
How organizations are rewiring to capture value

Extract text from PDF files

- We can use vision language (VL) models to parse PDF documents and images.
- VL models can see images

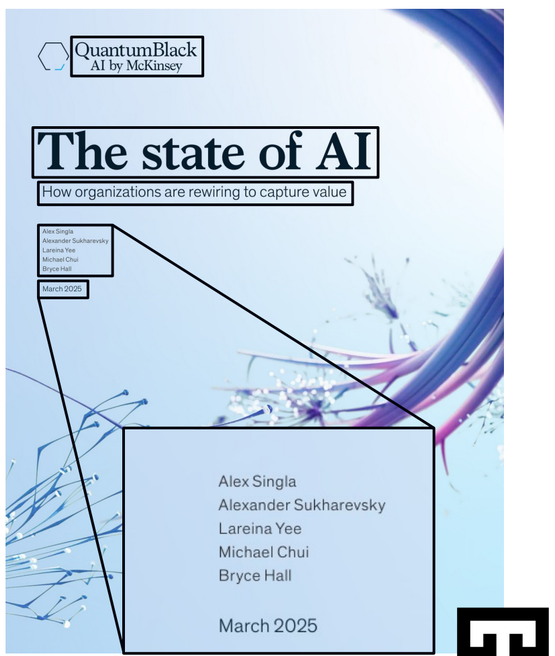
Extract text from PDF files

- We can use vision language (VL) models to parse PDF documents and images.
- VL models can see images, read and understand text.



Extract text from PDF files

- We can use vision language (VL) models to parse PDF documents and images.
- VL models can see images, read and understand text.
- The vision and language parts work together to parse the image effectively.



QuantumBlack

AI by McKinsey

The state of AI

How organizations are rewiring to capture value

Alex Singla

Alexander Sukharevsky

Lareina Yee

Michael Chui

Bryce Hall

March 2025

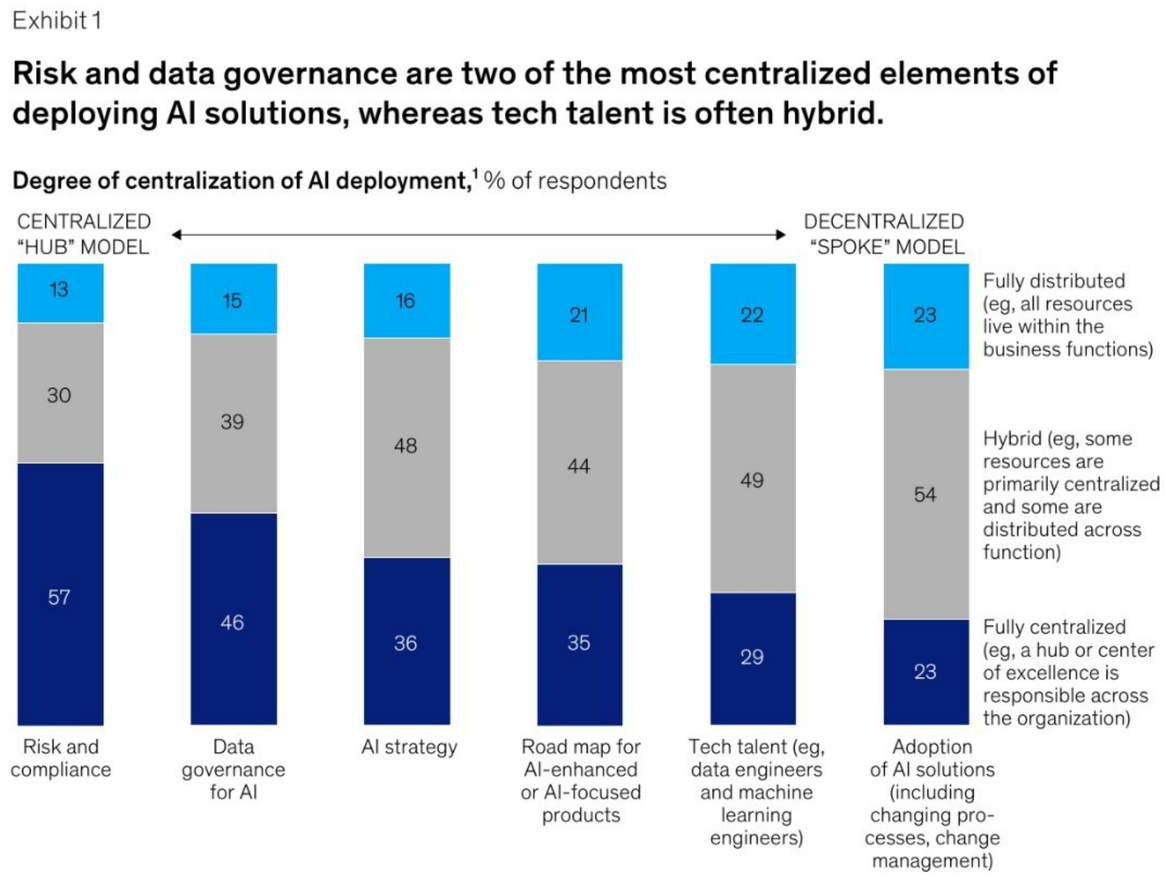
Extract text from PDF files

- What is the trade-off?
- Let’s compare both methods side-by-side.

	Python libraries	**VL models**
Cost	**Free**	**Free** / Paid
Scanned input	No	**Yes**
Resources	**Low**	High / **Low**
Preserve structure	No	**Yes**
Handle complex layouts	No	**Yes**
Understand the content	No	**Yes**
Speed	**Fast**	Slow

Extract text from PDF files

Side-by-side comparison: Test N°1



Extract text from PDF files

Side-by-side comparison: Test N°1

PyMuPDF

Exhibit 1

Degree of centralization of AI deployment,[1] % of respondents

McKinsey & Company

[1]Question was asked only of respondents whose organizations use AI in at least 1 function, n = 1,229. Figures were calculated after removing the share who said

“don’t know/not applicable.”

Source: McKinsey Global Survey on the state of AI, 1,491 participants at all levels of the organization, July 16–31, 2024

Risk and data governance are two of the most centralized elements of

deploying AI solutions, whereas tech talent is often hybrid.

57

46

36

35

29

23

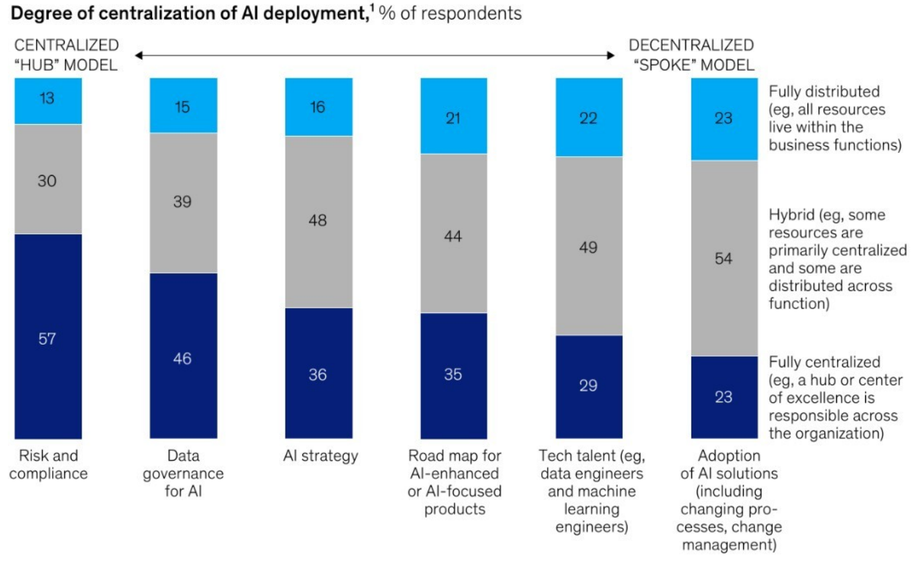
30

39

Exhibit 1

Risk and data governance are two of the most centralized elements of deploying AI solutions, whereas tech talent is often hybrid.

Degree of centralization of AI deployment,[1] % of respondents



[1]Question was asked only of respondents whose organizations use AI in at least 1 function, n = 1,229. Figures were calculated after removing the share who said “don’t know/not applicable.”
Source: McKinsey Global Survey on the state of AI, 1,491 participants at all levels of the organization, July 16–31, 2024

Extract text from PDF files

Side-by-side comparison: Test N°1

PyMuPDF

Exhibit 1

Degree of centralization of AI deployment,[1] % of respondents

McKinsey & Company

[1]Question was asked only of respondents whose organizations use AI in at least 1 function, n = 1,229. Figures were calculated after removing the share who said

“don’t know/not applicable.”

Source: McKinsey Global Survey on the state of AI, 1,491 participants at all levels of the organization, July 16–31, 2024

Risk and data governance are two of the most centralized elements of

deploying AI solutions, whereas tech talent is often hybrid.

57

46

36

35

29

23

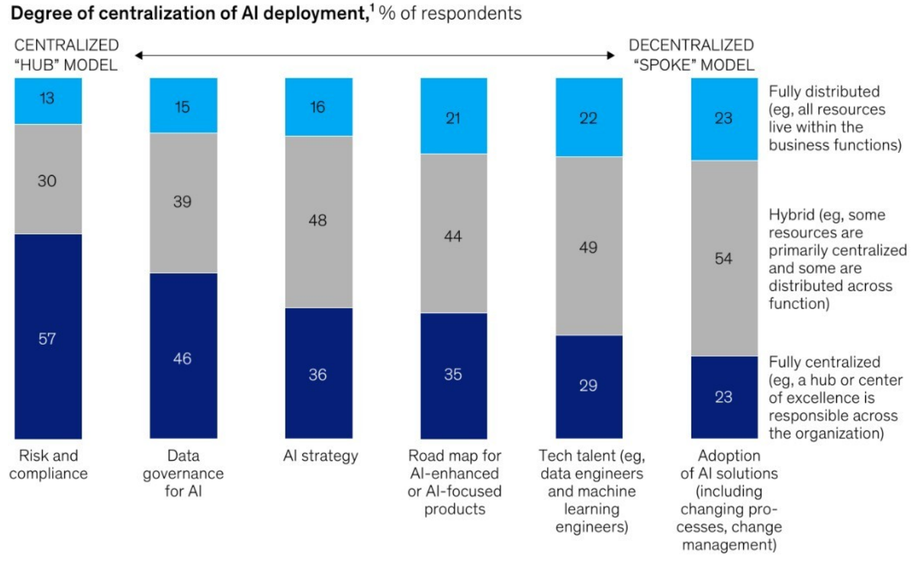
30

39

Exhibit 1

Risk and data governance are two of the most centralized elements of deploying AI solutions, whereas tech talent is often hybrid.

Degree of centralization of AI deployment,[1] % of respondents



[1]Question was asked only of respondents whose organizations use AI in at least 1 function, n = 1,229. Figures were calculated after removing the share who said “don't know/not applicable.”
Source: McKinsey Global Survey on the state of AI, 1,491 participants at all levels of the organization, July 16–31, 2024

Extract text from PDF files

Side-by-side comparison: Test N°1

PyMuPDF

Exhibit 1

Degree of centralization of AI deployment,[1] % of respondents

McKinsey & Company

[1]Question was asked only of respondents whose organizations use AI in at least 1 function, n = 1,229. Figures were calculated after removing the share who said

“don’t know/not applicable.”

Source: McKinsey Global Survey on the state of AI, 1,491 participants at all levels of the organization, July 16–31, 2024

Risk and data governance are two of the most centralized elements of

deploying AI solutions, whereas tech talent is often hybrid.

