

Multimedia Retrieval

Chapter 1: Introduction

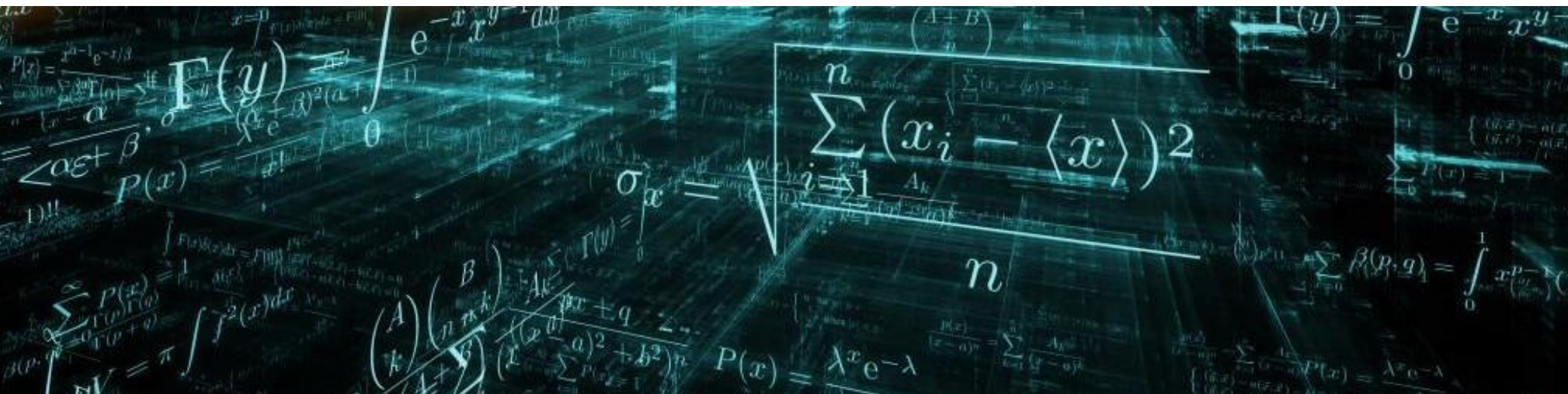
Dr. Roger Weber, roger.weber@gmail.com

[1.1 The Long Road to Modern Search](#)

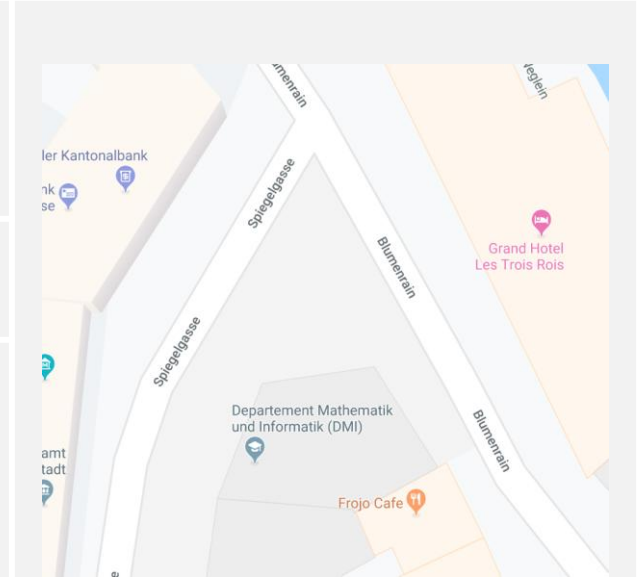
[1.2 Breakthroughs Behind Modern Retrieval](#)

[1.3 How Retrieval Systems Find Answers](#)

[1.4 References & Links](#)



Course ID	15731-01
Lecturer	Dr. Roger Weber, roger.weber@gmail.com
Time	Friday 15:15 - 18:00 (1 st /2 nd hour for theory, 3 rd hour for exercises) Note: changes are announced on web site and / or per e-mail ahead of lectures
Location	<p><u>Physical presence:</u> Seminarraum 05.002, Spiegelgasse 5</p> <p>If physical presence is not possible, we use Zoom Meetings. Please check the schedule for updates. During physical presence lectures, no Zoom meetings and no video recordings are available.</p>
Prerequisites	Basics of programming (Python preferred) Mathematical foundations (for some parts)
Content	Introduction to multimedia retrieval with a focus on classical text retrieval, web retrieval, extraction and machine learning of features for images, audio, and video, index structures, search algorithms, and concrete implementations. The course is touching on classical and current information retrieval techniques and search algorithms.
Exam	Oral exam (30 minutes) in January / February 2026 (tba)
Credit Points	6
Grades	From 1 to 6 with 0.5 steps. 4.0 or higher required to pass exam.
Homepage	<p>WEB: https://dmi.unibas.ch/de/studium/computer-science-informatik/lehrangebot-hs24/15731-lecture-multimedia-retrieval/ ADAM: https://adam.unibas.ch/goto_adam_crs_1995846.html</p> <p>All materials are published in advance. Practical exercises to be submitted to ADAM</p>