57

46

36

35

29

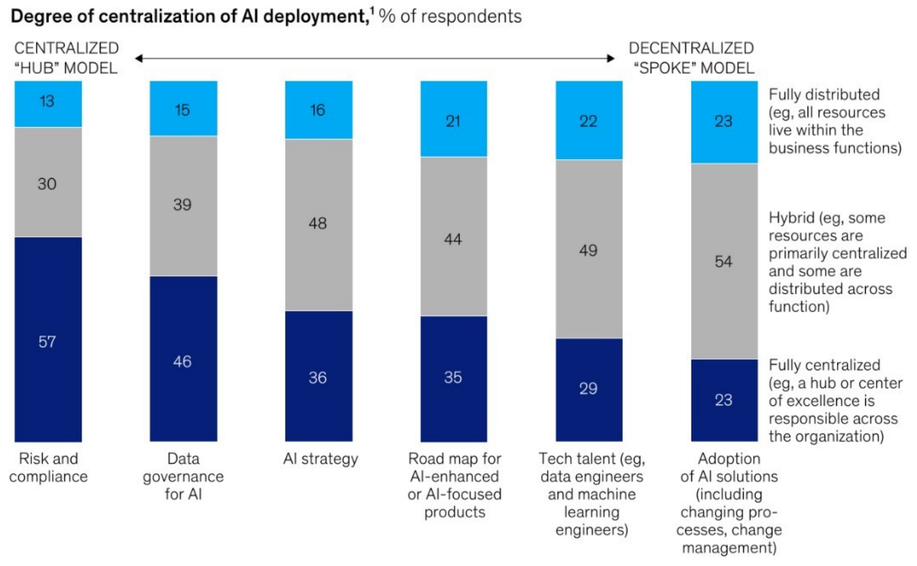
23

30

39

Exhibit 1

Risk and data governance are two of the most centralized elements of deploying AI solutions, whereas tech talent is often hybrid.



[1]Question was asked only of respondents whose organizations use AI in at least 1 function, n = 1,229. Figures were calculated after removing the share who said “don't know/not applicable.”
Source: McKinsey Global Survey on the state of AI, 1,491 participants at all levels of the organization, July 16–31, 2024

Extract text from PDF files

Side-by-side comparison: Test N°1

PyMuPDF

Exhibit 1

Degree of centralization of AI deployment,[1] % of respondents

McKinsey & Company

[1]Question was asked only of respondents whose organizations use AI in at least 1 function, n = 1,229. Figures were calculated after removing the share who said

“don’t know/not applicable.”

Source: McKinsey Global Survey on the state of AI, 1,491 participants at all levels of the organization, July 16–31, 2024

Risk and data governance are two of the most centralized elements of deploying AI solutions, whereas tech talent is often hybrid.

57

46

36

35

29

23

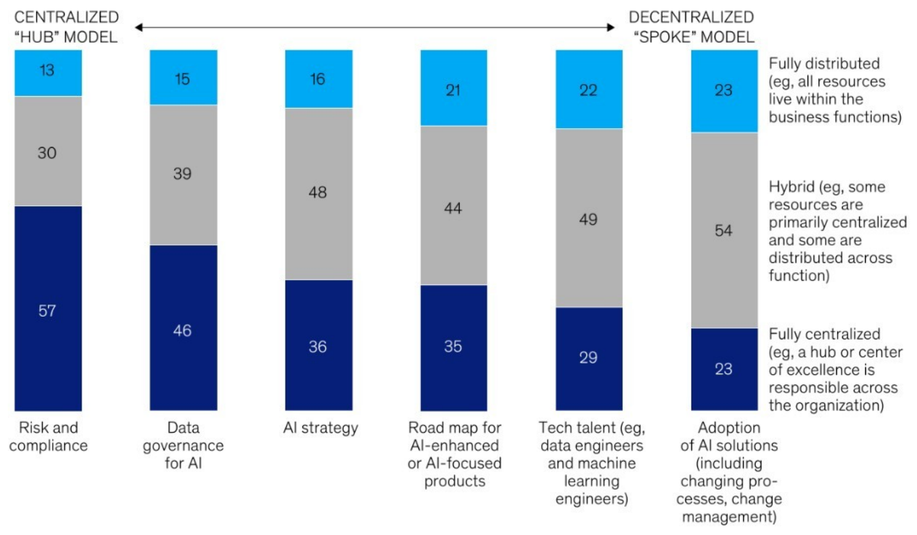
30

39

Exhibit 1

Risk and data governance are two of the most centralized elements of deploying AI solutions, whereas tech talent is often hybrid.

Degree of centralization of AI deployment,[1] % of respondents



[1]Question was asked only of respondents whose organizations use AI in at least 1 function, n = 1,229. Figures were calculated after removing the share who said “don't know/not applicable.”
Source: McKinsey Global Survey on the state of AI, 1,491 participants at all levels of the organization, July 16–31, 2024

Extract text from PDF files

Side-by-side comparison: Test N°1

PyMuPDF (Continuation)

49

54

13

15

16

21

22

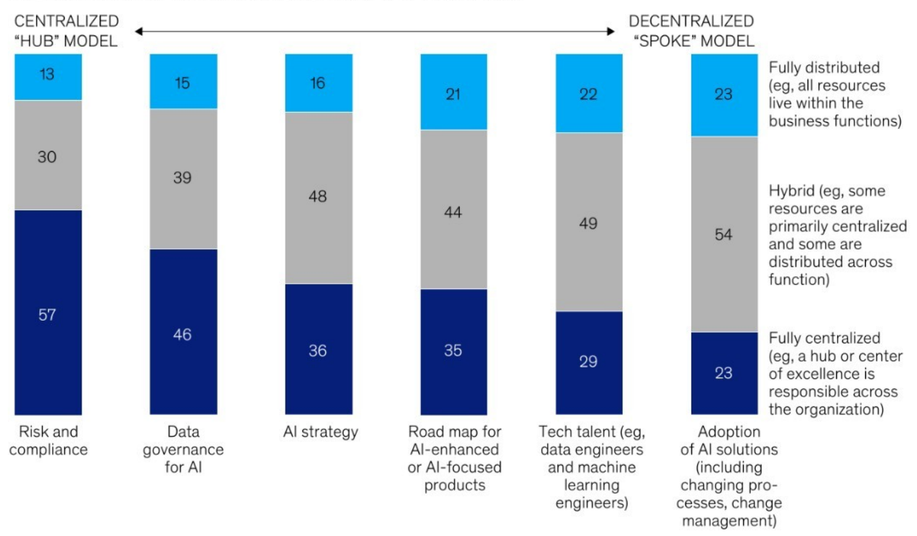
23

Fully centralized (eg, a hub or center of excellence is responsible across the organization) Hybrid (eg, some resources are primarily centralized and some are ...

Exhibit 1

Risk and data governance are two of the most centralized elements of deploying AI solutions, whereas tech talent is often hybrid.

Degree of centralization of AI deployment,[1] % of respondents



[1]Question was asked only of respondents whose organizations use AI in at least 1 function, n = 1,229. Figures were calculated after removing the share who said "don't know/not applicable."
Source: McKinsey Global Survey on the state of AI, 1,491 participants at all levels of the organization, July 16–31, 2024

Extract text from PDF files

Side-by-side comparison: Test N°1

VL model (Gemini 2.5 Pro)

Exhibit 1

Risk and data governance are two of the most centralized elements of deploying AI solutions, whereas tech talent is often hybrid.

The chart shows two models of centralization: a "**CENTRALIZED 'HUB' MODEL**" on the left and a "**DECENTRALIZED 'SPOKE' MODEL**" on the right. The chart has six vertical stacked bars, each representing a different aspect of AI deployment. The legend on the right explains the color coding for the segments of each bar: a dark blue segment represents "**Fully centralized** (eg, a hub or center of excellence is responsible across the organization)", a gray segment represents "**Hybrid** (eg, some resources are primarily centralized and some are distributed across function)", and a light blue segment represents "**Fully distributed** (eg, all resources live within the business functions)".

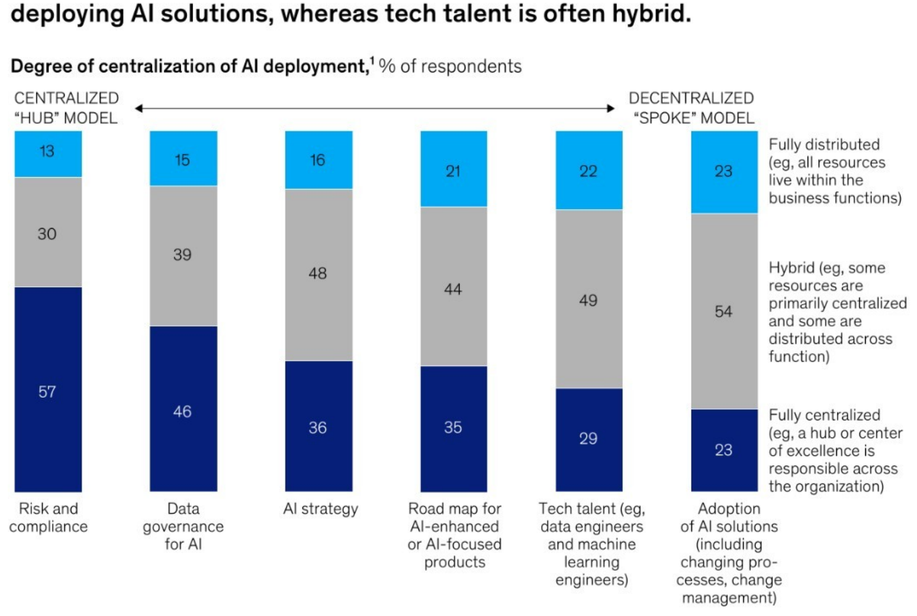
The six bars are for the following categories, with their respective data breakdowns:

1. **Risk and compliance**: **57%** Fully centralized, **30%** Hybrid, **13%** Fully distributed.
2. **Data governance for AI**: **46%** Fully centralized, **39%** Hybrid, **15%** Fully distributed.
3. **AI strategy**: **36%** Fully centralized, **48%** Hybrid, **16%** Fully distributed.
4. **Road map for AI-enhanced or AI-focused products**: **35%** Fully centralized, **44%** Hybrid, **21%** Fully distributed.
5. **Tech talent** (eg, data engineers and machine learning engineers): **29%** Fully centralized, **49%** Hybrid, **22%** Fully distributed.
6. **Adoption of AI solutions** (including changing processes, change management): **23%** Fully centralized, **54%** Hybrid, **23%** Fully distributed.] ...

Exhibit 1

Risk and data governance are two of the most centralized elements of deploying AI solutions, whereas tech talent is often hybrid.

Degree of centralization of AI deployment,[1] % of respondents



[1]Question was asked only of respondents whose organizations use AI in at least 1 function, n = 1,229. Figures were calculated after removing the share who said "don't know/not applicable."
Source: McKinsey Global Survey on the state of AI, 1,491 participants at all levels of the organization, July 16–31, 2024

Extract text from PDF files

Side-by-side comparison: Test N°1

VL model (Gemini 2.5 Pro)

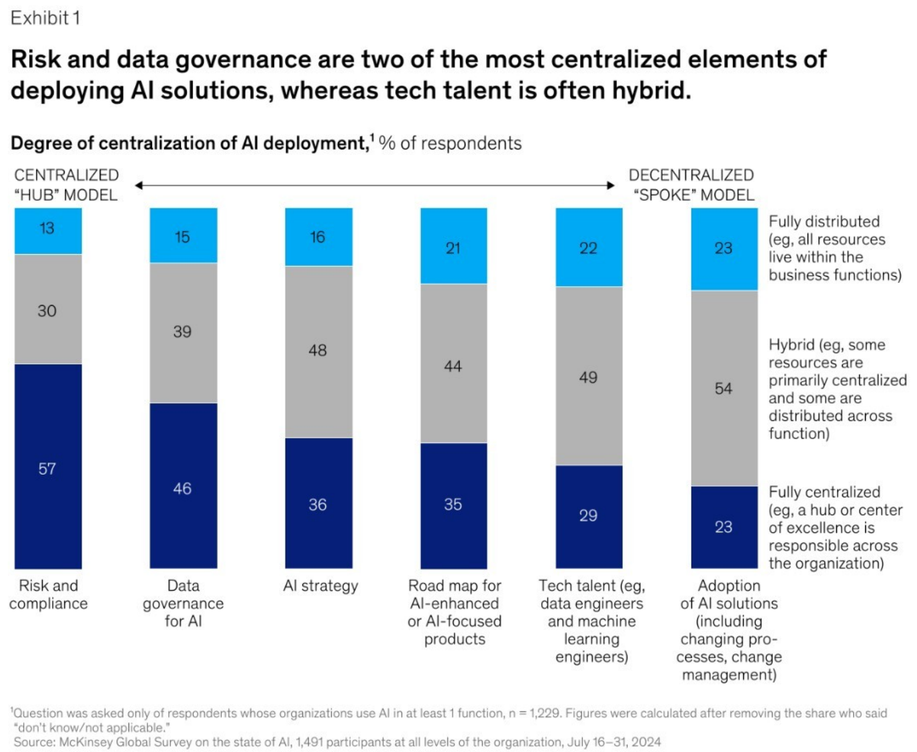
Exhibit 1

Risk and data governance are two of the most centralized elements of deploying AI solutions, whereas tech talent is often hybrid.