Classical Retrieval	1 - Introduction	We will cover the motivation, a short history, the general retrieval process and its variations, and watch demonstrations to get started.
	2 - Classical Text Retrieval	We discuss the main classical retrieval models: Boolean, vector space, and probabilistic. We conclude with BM25, the leading model used in many systems today.
	3 - Performance Evaluation	We evaluate and compare retrieval systems to determine how well the methods described in the chapters perform. Focus is on precision and recall related metrics.
	4 - Advanced Text Processing	We study how to extract improved text features, examine tokenization strategies for machine learning, and discuss various linguistic analysis and transformation methods.
	5 - Index for Text Retrieval	We explore methods to quickly find relevant documents and study Lucene, a widespread library that provides classical and modern text retrieval capabilities.
Advanced Retrieval	6 - Semantic Search	We explore semantic search: first Latent Semantic Indexing, then word embeddings, and finally modern transformer-based semantic search.
	7 - Vector Search	We examine the challenge of searching embeddings and feature vectors. We explain the curse of dimensionality and review techniques used by vector search engines today.
	8 - Retrieval Augmented Generation	We show how large language models can improve responses to users by using retrieved information. We apply this approach to text search.
	9 - Web Search	We study web retrieval, specifically methods that influence rankings using the relationships among documents or web pages.
Multi Modal Retrieval	10 - Multimodal Content Analysis	We explain multimodal content analysis and how to evaluate extracted features using a confusion matrix. We present metadata extraction as a simple method to bridge the semantic gap.
	11 - Visual Features	We cover the human perception of visual signal information and examine several algorithms for extracting features that describe color, texture, and shape aspects found in the images
	12 - Acoustic Features	We cover the human perception of audio signals and study various algorithms for extracting features in both the time and frequency domains.
	13 - Spatiotemporal Features	We present simple methods to describe how videos change over time and across space.
	14 - Classifiers	We study network architectures that extract classifiers from images and audio files to serve as focused content descriptors.
	15 - Multimodal Search	We use transformer-based models to generate improved descriptions and classifiers and examine how to integrate them into the retrieval process.
	99 - ML Methods	We review key machine learning methods used for content analysis and for extracting metadata. This chapter is not part of the exam; it is supplemental material to help you understand the methods discussed in this course.

Timeline and Organization of the course

Date	Theory: 15:15 / 16:15	Practice: 17:05	Where*
Sep 19	1 - Introduction, 2 - Classical Text Retrieval		University*
Sep 26	2 - Classical Text Retrieval	Ex 1 (new)	University*
Oct 3	3 - Performance Evaluation		Zoom**
Oct 10	4 - Advanced Text Processing	Ex 1 (discuss), Ex 2 (new)	Zoom**
Oct 17	5 - Index for Text Retrieval		Zoom**
Oct 24	6 - Semantic Search	Ex 2 (discuss), Ex 3 (new)	University*
Oct 31	7 - Vector Search	Prep Exam	University*
Nov 7	8 - Retrieval Augmented Generation, 9 - Web Search	Ex 3 (discuss), Ex 4 (new), deep learning	University*
Nov 14	10 - Multimodal Content Analysis	Ex 4 (discuss), Ex 5 (new)	University*
Nov 21	11 - Visual Features	Ex 5 (discuss), Ex 6 (new)	University*
Nov 28	<i>No Lessons (Dies Academicus, last Friday in November)</i>		
Dec 5	12 - Acoustic Features	Ex 6 (discuss), Ex 7 (new)	University*
Dec 12	13 - Spatiotemporal Features, 14 - Classifiers	Ex 7 (discuss), Ex 8 (new)	University*
Dec 19	14 - Classifiers, 15 - Multimodal Search	Ex 8 (discuss), Eval & Prep Exam	University*

- **Theory:** Please study the material in advance. During lessons, we will cover the essentials with demonstrations and discussions, but some details will be left for self-study to keep a good pace. Check the schedule and announcements in ADAM. As a general rule, aim to read about one chapter ahead.
- **Practice:** During the third hour, we combine theory and hands-on exercises. We work with Python and software packages related to the retrieval topics we cover. This session is optional, but active participation can help you in the exams, as outlined on the next page

* University: Spiegelgasse 5, Seminarraum 05.002, **no zoom available, no video uploads** after lecture

** Zoom: see meeting link on Web / in ADAM, video uploads after lecture

Exams and how to prepare for them

- Exams are scheduled in January / February 2026 (dates will be announced)
 - each student will have a 30-minute slot
 - slots will be assigned in December based on your preferences and available slots
 - each exam covers three topics, with each topic allocated around 8-10 minutes
 - each topic will include several questions of increasing difficulty, so be accurate and fast to earn maximum points
- Prerequisites for Exams: all exercises are optional, but practical exercises can help to round-up exam grades
 - you don't need to submit theoretical exercises; we'll provide and discuss solutions during the 3rd hour
 - you can submit practical exercises but you won't receive grades; **instead, you earn points for the exam**
 - **you do not earn points for theoretical exercises**
 - points will be converted into an **upgrade of your grade ranging from 0 to 0.3**
 - the upgrade is added to your oral exam grade for the final result (rounded to the nearest 0.5 grade step)
 - Examples:
 1. Student submits some practical exercises, earning an upgrade of 0.2. In the oral exam, the student doesn't perform well and receives a grade of 3.6. Final result: $3.6 + 0.2$ rounded up to 4.0 (pass)
 2. Student earns an upgrade of 0.3 for very good submissions. In the oral exam, the student receives a grade of 5.5, due to some difficulty with challenging questions. Final result: $5.5 + 0.3$ rounded up to 6.0 (pass, excellent)
 3. Student doesn't have time, earns no upgrade, but performs well in the oral exam, receiving a grade of 5.7 struggling only with the toughest questions. Final result: $5.7 + 0.0$ rounded down to 5.5 (pass, very good)
 4. Student doesn't have time, earns no upgrade, and doesn't perform well in the oral exam, receiving a grade of 3.7 with struggles in many questions. Final result: $3.7 + 0.0$ rounded down to 3.5 (failed)
- Submission deadlines:
 - Theoretical exercises: no submissions to ADAM; self correction against distributed solution
 - Practical exercises: submission to ADAM by 31st Dec (23:59); groups of 2 are possible, but you need to provide more / better results to earn points

1.1 The Long Road to Modern Search

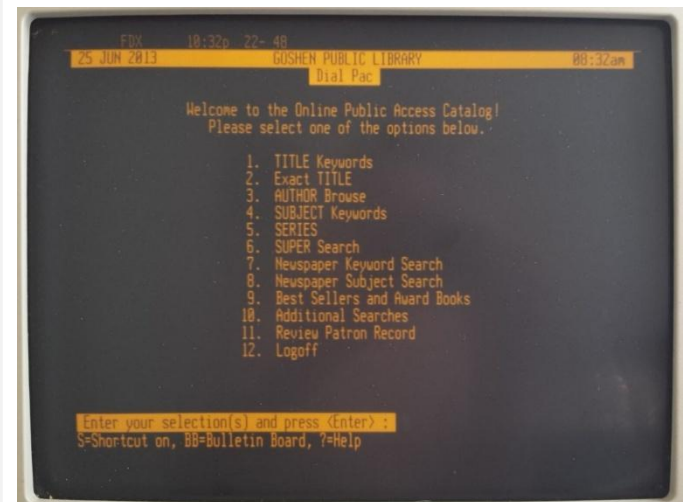
- The study of information retrieval and knowledge management is closely linked to the broader story of how societies create, store, and access knowledge. From early library catalogues to today's vast digital ecosystems, each technological advance has reshaped the ways in which people find and use information. Understanding this progression is essential for students of information science, computer science, and related disciplines because it highlights not only the technical breakthroughs but also the social and economic forces that drive innovation.
- Before computers became common tools for research and communication, information retrieval was a slow, manual process. Scholars and librarians relied on carefully maintained card catalogues, subject indexes, and classification systems to locate relevant works. These methods were effective for the printed collections of the time, but they could not scale to meet the demands of a rapidly growing body of scientific literature and government documentation. By the mid-twentieth century, the need for more efficient retrieval methods had become urgent. Governments, universities, and industries all required faster ways to search large collections of technical documents, and this demand set the stage for the field's first major transformation.
- The arrival of digital computing in the post-war era provided the tools to address these challenges. Early experiments with computerized catalogues and automated indexing showed that machines could, in principle, manage and search much larger volumes of information than human indexers. Researchers began to formalize concepts such as "recall" and "precision" which remain core metrics today. Yet the transition from theory to practical systems was gradual. Early machines were expensive, text collections were limited, and few documents were available in machine-readable form. Nevertheless, the foundations laid during this period would guide the next decades of development.
- The history that follows traces how information retrieval evolved from these early experiments into the sophisticated, large-scale systems of the present. Each decade introduced new models—Boolean retrieval, vector spaces, probabilistic approaches, and eventually neural networks—that improved how systems interpret queries and rank results. At the same time, the volume of data expanded exponentially, requiring continual advances in storage, processing power, and algorithmic efficiency. By exploring these developments chronologically, we can see how the field adapted to shifting technological landscapes, from the first keyword indexes to today's transformer-based semantic search and retrieval-augmented generation. In the following, we provide the context needed to appreciate both the theoretical underpinnings and the practical achievements described in this course.

- **The 1960s: Foundations of Knowledge Management**

- The 1960s saw the rise of computerized information retrieval, built on centuries of manual cataloging. Calvin Mooers had coined the term "information retrieval" in 1950, but serious research did not begin until the 1960s. That decade became a "boom time for information retrieval" because of widespread research and development.
- Important advances included H. P. Luhn's creation of KWIC (Key Word In Context) indexes and Calvin Mooers's development of edge-notched card systems. Western Reserve University's Searching Selector, built by Allen Kent, was one of the era's best-known machines. Most work, however, remained experimental, and there was little computerized retrieval because few texts were machine-readable.
- During the 1960s, researchers established core evaluation metrics for retrieval systems, defining recall and precision. IBM developed STAIRS, the Storage and Information Retrieval System, one of the first large-scale, general-purpose information retrieval systems for text datasets. At Cornell, Gerard Salton began developing SMART, the System for the Mechanical Analysis and Retrieval of Text, which became foundational to modern search technology.