The chart shows two models of centralization: a "**CENTRALIZED 'HUB' MODEL**" on the left and a "**DECENTRALIZED 'SPOKE' MODEL**" on the right. The chart has six vertical stacked bars, each representing a different aspect of AI deployment. The legend on the right explains the color coding for the segments of each bar: a dark blue segment represents "**Fully centralized** (eg, a hub or center of excellence is responsible across the organization)", a gray segment represents "**Hybrid** (eg, some resources are primarily centralized and some are distributed across function)", and a light blue segment represents "**Fully distributed** (eg, all resources live within the business functions)".

The six bars are for the following categories, with their respective data breakdowns:

1. **Risk and compliance**: **57%** Fully centralized, **30%** Hybrid, **13%** Fully distributed.
2. **Data governance for AI**: **46%** Fully centralized, **39%** Hybrid, **15%** Fully distributed.
3. **AI strategy**: **36%** Fully centralized, **48%** Hybrid, **16%** Fully distributed.
4. **Road map for AI-enhanced or AI-focused products**: **35%** Fully centralized, **44%** Hybrid, **21%** Fully distributed.
5. **Tech talent** (eg, data engineers and machine learning engineers): **29%** Fully centralized, **49%** Hybrid, **22%** Fully distributed.
6. **Adoption of AI solutions** (including changing processes, change management): **23%** Fully centralized, **54%** Hybrid, **23%** Fully distributed.] ...



Extract text from PDF files

Side-by-side comparison: Test N°1

VL model (Gemini 2.5 Pro)

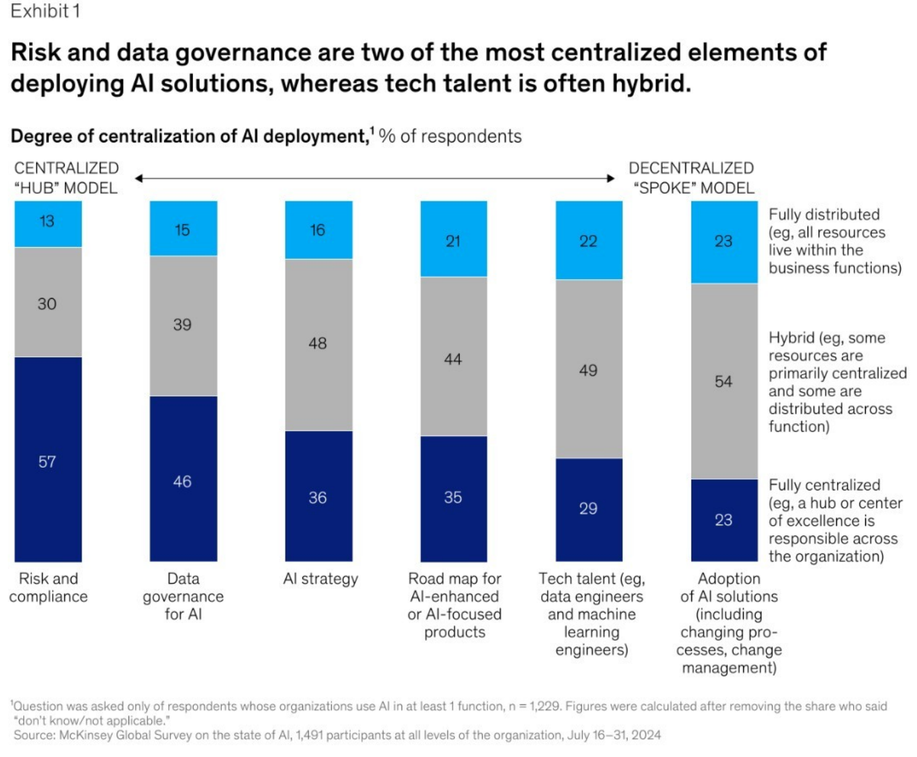
Exhibit 1

Risk and data governance are two of the most centralized elements of deploying AI solutions, whereas tech talent is often hybrid.

The chart shows two models of centralization: a "**CENTRALIZED 'HUB' MODEL**" on the left and a "**DECENTRALIZED 'SPOKE' MODEL**" on the right. The chart has six vertical stacked bars, each representing a different aspect of AI deployment. The legend on the right explains the color coding for the segments of each bar: a dark blue segment represents "**Fully centralized** (eg, a hub or center of excellence is responsible across the organization)", a gray segment represents "**Hybrid** (eg, some resources are primarily centralized and some are distributed across function)", and a light blue segment represents "**Fully distributed** (eg, all resources live within the business functions)".

The six bars are for the following categories, with their respective data breakdowns:

1. **Risk and compliance**: **57%** Fully centralized, **30%** Hybrid, **13%** Fully distributed.
2. **Data governance for AI**: **46%** Fully centralized, **39%** Hybrid, **15%** Fully distributed.
3. **AI strategy**: **36%** Fully centralized, **48%** Hybrid, **16%** Fully distributed.
4. **Road map for AI-enhanced or AI-focused products**: **35%** Fully centralized, **44%** Hybrid, **21%** Fully distributed.
5. **Tech talent** (eg, data engineers and machine learning engineers): **29%** Fully centralized, **49%** Hybrid, **22%** Fully distributed.
6. **Adoption of AI solutions** (including changing processes, change management): **23%** Fully centralized, **54%** Hybrid, **23%** Fully distributed.] ...



Extract text from PDF files

Side-by-side comparison: Test N°2

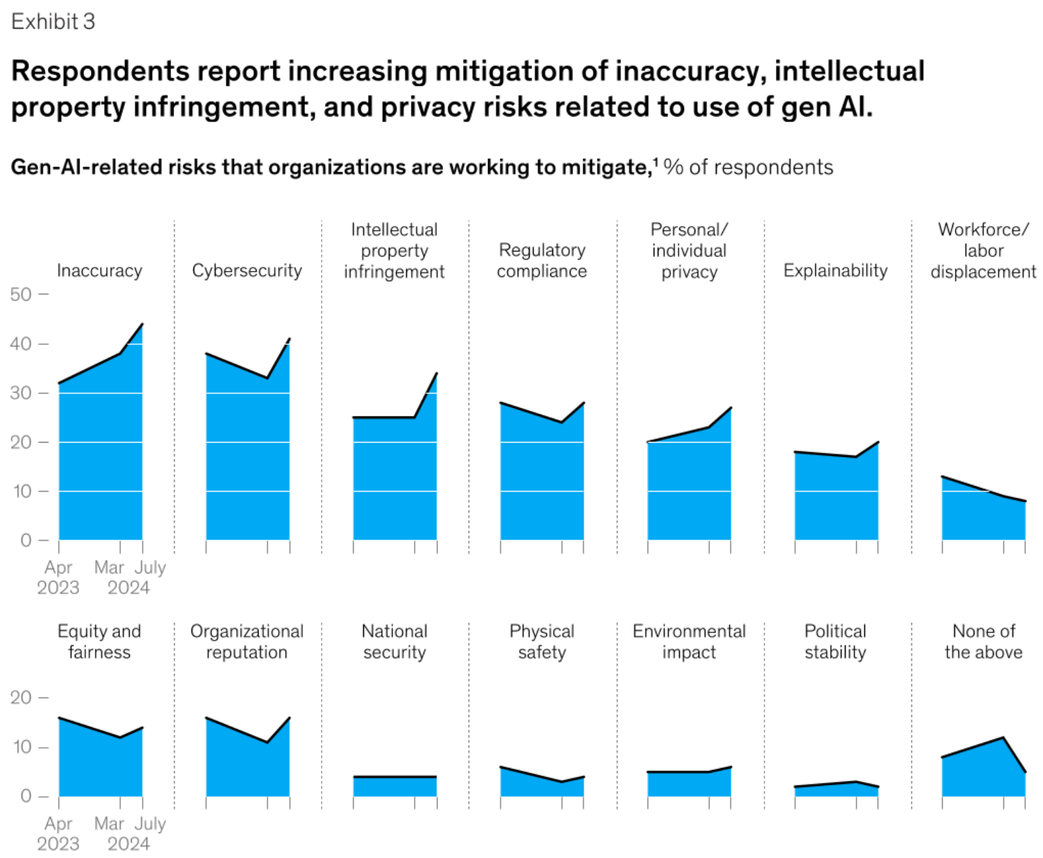


Exhibit 3

Respondents report increasing mitigation of inaccuracy, intellectual property infringement, and privacy risks related to use of gen AI.

Gen-AI-related risks that organizations are working to mitigate,[1] % of respondents

[1]Only asked of respondents whose organizations use AI in at least 1 business function. Respondents who said "don't know/not applicable" are not shown.
Source: McKinsey Global Surveys on the state of AI, 2023–24

Extract text from PDF files

Side-by-side comparison: Test N°2

PyMuPDF

Exhibit 3

Gen-AI-related risks that organizations are working to mitigate,1 % of respondents

1Only asked of respondents whose organizations use AI in at least 1 business function. Respondents who said “don’t know/not applicable” are not shown.

Source: McKinsey Global Surveys on the state of AI, 2023–24

Respondents report increasing mitigation of inaccuracy, intellectual property infringement, and privacy risks related to use of gen AI.

McKinsey & Company

0

10

20

30

40

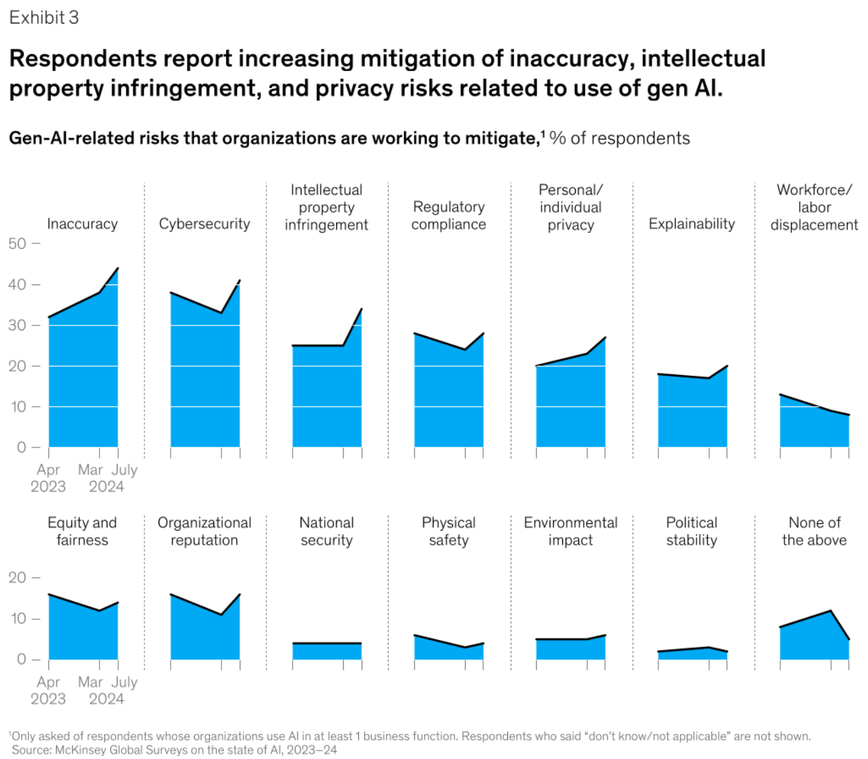
50

0

10

20

...



Extract text from PDF files

Side-by-side comparison: Test N°2

VL model (Gemini 2.5 Pro)

Exhibit 3

Respondents report increasing mitigation of inaccuracy, intellectual property infringement, and privacy risks related to use of gen AI.

Gen-AI-related risks that organizations are working to mitigate,[1] % of respondents

Image of a series of **14 area charts** arranged in **two rows of seven**. The charts show the percentage of respondents working to mitigate various generative AI-related risks over time, from **April 2023 to July 2024**.

The charts in the **top row** have a **y-axis ranging from 0 to 50**. The charts are:

* Inaccuracy: Shows a **significant increasing trend**, starting around **32%** in April 2023 and rising to approximately **45%** by July 2024.

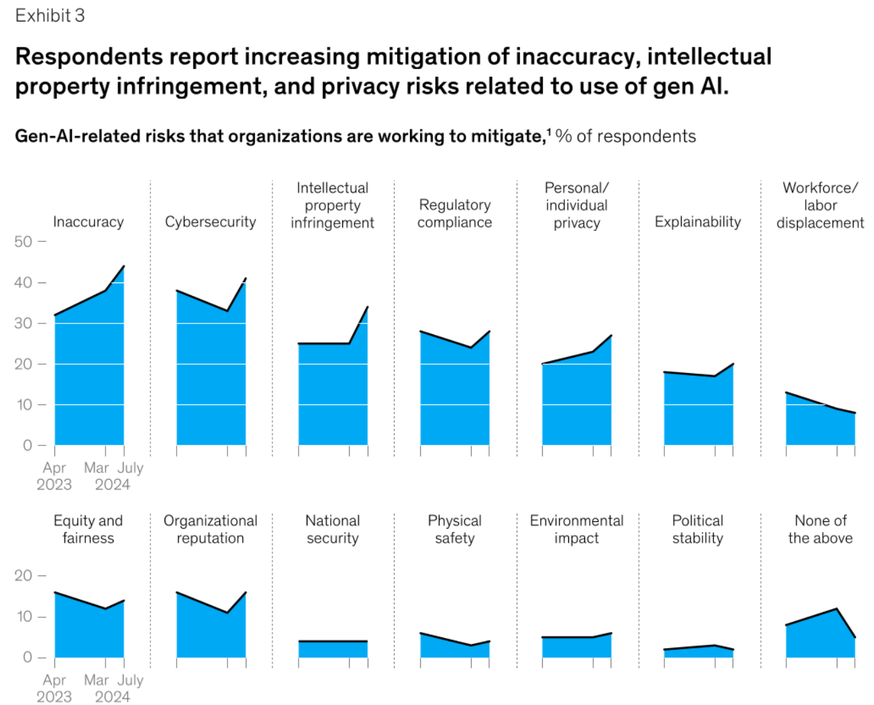
* Cybersecurity: Shows a **slight overall decrease**, starting near **38%** in April 2023 and ending around **35%** in July 2024, with a dip in between.

* Intellectual property infringement: Shows a **clear increasing trend**, starting at about **25%** in April 2023 and rising to roughly **35%** in July 2024.

* Regulatory compliance: Shows a **slight downward trend**, starting at about **28%** in April 2023 and ending around **25%** in July 2024.

* Personal/individual privacy: Shows an **increasing trend**, starting from approximately **22%** in April 2023 and rising to about **28%** by July 2024.

* Explainability: Shows a **slight increase**, starting from about **18%** in April 2023 and ending around **20%** in July 2024 ...



Extract text from PDF files

Side-by-side comparison: Test N°2

VL model (Gemini 2.5 Pro)

Exhibit 3

Respondents report increasing mitigation of inaccuracy, intellectual property infringement, and privacy risks related to use of gen AI.

Gen-AI-related risks that organizations are working to mitigate,[1] % of respondents

Image of a series of **14 area charts** arranged in **two rows of seven**. The charts show the percentage of respondents working to mitigate various generative AI-related risks over time, from **April 2023 to July 2024**.

The charts in the **top row** have a **y-axis ranging from 0 to 50**. The charts are:

* Inaccuracy: Shows a **significant increasing trend**, starting around **32%** in April 2023 and rising to approximately **45%** by July 2024.

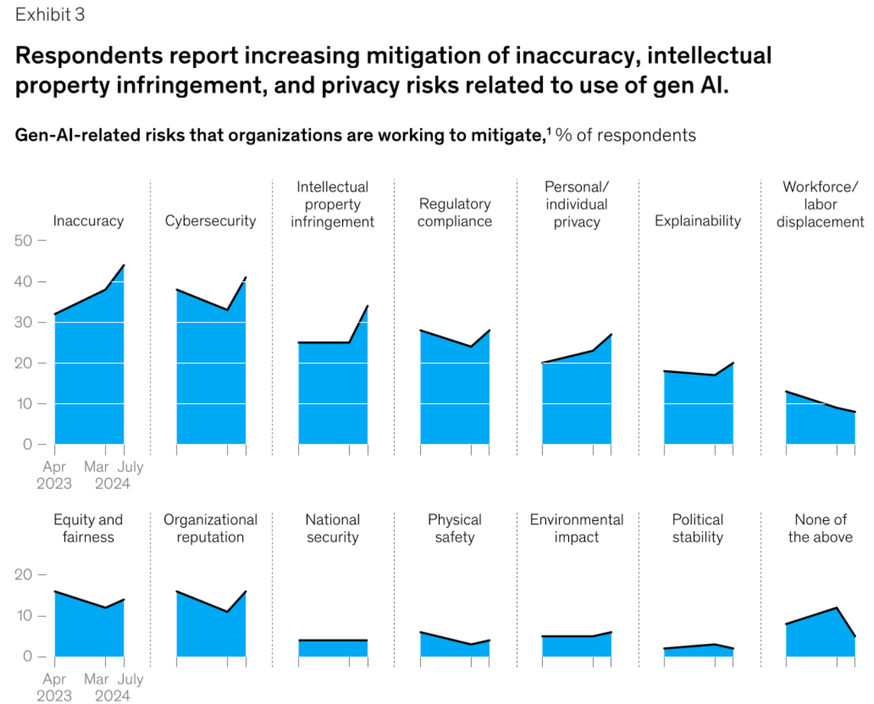
* Cybersecurity: Shows a **slight overall decrease**, starting near **38%** in April 2023 and ending around **35%** in July 2024, with a dip in between.

* Intellectual property infringement: Shows a **clear increasing trend**, starting at about **25%** in April 2023 and rising to roughly **35%** in July 2024.

* Regulatory compliance: Shows a **slight downward trend**, starting at about **28%** in April 2023 and ending around **25%** in July 2024.

* Personal/individual privacy: Shows an **increasing trend**, starting from approximately **22%** in April 2023 and rising to about **28%** by July 2024.

* Explainability: Shows a **slight increase**, starting from about **18%** in April 2023 and ending around **20%** in July 2024 ...



Extract text from PDF files

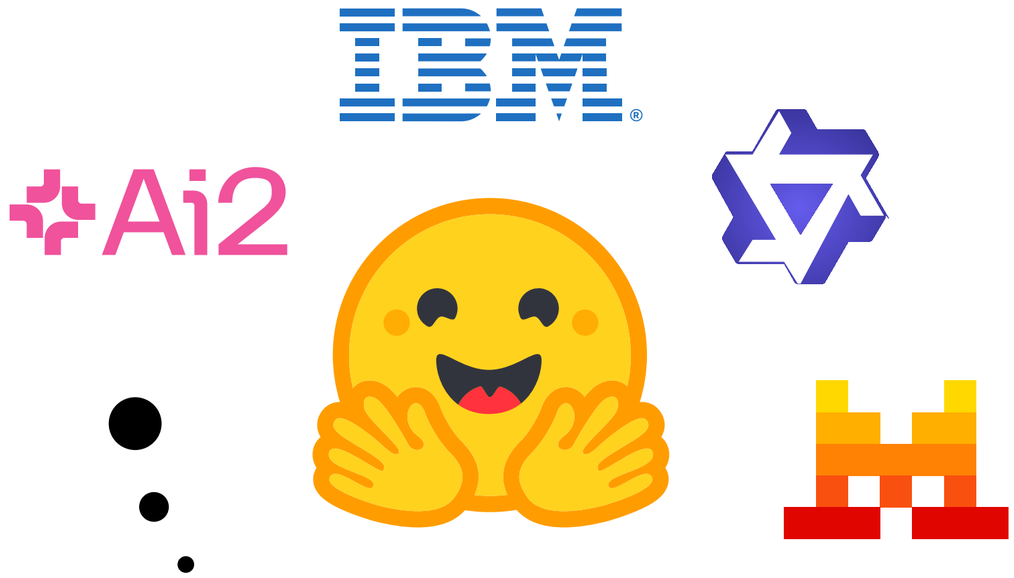
- VL models will help you create high quality datasets by:
  - **Annotating** images.
  - Describing tables or outputting them in **Markdown**.
  - **Preserve the layout**.
  - **Extract only what you need** from the documents.
  - Working with **scanned** documents.

Extract text from PDF files

- What model should you use?
- Use proprietary and open models through the API. **Cheaper!**
- Host the models in your own servers. **Expensive!!!**

Proprietary models

Open models



Extract text from PDF files

- Keep track of the best models with these leaderboards:
  - OpenVLM leaderboard.
  - Intelligent document processing leaderboard.

Evaluation Dimension

Avg Score | Avg Rank | MMBench_V11 | MMStar | MME | MMMU_VAL | MathVista | OCRBench | AI2D | HallusionBench | SEEDBench_IMG | MMVet | LL | RealWorldQA | POPE | ScienceQA_TEST | SEEDBench2_Plus | MMT-Bench_VAL | BLINK

Model Name: Input the Model Name (fuzzy, case insensitive)

Model Size: <4B | 4B-10B | 10B-20B | 20B-40B | >40B | Unknown

Model Type: API | OpenSource

Rank	Method	Param (B)	Language Model	Vision Model	Eval Date	Avg Score	Avg Rank	MMBench_V11	MMStar	MMMU_VAL	MathVista	OCRBench	AI2D
1	SenseNova-V6-5-Pro				2025/09/03	82.2	6.12	87.3	76.1	77	82.8	885	90.2
2	CongRong-v2.0				2025/05/20	80.7	4.88	88.1	75.3	75.6	76.8	927	90
3	SenseNova-V6-Pro				2025/05/05	80.4	7.12	88	73.7	70.4	76.9	895	89.2
4	Gemini-2.5-Pro				2025/04/07	80.1	9.25	88.3	73.6	74.7	80.9	862	89.5
5	JT-VL-Chat-V3.0				2025/08/04	79.9	11.88	87.5	82.1	68.7	72.8	950	88.3
6	GPT-5-20250807				2025/08/14	79.9	18.25	86.6	75.7	81.8	81.9	807	89.5
7	InternVL3-78B	78.4	Qwen2.5-72B	InternViT-6B-v2.5	2025/04/14	79.1	8.75	87.7	73.4	72.2	79	908	89.8
8	BlueLM-2.6-3	3			2025/09/	78.4	19.62	86.4	80.1	62.4	82.3	881	86.1

Leaderboard

RANK	MODEL	COST	AVG	KIE	VQA	OCR	CLASSIFICATION	LONGDOCBENCH	TABLE
1	gemini-2.5-pro-preview-06-05 (reasoning: low)	-	82.32	78.92	86.29	78.54	99.31	68.57	82.28
2	gemini-2.5-pro-preview-03-25 (reasoning: low)	1.113	82.04	79.66	85.99	81.18	99.18	66.69	79.51
3	gemini-2.5-flash-preview-04-17	0.133	81.00	77.99	85.16	78.9	99.05	69.08	75.82
4	claude-3.7-sonnet (reasoning:low)	1.748	79.99	76.09	83.47	69.19	98.92	75.93	91.23
5	o4-mini-2025-04-16	2.595	78.56	75.43	87.07	72.82	99.14	66.13	70.76
6	gpt-4.1-2025-04-14	1.583	78.05	72.68	80.37	75.64	99.27	66	74.34
7	gemini-2.0-flash	0.022	77.62	77.22	82.03	80.05	99.1	56.01	71.32
8	gpt-5-2025-08-07 (reasoning: low)	-	76.18	72.19	87.72	73.76	99.40	67.79	56.25
9	gpt-4o-2024-08-06	1.979	75.40	71.83	79.08	74.56	95.74	66.9	64.3
10	claude-sonnet-4	0.959	75.15	71.91	82.51	64.09	98.88	40.06	93.44
11	InternVL3-38B-Instruct	-	72.77	70.31	74.82	66.31	98.84	68.30	58.03
12	gemini-2.5-flash-lite-preview-06-17	0.0555	71.73	77.20	76.28	77.12	98.88	42.36	58.55
13	llama-4-maverick(400B-A17B)	0.058	70.80	73.3	80.1	70.66	98.84	27.74	74.15
14	gpt-4o-mini-2024-07-18	2.990	69.95	70.03	72.86	72.43	98.41	55.48	50.47
15	gemma-3-27b-it	-	69.71	72.81	66.85	54.75	98.49	72.95	52.38
16	qwen2.5-vl-72b-instruct	0.242	68.48	76.11	80.1	69.61	99.01	37.47	48.58
17	gpt-4.1-nano-2025-04-14	0.071	64.56	66.25	74.08	67.09	87.34	27.89	50.83
18	mistral-small-3.1-24b-instruct	0.02	61.50	63.73	71.5	51.01	91.86	29.23	61.64
19	gpt-4o-2024-11-20	1.868	60.08	70.91	75.6	74.91	14.38	63.95	60.74
Pending	qwen2.5-vl-32b-instruct	Pending	Pending	79.63	81.36	Pending	98.71	75.62	77.46
Pending	mistral-medium-3	Pending	Pending	74.21	80.02	69.05	98.39	Pending	70.21

1. Cost represents the average cost in cents per requests for each model.
2. The score for each task in the leaderboard is the average across all the datasets for the corresponding task.
3. We compute edit distance accuracy for all tasks and datasets except classification where we compute exact match, and table extraction where we use GriTS. Please check our paper for more details.