- **Characteristics:**

- **Users:** Researchers, government/industrial R&D groups
- **Use Cases:** Experimental text search, bibliographic retrieval, indexing scientific literature, cataloging documents
- **Key Technologies:** Boolean retrieval, KWIC (Key Word In Context) indexing, edge-notched cards, classification systems (Dewey Decimal)
- **Retrieval model:** Retriever-only, Retriever-Filter
- **Limitations:** Small machine-readable text collections, slow response times, complex Boolean queries required, limited scope, largely experimental, few operational systems



- **The 1970s: The Rise of Computational Models**

- The 1970s marked a turning point when retrieval systems became practical. Widespread use of computer typesetting and word processing produced the machine-readable text needed for large-scale systems. By the end of the decade, most printed material passed through computer input stages, creating large repositories for retrieval systems.
- This decade saw the rise of classical retrieval models that would shape the field for years. Gerard Salton's vector space model represented documents and queries as vectors in a high-dimensional space, allowing their similarity to be measured mathematically. During the same period, Stephen E. Robertson, Karen Sparck Jones, and others developed a probabilistic retrieval framework, which laid the groundwork for probabilistic models. Both approaches were later combined into BM25, which still produces excellent retrieval results.
- Large-scale commercial systems like the Lockheed Dialog system came into operation in the early 1970s. The MEDLARS (Medical Literature Analysis and Retrieval System) became operational, providing computerized access to biomedical literature. These systems demonstrated the practical viability of automated information retrieval on a commercial scale.

- **Characteristics:**

- **Users:** Researchers, Librarians & Government analysts
- **Use Cases:** Academic & biomedical literature search, Library catalog/book search, Bibliographic & reference retrieval
- **Key Technologies:** Boolean, vector space, probabilistic retrieval
- **Retrieval model:** Retriever-only, Retriever-Filter, Retriever-Ranker
- **Limitations:** Limited coverage, Expensive access (subscriptions, terminals, telecom), Slow response times, Complex queries requiring trained intermediaries, Often abstract/metadata-only (not full-text)



Utility of MEDLARS

- : MEDLARS is known as the biggest bibliographical database of international level.
- It is important not only for the experts of medical science, but also for other scientists , sociologists, trades and businessmen etc.
- It has become more and more useful and its scope is very much wide.
- It also has been useful for the people which are not concerned with medical sciences.

INDEXING

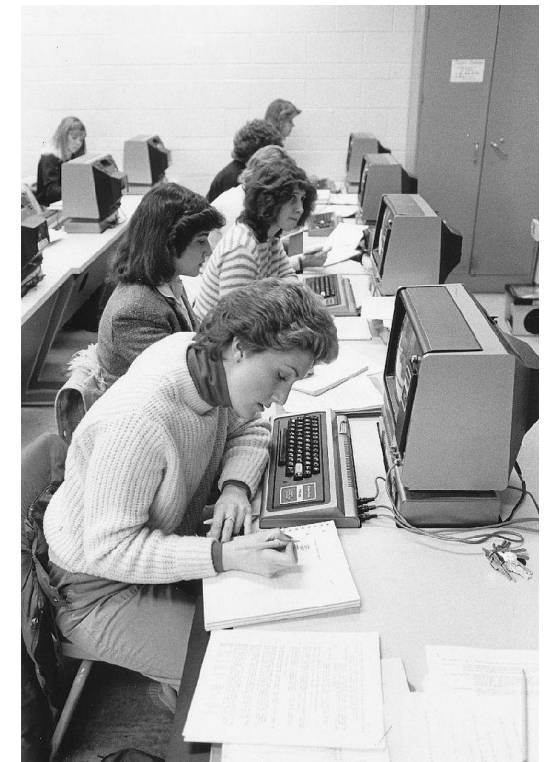
- The MEDLARS data base is created by indexers
- who analyze and describe the contents of journal articles by means of carefully selected terms.
- These terms, called *subject headings* or main headings, are contained in an authorized list.

• The 1980s: Probabilistic Models and Boolean Systems




- In the 1980s, retrieval models developed in the previous decade were refined and widely adopted. Boolean retrieval systems, derived from century-old indexing practices, became the dominant commercial approach. These systems let users combine search terms with logical operators (AND, OR, NOT), but they remained difficult for new users to operate effectively.
- The probabilistic retrieval approach continued to develop, notably advancing the Binary Independence Retrieval (BIR) model. Robertson formalized its theoretical basis, the Probability Ranking Principle, showing that ordering documents by their probability of relevance from highest to lowest yields the best retrieval performance.
- The Okapi information retrieval system was developed at London's City University during this decade; it later gave its name to the well known BM25 ranking function. Latent Semantic Indexing (LSI) appeared as a new method that uses singular value decomposition to reveal hidden relationships between terms and documents.