Dividing text into chunks

Divide text into chunks

- How do we **divide** the extracted text?
- Dividing the text **programmatically** is not a good idea.

The chart shows two models of centralization: a "**CENTRALIZED 'HUB' MODEL**" on the left and a "**DECENTRALIZED 'SPOKE' MODEL**" on the right. The chart has six vertical stacked bars, each representing a different aspect of AI deployment. The legend on the right explains the color coding for the segments of each bar: a dark blue segment represents "**Fully centralized** (eg, a hub or center of excellence is responsible across the organization)", a gray segment represents "**Hybrid** (eg, some resources are primarily centralized and some are distributed across function)", and a light blue segment represents "**Fully distributed** (eg, all resources live within the business functions)".

The six bars are for the following categories, with their respective data breakdowns:

1. **Risk and compliance**: **57%** Fully centralized, **30%** Hybrid, **13%** Fully distributed.
2. **Data governance for AI**: **46%** Fully centralized, **39%** Hybrid, **15%** Fully distributed.
3. **AI strategy**: **36%** Fully centralized, **48%** Hybrid, **16%** Fully distributed.

Divide text into chunks

- How do we **divide** the extracted text?
- Dividing the text **programmatically** is not a good idea.
- Each chunk should contain text that **focus on a specific idea**.

The chart shows two models of centralization: a "**CENTRALIZED 'HUB' MODEL**" on the left and a "**DECENTRALIZED 'SPOKE' MODEL**" on the right. The chart has six vertical stacked bars, each representing a different aspect of AI deployment. The legend on the right explains the color coding for the segments of each bar: a dark blue segment represents "**Fully centralized** (eg, a hub or center of excellence is responsible across the organization)", a gray segment represents "**Hybrid** (eg, some resources are primarily centralized and some are distributed across function)", and a light blue segment represents "**Fully distributed** (eg, all resources live within the business functions)".

The six bars are for the following categories, with their respective data breakdowns:

1. **Risk and compliance**: **57%** Fully centralized, **30%** Hybrid, **13%** Fully distributed.
2. **Data governance for AI**: **46%** Fully centralized, **39%** Hybrid, **15%** Fully distributed.
3. **AI strategy**: **36%** Fully centralized, **48%** Hybrid, **16%** Fully distributed.

Divide text into chunks

- How do we **divide** the extracted text?
- Dividing the text **programmatically** is not a good idea.
- Each chunk should contain text that **focus on a specific idea**.
- How do we do that?
  - We can do it **manually** (**slow**).
  - We use **large language models** (**fast**), but needs supervision.

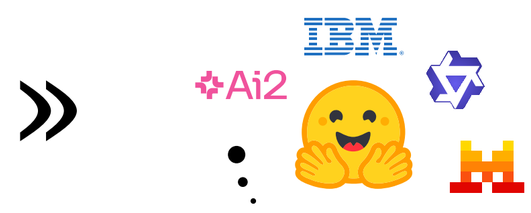
Antares is a bright sky, you can see it during summer in the constellation Scorpius

Text input

Divide text into chunks

- How do we **divide** the extracted text?
- Dividing the text **programmatically** is not a good idea.
- Each chunk should contain text that **focus on a specific idea**.
- How do we do that?
  - We can do it **manually** (**slow**).
  - We use **large language models** (**fast**), but needs supervision.

Antares is a bright sky, you can see it during summer in the constellation Scorpius

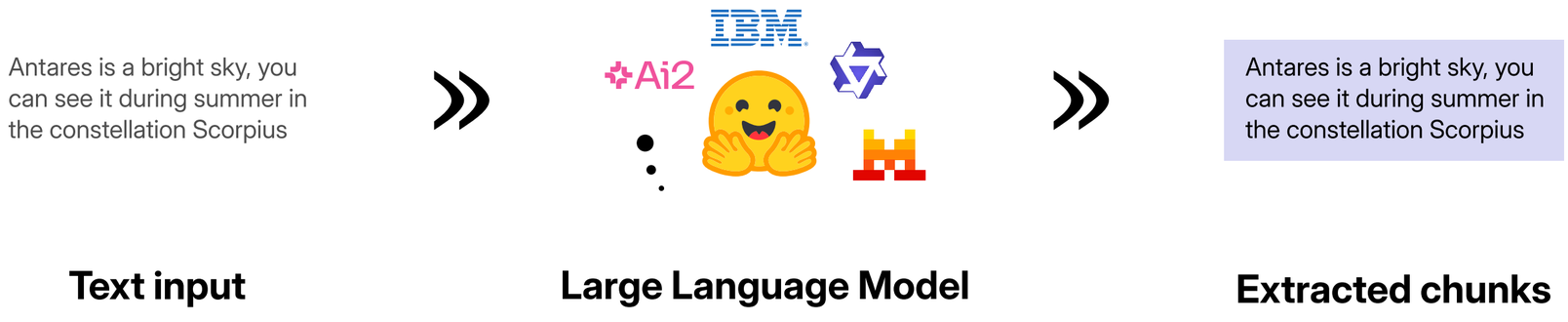


Text input

Large Language Model

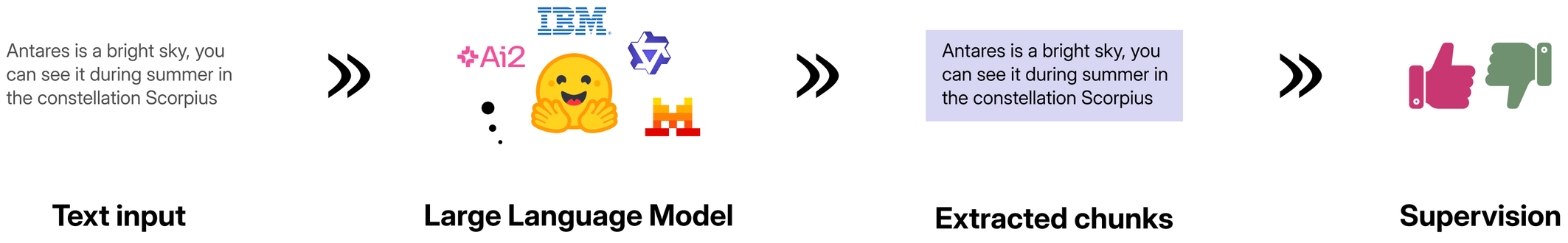
Divide text into chunks

- How do we **divide** the extracted text?
- Dividing the text **programmatically** is not a good idea.
- Each chunk should contain text that **focus on a specific idea**.
- How do we do that?
  - We can do it **manually** (**slow**).
  - We use **large language models** (**fast**), but needs supervision.



Divide text into chunks

- How do we **divide** the extracted text?
- Dividing the text **programmatically** is not a good idea.
- Each chunk should contain text that **focus on a specific idea**.
- How do we do that?
  - We can do it **manually** (**slow**).
  - We use **large language models** (**fast**), but needs supervision.



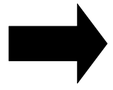
Generating question answer pairs

Generate question-answer pairs

- Generate questions that can be answered by a chunk of text.
- A chunk can answer one or more questions.
- A question can be answered by one or more chunks.

bitsandbytes enables accessible large language models via k-bit quantization for PyTorch. bitsandbytes provides **three main features** for dramatically reducing memory consumption for inference and training: **8-bit optimizers**, **LLM.int8()** or **8-bit quantization**, and **QLoRA** or **4-bit quantization**.

Text chunk



Q.1/ What is the primary purpose of the bitsandbytes library?

Q.2/ What are the three main features bitsandbytes provides for reducing memory consumption?

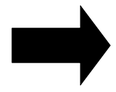
Questions

Text embeddings

Text embeddings

- A text embedding model **converts text into a numerical representation**.
- This numerical representation is a **dense vector containing n values**.

bitsandbytes enables accessible large language models via k-bit quantization for PyTorch. bitsandbytes provides **three main features** for dramatically reducing memory consumption for inference and training: **8-bit optimizers**, **LLM.int8()** or **8-bit quantization**, and **QLoRA** or **4-bit quantization**.



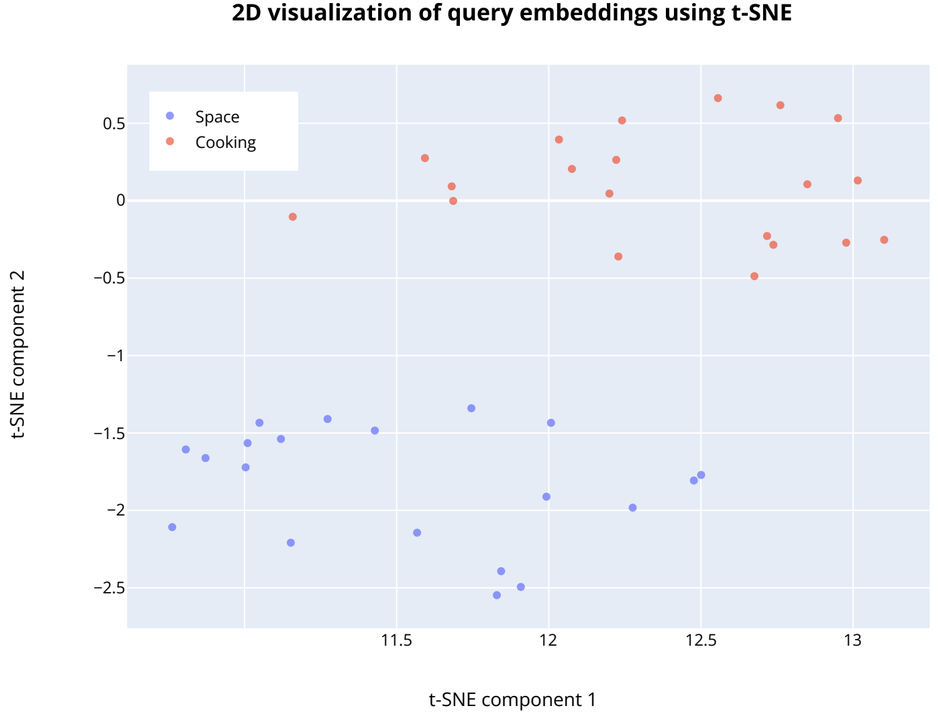
[0.58, 0.99, -0.1, 1, 0.03, …, 0.69]