• Characteristics:

- **Users:** Researchers, librarians, and commercial search system operators
- **Use Cases:** Academic literature retrieval, library book searches, bibliographic/reference lookup
- **Key Technologies:** Boolean retrieval, vector space model, probabilistic retrieval (e.g., Binary Independence Model)
- **Retrieval model:** Retriever-only, Retriever-Filter, Retriever-Ranker
- **Limitations:** Small-scale datasets, slow query response times, high learning curve for formulating queries, costly system implementation and operation



How to Search Using Boolean Operators:

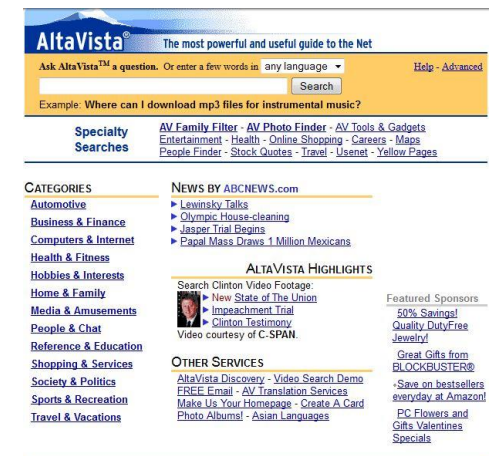
Concept	Search Examples	Results
AND	politics AND media children AND poverty "civil war" AND Virginia	 Results will include both terms
OR	"law enforcement" OR police labor OR labour 60s OR sixties	 Results will include one or both terms
NOT	"civil war" NOT American Caribbean NOT Cuba therapy NOT physical	 Excludes results with the term following NOT

• The 1990s: The Web Revolution and Search Engines

- The 1990s brought a major change with the arrival of the World Wide Web and the first web search engines. When the Web opened to the public in 1991, available information expanded rapidly, and new methods were needed to find it.
- Early search engines such as Archie, Gopher, and WebCrawler appeared to help users find their way through the new online world. Yahoo! Search became one of the first widely used web search services; it began as a directory-based system. Launched in 1995, AltaVista introduced full-text searching and indexed multimedia content.
- The most significant development occurred when Larry Page and Sergey Brin created PageRank at Stanford University in 1996. The algorithm changed web search by ranking pages according to the authority of links pointing to them instead of relying only on keyword matching.
- During this period, machine learning began to be used in information retrieval. Researchers explored neural networks, symbolic learning, and genetic algorithms for various IR tasks. Relevance feedback grew more sophisticated, enabling systems to learn from user interactions.

• Characteristics:

- **Users:** General public, e-commerce consumers, professionals and business users
- **Use Cases:** web search, e-commerce search, business/market intelligence, academia
- **Key Technologies:** Vector Space Retrieval, Probabilistic Retrieval, Web Search Engines, Query Expansion, separation of retrieval & sort
- **Retrieval model:** Retriever-Ranker, (Retriever-Filter)
- **Limitations:** data & index explosion due to exponential growth, large retrieval and indexing costs, good recall values but poor perceived precision (i.e., document not relevant for the user), quality of data

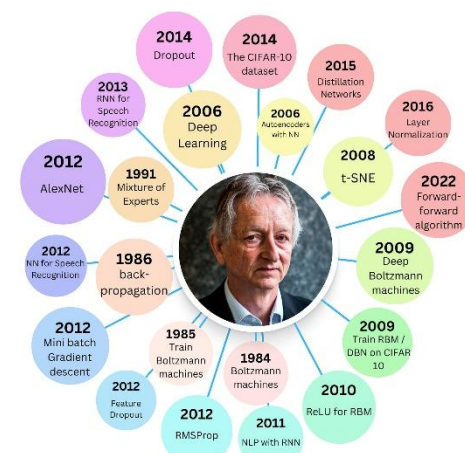
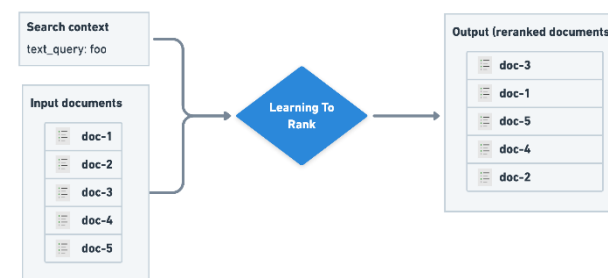
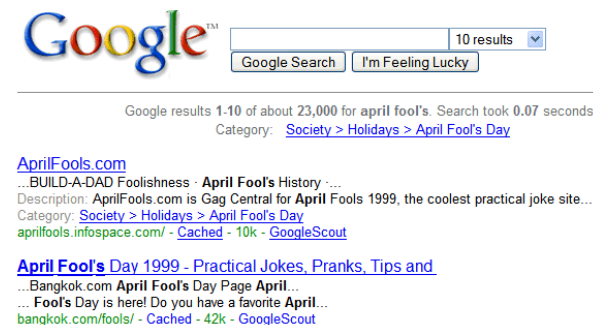