Text chunk

Dense vector

Text embeddings

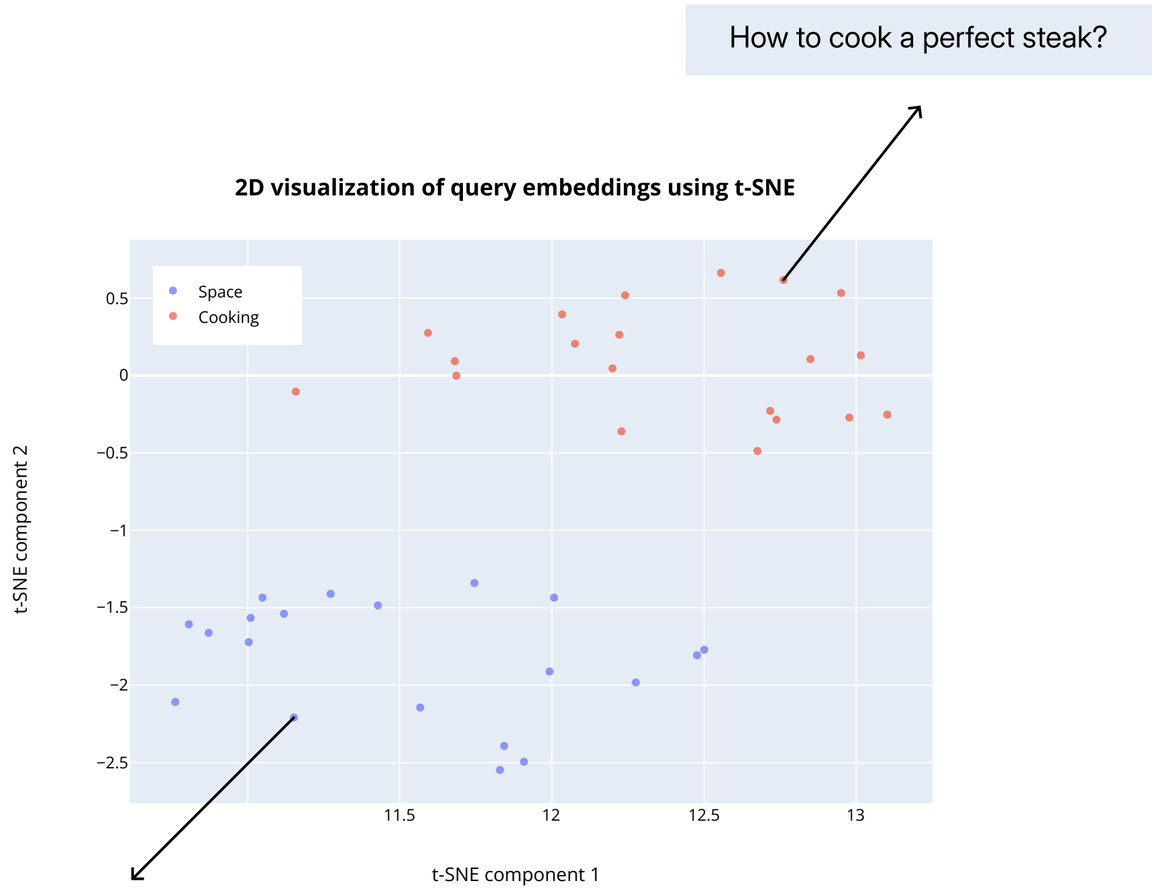
- A text embedding model **converts text into a numerical representation**.
- This numerical representation is a **dense vector containing n values**.
- What is the purpose?



Text embeddings

- A text embedding model **converts text into a numerical representation**.
- This numerical representation is a **dense vector containing n values**.
- What is the purpose?

How to cook a perfect steak?



How far is the Andromeda galaxy?

Text embeddings

- A text embedding model **converts text into a numerical representation**.
- That numerical representation is a **dense vector with n values**.
- What is the purpose?
- Does the size of the dense vector matter?

Dimension	Advantages	Disadvantages
Large	• Captures nuances in semantic information • Distinguishes finer details within the data	• Requires more computational resources for processing • Needs increased storage space for the vectors • Results in increased latency in vector databases
Small	• Requires few computational resources • Enables faster processing • Reduces storage costs • Achieves low latency	• May not capture as much semantic detail as larger embeddings

Text embeddings

- A text embedding model **converts text into a numerical representation**.
- That numerical representation is a **dense vector with n values**.
- What is the purpose?
- Does the size of the dense vector matter?

Model	Output dimension
Qwen3-Embedding-0.6B	1024
Qwen3-Embedding-4B	2560
Qwen3-Embedding-8B	**4096**
all-MiniLM-L6-v2	**384**
Gemini embedding 001	3072
text-embedding-3-small	1536
text-embedding-3-large	3072

Text embeddings

- A text embedding model **converts text into a numerical representation**.
- That numerical representation is a **dense vector with n values**.
- What is the purpose?
- Does the size of the dense vector matter?

Model	Output dimension	Size of 10k vectors (MB)
Qwen3-Embedding-0.6B	1024	39
Qwen3-Embedding-4B	2560	98
Qwen3-Embedding-8B	**4096**	**156**
all-MiniLM-L6-v2	**384**	**14**
Gemini embedding 001	3072	117
text-embedding-3-small	1536	59
text-embedding-3-large	3072	117

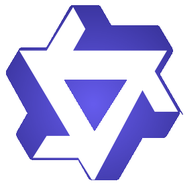
Text embeddings

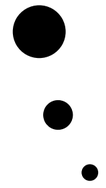
- You can use both open and closed-source models.
- There are many great open-source models that you can use without having a powerful device.

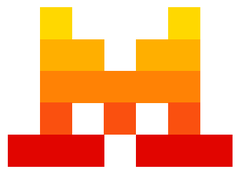
Proprietary models

ANTHROP\C

Open models







Text embeddings

- You can use both open and closed-source models.
- There are many great open-source models that you can use without having a powerful device.
- Use the MTEB leaderboard to get a quick look at the best embedding models.

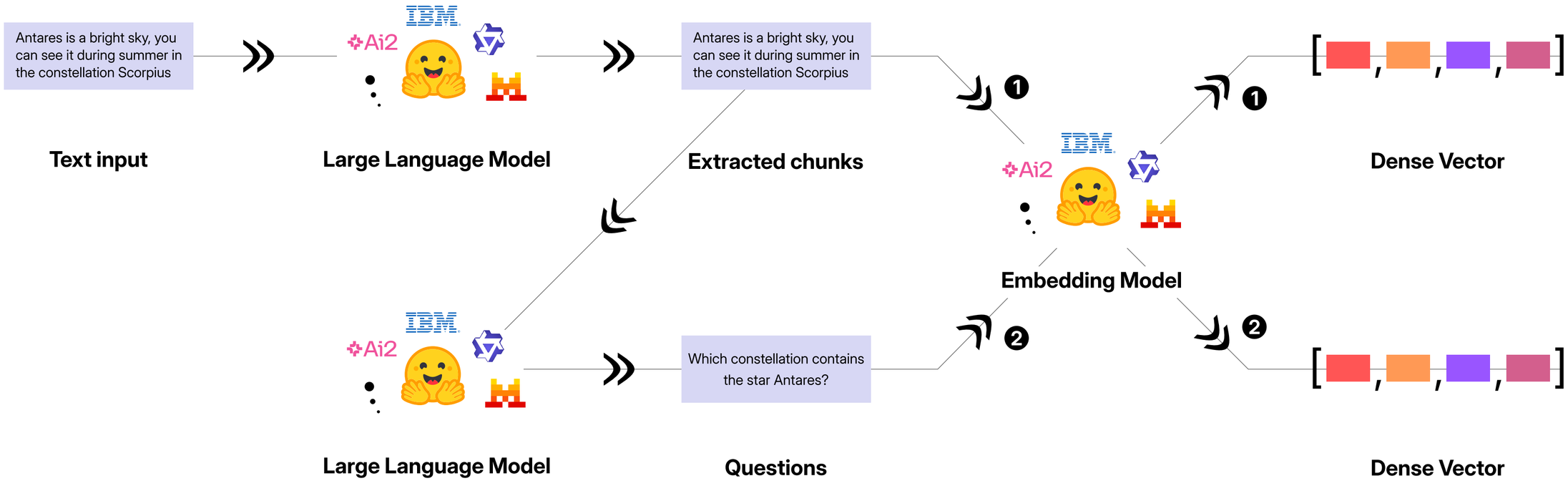
Summary | Performance per Model Size | Performance per Task Type | Performance per task | Task information

Filter...

Rank (Bor…	Model	Zero-shot	Memory U…	Number of P…	Embedding D…	Max Tokens	Mean (T…	Mean (TaskT…	Bitext …	Classification
1	llama-embed-nemotron-8b	99%	28629	7B	4096	32768	69.46	61.09	**81.72**	73.21
2	gemini-embedding-001	99%	Unknown	Unknown	3072	2048	68.37	59.59	79.28	71.82
3	Qwen3-Embedding-8B	99%	28866	7B	4096	32768	**70.58**	**61.69**	80.89	**74.00**
4	Qwen3-Embedding-4B	99%	15341	4B	2560	32768	69.45	60.86	79.36	72.33
5	Qwen3-Embedding-0.6B	99%	2272	595M	1024	32768	64.34	56.01	72.23	66.83
6	gte-Qwen2-7B-instruct	⚠️ NA	29040	7B	3584	32768	62.51	55.93	73.92	61.55
7	Linq-Embed-Mistral	99%	13563	7B	4096	32768	61.47	54.14	70.34	62.24
8	multilingual-e5-large-instruct	99%	1068	560M	1024	514	63.22	55.08	80.13	64.94
9	embeddinggemma-300m	99%	578	307M	768	2048	61.15	54.31	64.40	60.90
10	SFR-Embedding-Mistral	96%	13563	7B	4096	32768	60.90	53.92	70.00	60.02
11	text-multilingual-embedding-002	99%	Unknown	Unknown	768	2048	62.16	54.25	70.73	64.64

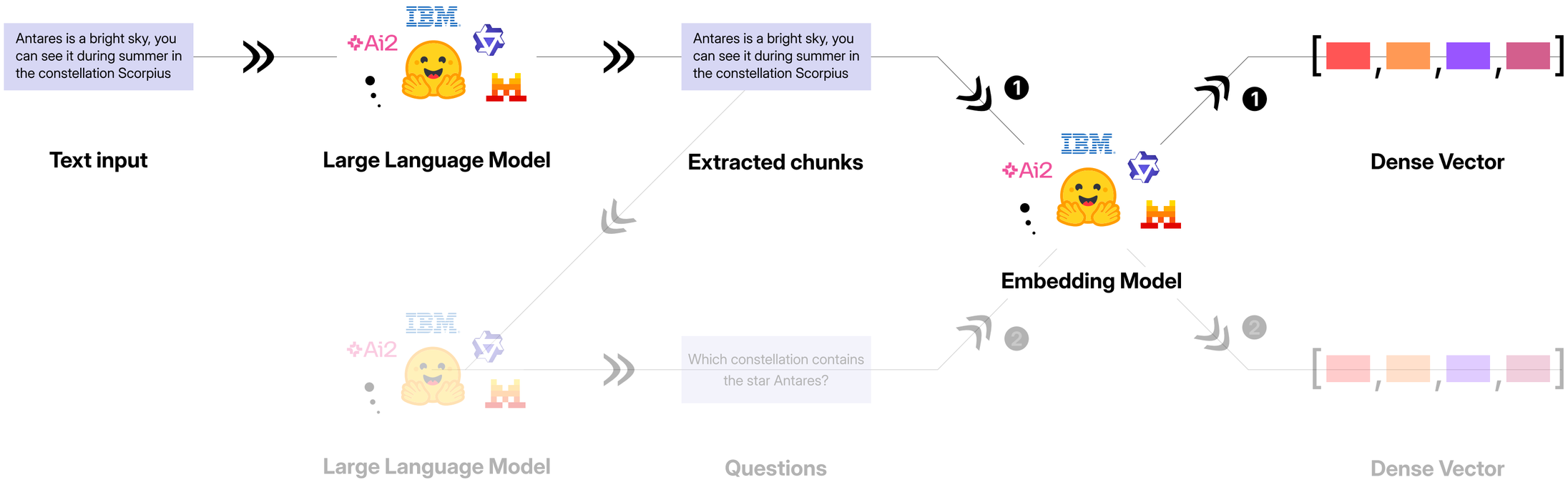
Text embeddings

- Here is the updated pipeline.