• The 2000s: Machine Learning and Ranking Algorithms

- In the 2000s, information retrieval systems began using advanced machine learning. Learning to Rank (LTR) became an important approach: supervised models were trained to sort search results by relevance. This shift moved ranking away from hand-designed features toward data-driven optimization.
- Based on a probabilistic framework from earlier decades, BM25 became a standard ranking function. It combines term frequency and inverse document frequency with document-length normalization, delivering strong performance across diverse collections.
- Deep learning reemerged after research largely stopped at the end of the 1990s. More powerful computers, especially GPUs, larger datasets, and better training methods brought neural networks back into use. Researchers developed deep belief networks (Hinton, 2006) and applied convolutional neural networks to vision tasks.
- Search engines continued to evolve their algorithms, and Google's PageRank became the basis of its dominance in web search. User behavior signals and personalization began to influence the ranking.

• Characteristics:

- Users: Web search users, e-commerce consumers, data scientists & academics, social media users
- Use Cases: Web search personalization, e-commerce search & recommendation, emergent semantic search research
- Key Technologies: LTR, BM25, personalization via behavior logs
- Retrieval model: Retriever-Ranker (enhanced with ML)
- Limitations: Data/index explosion, high retrieval/indexing cost, data quality, model transparency, update latency



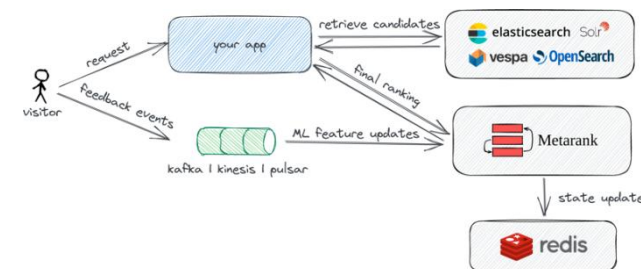
Geoffrey Hinton
Nobel Prize in Physics 2024

• The 2010s: Deep Learning and Neural Information Retrieval

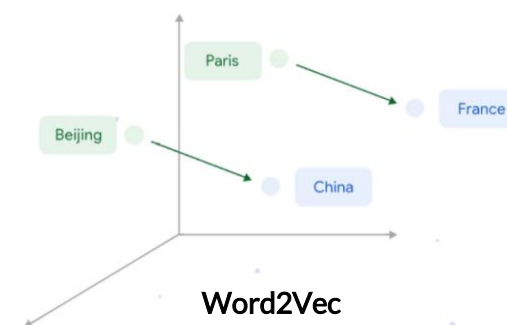
- The 2010s saw a major shift as deep learning techniques were applied to information retrieval. Neural ranking models began using shallow and deep neural networks to rank search results, moving away from traditional handcrafted features toward learned representations.
- Classical learning-to-rank models relied on manual feature engineering. Neural models learned representations that bridged the vocabulary gap between queries and documents; they required large training sets but captured semantic meaning far more effectively.
- Google introduced BERT in 2018 as a major advance in natural language understanding. BERT's ability to capture context and relationships between words allowed for more accurate interpretation of complex queries. Google integrated BERT into its search algorithm in 2019.
- Transformer architectures fundamentally changed how machines handle sequential data, enabling them to capture long-range dependencies in text. Capabilities in 2010s were still constrained lacking datasets, scalability, and performance.

• Characteristics:

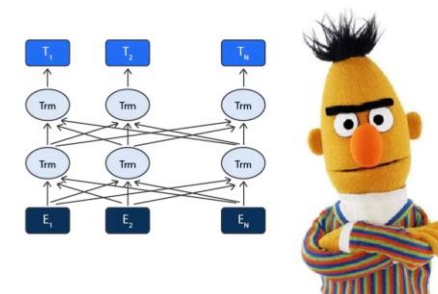
- Users: Web search users, e-commerce consumers, mobile/app users, data scientists & academics
- Use Cases: Web search personalization, e-commerce search & recommendation, conversational search, semantic search research
- Key Technologies: Deep learning (CNNs, RNNs, Transformers), BERT, early neural ranking models, embeddings, large-scale behavior log personalization
- Retrieval model: Neural retriever-ranker pipelines
- Limitations: High computational cost, large data requirements, latency in updates, model interpretability, scalability of training & indexing



<https://opensearch.org/blog/ltr-with-opensearch-and-metarank>



Google

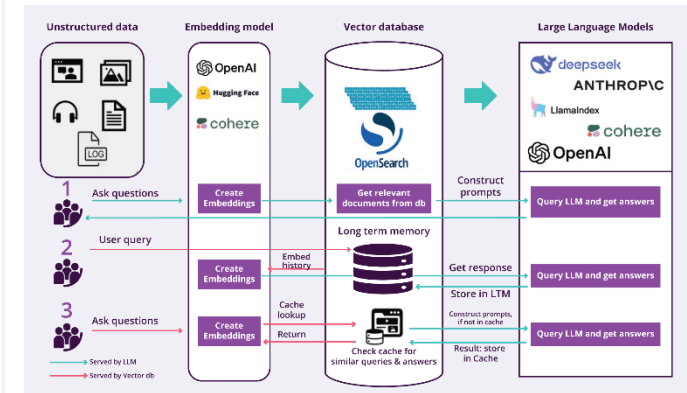


• The 2020s: Semantic Search and Vector Databases

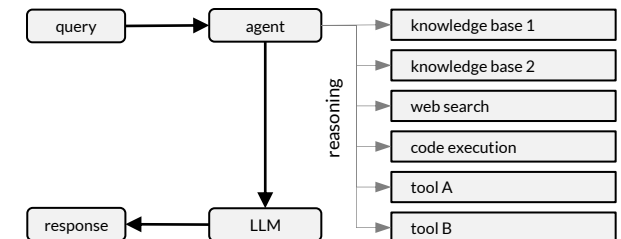
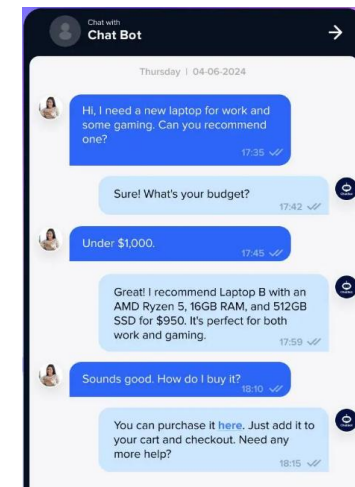
- The 2020s marked the start of semantic search and vector-based retrieval. Transformer models improved information retrieval by enabling systems to understand context and meaning rather than rely solely on keyword matching.
- Vector databases are now essential infrastructure for modern search systems. They allow fast similarity search over high-dimensional embeddings and can match semantically similar content even when keywords do not overlap, for example "car" with "automobile".
- Retrieval-Augmented Generation (RAG) combines large language models with external knowledge retrieval. It lets models access up-to-date information that is not in their training data, reducing hallucinations and improving factual accuracy.
- Agentic information retrieval systems can understand complex information needs, create multiple search strategies, and combine results from different sources. They act as intelligent intermediaries that reason about information requirements, carry out multi-step retrieval, and deliver thorough responses tailored to each context.

• Characteristics:

- Users: Web search users, e-commerce consumers, mobile/app users, data scientists, knowledge workers, AI developers, enterprise teams
- Use Cases: Semantic search, retrieval-augmented generation (RAG), conversational AI, personalized recommendations, knowledge bases, intelligent virtual assistants, multi-step query resolution
- Key Technologies: Large language models, embeddings, vector search
- Retrieval model: Neural retriever-ranker, retriever-generator
- Limitations: Requires high-quality embeddings and up-to-date data, may still hallucinate if retrieval fails, computationally intensive, complexity in multi-source integration