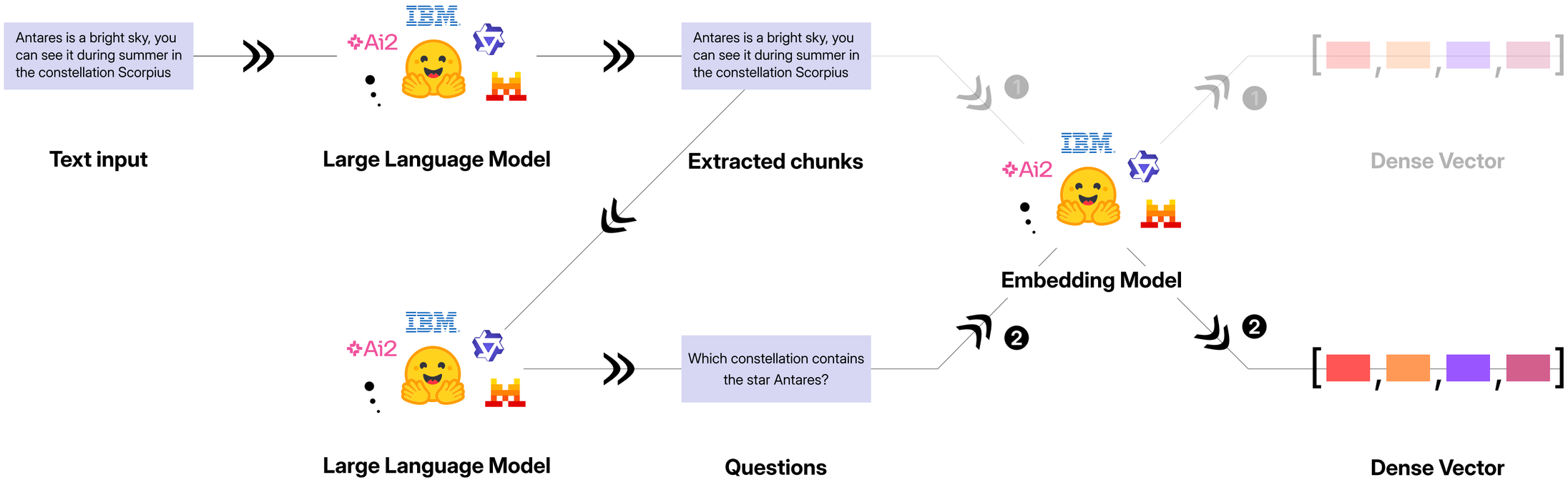
Text embeddings

- Here is the updated pipeline.



Text embeddings

- Here is the updated pipeline.



Benchmark the models

Benchmark the models

- You will compare the performance of the models in two different ways:
  - Manually.
  - With the Ranx library.
- The performance of each model will be displayed in a table like this.

#	Model	mrr	recall@1	recall@5	ndcg@5
a	all-minilm-l6-v2	0.7121	0.6127	0.8408	0.7341
b	qwen3-embedding-0.6b	0.8075 acfg	0.7188 a	0.9098 ag	0.8254 acfg
c	gemini-embedding-001	0.7836 a	0.7029 a	0.8886 a	0.8015 a
d	qwen3-embedding-4b	0.8390 abcfg	0.7480 acfg	0.9682 abcfg	0.8684 abcfg
e	qwen3-embedding-8b	0.8307 abcfg	0.7401 acg	0.9496 abcfg	0.8559 abcfg
f	text-embedding-3-small	0.7851 a	0.7056 a	0.8859 a	0.8028 a
g	text-embedding-3-large	0.7847 a	0.7003 a	0.8780 a	0.7991 a

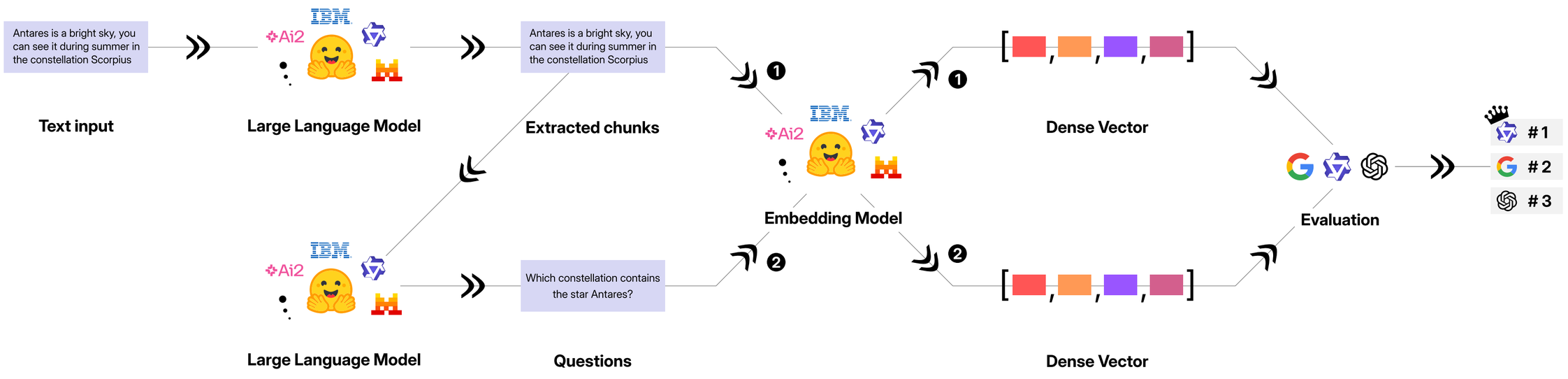
Benchmark the models

- You will compare models **side-by-side** to get the count of **wins, ties, and losses**.
- You will learn about the **metrics** we use to compare embedding models.
- You will use **statistical tests** to **analyze the difference in performance** between two models.

Model A	Model B	Metric	Wins (A)	Ties	Losses (A)	p-value
all-minilm-l6-v2	qwen3-embedding-0.6b	mrr	37	229	111	0.000
all-minilm-l6-v2	qwen3-embedding-0.6b	recall@1	15	307	55	0.000
all-minilm-l6-v2	qwen3-embedding-0.6b	recall@5	10	331	36	0.000
all-minilm-l6-v2	qwen3-embedding-0.6b	ndcg@5	29	252	96	0.000
all-minilm-l6-v2	gemini-embedding-001	mrr	36	233	108	0.000
all-minilm-l6-v2	gemini-embedding-001	recall@1	17	309	51	0.000
all-minilm-l6-v2	gemini-embedding-001	recall@5	8	343	26	0.004
all-minilm-l6-v2	gemini-embedding-001	ndcg@5	29	264	84	0.000
all-minilm-l6-v2	qwen3-embedding-4b	mrr	25	231	121	0.000
all-minilm-l6-v2	qwen3-embedding-4b	recall@1	16	294	67	0.000
all-minilm-l6-v2	qwen3-embedding-4b	recall@5	2	325	50	0.000
all-minilm-l6-v2	qwen3-embedding-4b	ndcg@5	22	240	115	0.000

Benchmark the models

- The dense vectors are used in the **evaluation step** to rank the models.
- The output of the evaluation is a **rich comparison between the models**.



Benchmark the models - Metrics

- Metrics are used to **assess the performance** of an embedding model.
- There are many metrics that you can use such as:
  - Precision
  - Recall
  - F1
  - R-Precision
  - Bpref
  - Mean Reciprocal Rank (MRR)
  - Normalized Discounted Cumulative Gain (NDCG)
- Check this list of metrics.
- In this course, you will use **MRR, NDCG@K, and Recall@K**

Benchmark the models - Metrics

Mean Reciprocal Rank (MRR)

- This metric measures how quickly the first relevant result appears in the ranked list.
- The equation is:

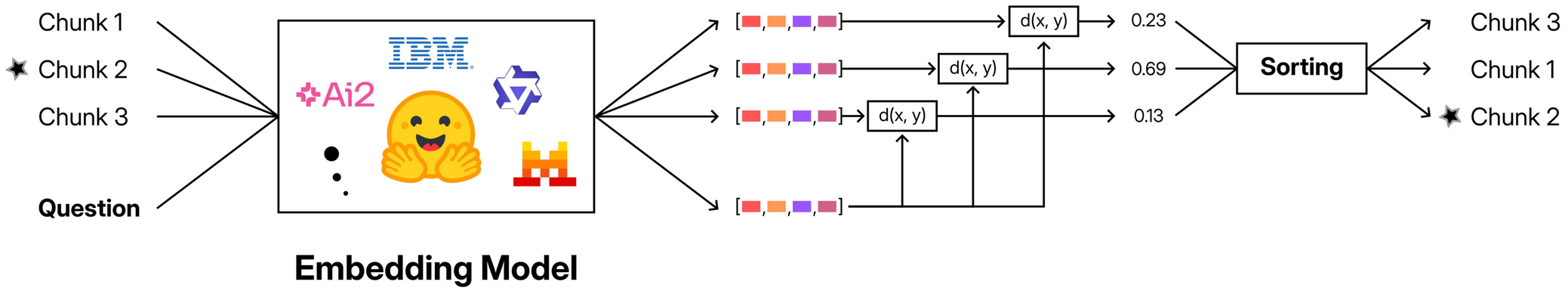
$$MRR = \frac{1}{rank}$$

- **rank** is the position of the first relevant document.

Benchmark the models - Metrics

Mean Reciprocal Rank (MRR)

- Example:
  - You have three chunks where **one chunk answers the question** (marked with a ***).
  - To rank the chunks you follow this pipeline.

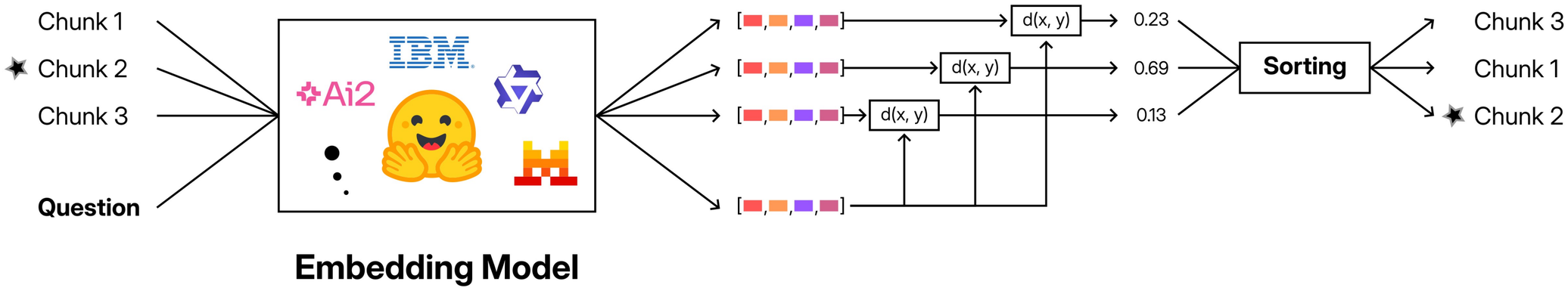


Benchmark the models - Metrics

Mean Reciprocal Rank (MRR)

- Example:
  - You have three chunks where **one chunk answers the question** (marked with a ***).
  - To rank the chunks you follow this pipeline.

$$MRR = \frac{1}{rank} = \frac{1}{3} = 0.33$$



Benchmark the models - Metrics

Recall@K

- Recall@K indicates whether the relevant document appears **within the top-K results**.
- For datasets with **one relevant document**, the equation is:

$$Recall@K = \frac{r}{rank}$$

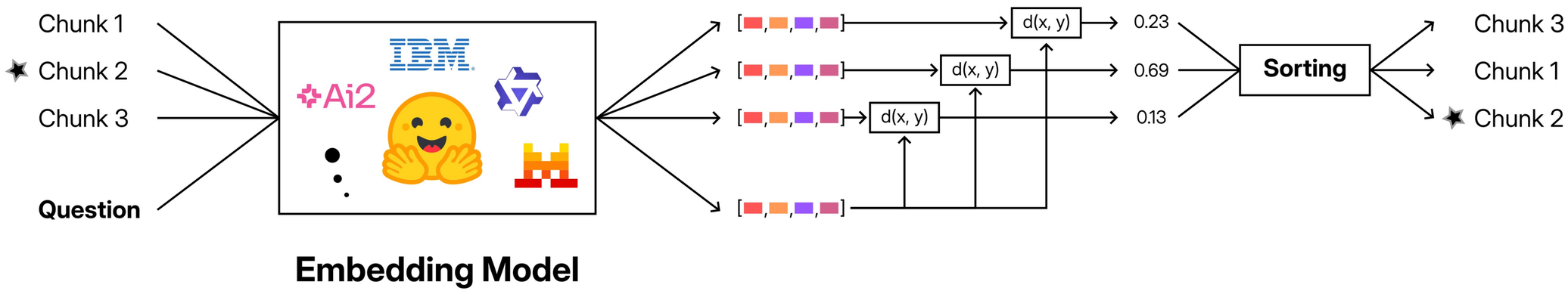
- **r** is 1 if the relevant document is in the top-K, otherwise 0.
- **rank** is the position of the first relevant document.

Benchmark the models - Metrics

Recall@K

- Example:
  - The recall score is good in this case:

$$Recall@3 = \frac{r}{rank} = \frac{1}{1} = 1$$



Benchmark the models - Metrics

NDCG@K (Normalized Discounted Cumulative Gain)

- NDCG@K considers the position of the relevant document.
- The equation is:

$$\mathrm{DCG@K} = \begin{cases} \dfrac{1}{\log_2(\mathrm{rank}+1)}, & \text{if } \mathrm{rank} \le K \\ 0, & \text{otherwise} \end{cases} \quad \Rightarrow \quad \mathrm{NDCG@K} = \frac{\mathrm{DCG@K}}{\mathrm{IDCG@K}} = \mathrm{DCG@K}$$

$$\mathrm{IDCG@K} = 1$$

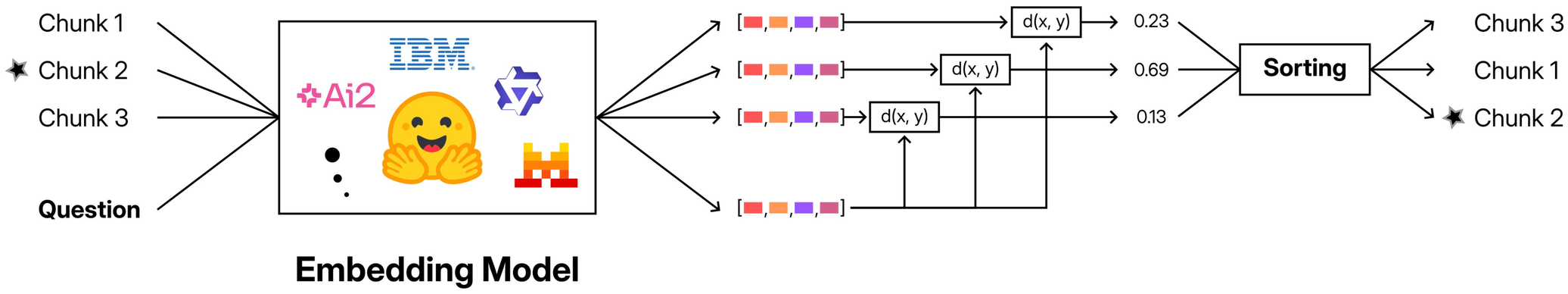
- **rank** is the position of the first relevant document.

Benchmark the models - Metrics

NDCG@K (Normalized Discounted Cumulative Gain)

- Example:
  - Unlike recall, NDCG@K is worst because the document appeared last.

$$\text{NDCG@3} = \frac{1}{\log_2(\text{rank}+1)} = \frac{1}{\log_2(3+1)} = \frac{1}{2} = 0.5$$



Benchmark the models – Statistical tests

- After applying those equations to every model, you get a table like this.
- What is the meaning of the characters inside the cells?
- **0.8075 acfg** means that the *qwen3-embedding-0.6b* **is better** than:
  - *all-minilm-l6-v2 (a)*
  - *gemini-embedding-001 (c)*
  - *text-embedding-3-small (f)*
  - *text-embedding-3-large (g)*
- The performance of *qwen3-embedding-0.6b* is **not due to chance because it is validated statistically.**

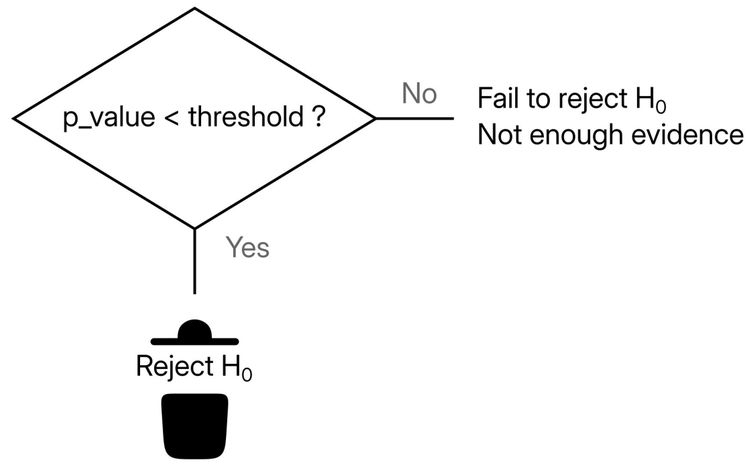
#	Model	mrr	recall@1	recall@5	ndcg@5
a	all-minilm-l6-v2	0.7121	0.6127	0.8408	0.7341
b	qwen3-embedding-0.6b	0.8075 acfg	0.7188 a	0.9098 ag	0.8254 acfg
c	gemini-embedding-001	0.7836 a	0.7029 a	0.8886 a	0.8015 a
d	qwen3-embedding-4b	0.8390 abcfg	0.7480 acfg	0.9682 abcfg	0.8684 abcfg
e	qwen3-embedding-8b	0.8307 abcfg	0.7401 acg	0.9496 abcfg	0.8559 abcfg
f	text-embedding-3-small	0.7851 a	0.7056 a	0.8859 a	0.8028 a
g	text-embedding-3-large	0.7847 a	0.7003 a	0.8780 a	0.7991 a

Benchmark the models – Statistical tests

- When you benchmark models, a better mean score (like **NDCG@10**) might just be due to **luck or random noise** on your specific set of test queries.
- Statistical tests help you determine if the **performance difference is real** (statistically significant) or just a **random fluke**.
- These tests work by **challenging** a default assumption called the **null hypothesis**.
- After running the test, you get a **p-value**. This is the probability of seeing your results **if the null hypothesis were true**.

The two models are identical, and any difference in their mean scores is just due to random chance.

The null hypothesis (H_0)



Benchmark the models – Statistical tests

- **Ranx** provides tests to do this, including:
  - Fisher's randomization test
  - Two-sided paired student's t-test
  - Tukey's HSD test
- Let's benchmark two embedding models, **model A** (baseline) and **model B** (new model).
- We test them on **5 queries** using the metric **NDCG@10**.
- Model B's mean score is higher! But **is it significantly better, or just lucky?**
- Our goal is to see if we have enough evidence to **reject** this claim.

Query	Model A	Model B
Q1	0.5	0.6
Q2	0.4	0.3
Q3	0.7	0.8
Q4	0.5	0.7
Q5	0.6	0.6
Mean score	**0.54**	**0.6**

There is no real difference between Model A and Model B. The mean scores are different only because of random chance on this set of queries.

The null hypothesis (H_0)

Benchmark the models – Paired t-test

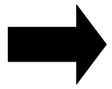
- The t-test looks at the **paired differences for each query**.
- It checks **if the mean** of these differences (0.06) **is significantly different from 0**.
- **Result:** A **p-value** is calculated.
- If **p < 0.05** (threshold): Reject the null hypothesis. The improvement is statistically significant.
- If **p > 0.05**: Can’t reject the null hypothesis. Can't be sure the improvement is real.

Query	Model A	Model B	Difference (B - A)
Q1	0.5	0.6	0.1
Q2	0.4	0.3	-0.1
Q3	0.7	0.8	0.1
Q4	0.5	0.7	0.2
Q5	0.6	0.6	0.0
Mean score	**0.54**	**0.6**	**0.06**

Benchmark the models – Randomization test

- This test simulates the **random chance** of the null hypothesis.
- **Original difference**: *0.06*
- **Shuffle**: What if the scores were just random? Randomly swap the scores for Q2 and Q4.
- **New difference**: *abs(0.56 - 0.58) = 0.02*

Query	Model A	Model B
Q1	0.5	0.6
Q2	0.4	0.3
Q3	0.7	0.8
Q4	0.5	0.7
Q5	0.6	0.6
Mean score	**0.54**	**0.6**



Query	Model A	Model B
Q1	0.5	0.6
Q2	0.3	0.4
Q3	0.7	0.8
Q4	0.7	0.5
Q5	0.6	0.6
Mean score	**0.56**	**0.58**

Benchmark the models – Randomization test

- This test simulates the **random chance** of the null hypothesis.
- **Original difference**: *0.06*
- **Shuffle**: What if the scores were just random? Randomly swap the scores for Q2 and Q4.
- **New difference**: *abs(0.56 - 0.58) = 0.02*
- The test **shuffles the pairs thousands of times** to see what differences are possible just by chance.
- Then counts how many shuffles (e.g., 850 of them) resulted in a difference of **0.06 or more**.
- **p_value** = *850 / 10,000 = 0.085*
- This **p-value > 0.05**. The null hypothesis **can’t be rejected**. You can’t conclude that Model B is significantly better.

Benchmark the models – Randomization test

- Ranx gives you the ability to look at the models side-by-side.
- You can see the count of **wins, ties, and losses**.
- Observe how the **p_value** is always less than the threshold when a model is dominating.

Model A	Model B	Metric	Wins (A)	Ties	Losses (A)	p-value
all-minilm-l6-v2	qwen3-embedding-0.6b	mrr	37	229	111	0.000
all-minilm-l6-v2	qwen3-embedding-0.6b	recall@1	15	307	55	0.000
all-minilm-l6-v2	qwen3-embedding-0.6b	recall@5	10	331	36	0.000
all-minilm-l6-v2	qwen3-embedding-0.6b	ndcg@5	29	252	96	0.000
all-minilm-l6-v2	gemini-embedding-001	mrr	36	233	108	0.000
all-minilm-l6-v2	gemini-embedding-001	recall@1	17	309	51	0.000
all-minilm-l6-v2	gemini-embedding-001	recall@5	8	343	26	0.004
all-minilm-l6-v2	gemini-embedding-001	ndcg@5	29	264	84	0.000
all-minilm-l6-v2	qwen3-embedding-4b	mrr	25	231	121	0.000
all-minilm-l6-v2	qwen3-embedding-4b	recall@1	16	294	67	0.000
all-minilm-l6-v2	qwen3-embedding-4b	recall@5	2	325	50	0.000
all-minilm-l6-v2	qwen3-embedding-4b	ndcg@5	22	240	115	0.000

Benchmark the models – Randomization test

- Ranx gives you the ability to look at the models side-by-side.
- You can see the count of **wins, ties, and losses**.
- Observe how the **p_value** is always less than the threshold when a model is dominating.
- It is always large when the models are similar in performance.

gemini-embedding-001	text-embedding-3-small	mrr	**58**	271	**48**	**0.890**
gemini-embedding-001	**text-embedding-3-small**	recall@1	**20**	336	**21**	**1.000**
gemini-embedding-001	text-embedding-3-small	recall@5	**13**	352	**12**	**1.000**
gemini-embedding-001	text-embedding-3-small	ndcg@5	**42**	296	**39**	**0.905**
gemini-embedding-001	text-embedding-3-large	mrr	**56**	277	**44**	**0.909**
gemini-embedding-001	text-embedding-3-large	recall@1	**14**	350	**13**	**1.000**
gemini-embedding-001	text-embedding-3-large	recall@5	**12**	357	**8**	**0.501**
gemini-embedding-001	text-embedding-3-large	ndcg@5	**37**	310	**30**	**0.787**
qwen3-embedding-4b	qwen3-embedding-8b	mrr	**44**	302	**31**	**0.326**
qwen3-embedding-4b	qwen3-embedding-8b	recall@1	**14**	352	**11**	**0.691**
qwen3-embedding-4b	qwen3-embedding-8b	recall@5	**9**	366	**2**	**0.072**
qwen3-embedding-4b	qwen3-embedding-8b	ndcg@5	**41**	308	**28**	**0.115**
qwen3-embedding-4b	text-embedding-3-small	mrr	**81**	272	**24**	**0.000**