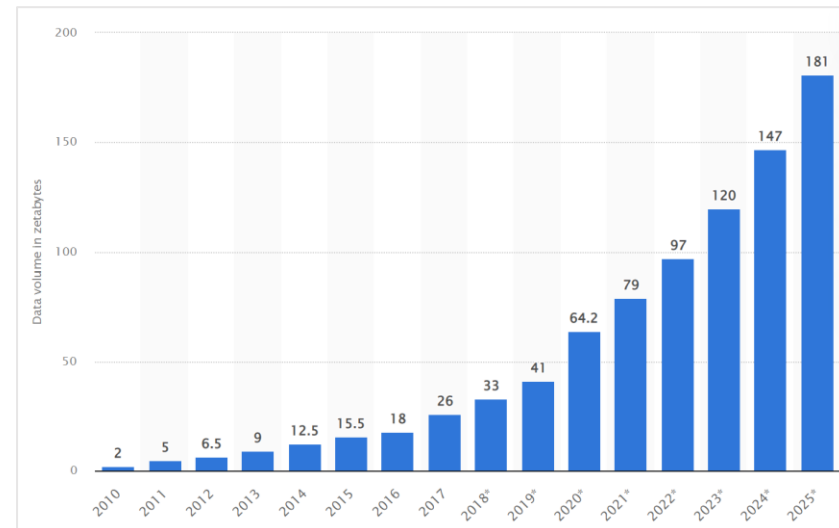


<https://opensearch.org/platform/vector-engine>

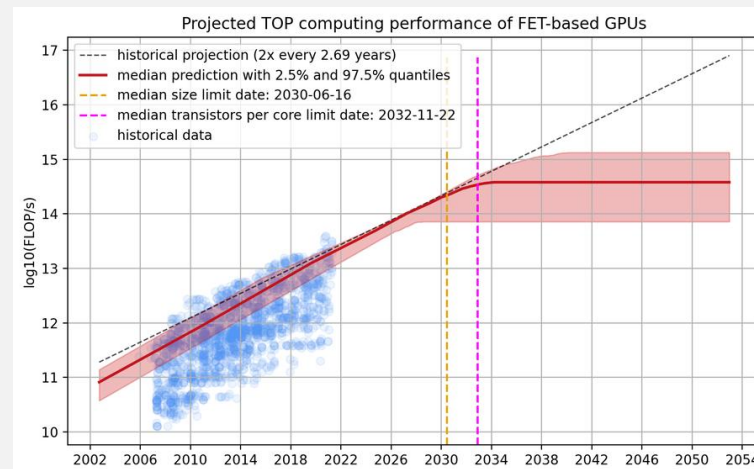
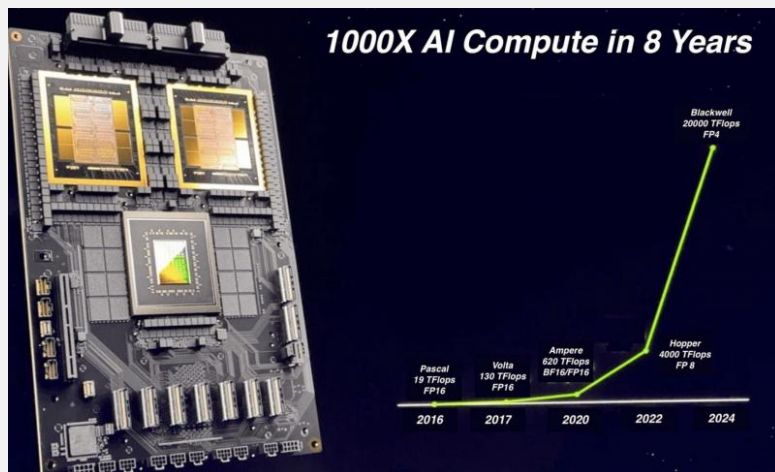


1.2 Breakthroughs Behind Modern Retrieval

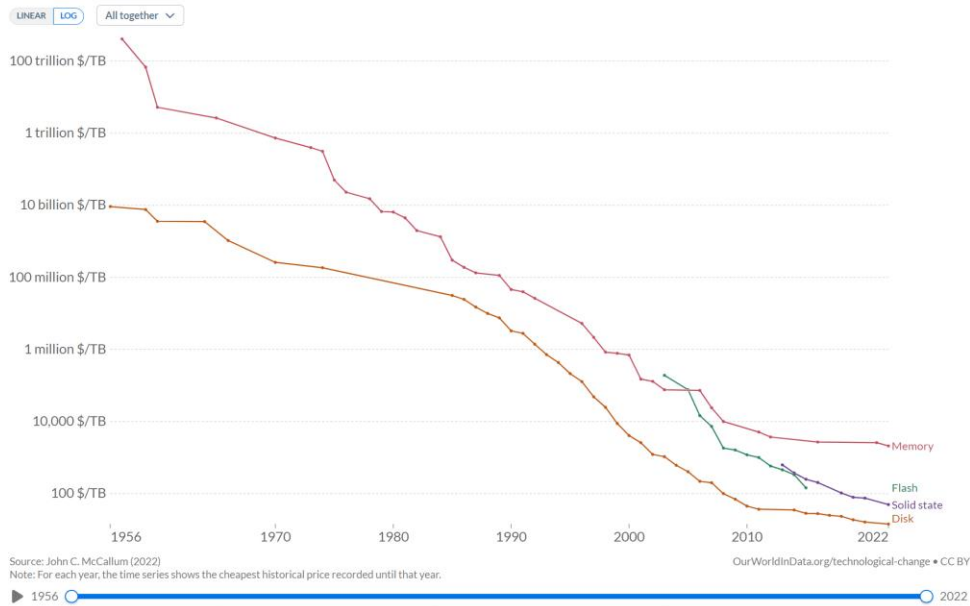
- The history of information retrieval is inseparable from the growth of the data it seeks to organize. Over the past three decades, the scale of digital data has expanded so dramatically that it has transformed not only retrieval methods but also the very infrastructure of computation and storage.
- By the mid-1990s most digital information remained concentrated in business applications and private databases. The internet itself was still modest in scope: in 1999 it contained fewer than one billion web pages. Retrieval systems of the time, still grounded in probabilistic and vector-space models, were adequate for these collections.
- But the explosive adoption of the web, broadband connections, and early mobile devices rapidly outstripped those capabilities. Social media platforms and e-commerce generated continuous streams of user content. At the same time, the emergence of cloud computing separated storage from local physical devices, allowing organizations to keep vastly larger datasets online.
 - In 2010 the world's data volume was estimated at 2 zettabytes. By 2023 its estimate had grown to 120 zettabytes, is expected to exceed 181 zettabytes in 2025, and to top 400-500 zettabytes by 2028.
 - This means that 90% of the world's total digital data has been created in the past 10 years.
 - 1 zettabyte equals 1 billion terabytes or 12 billion 4k videos (1000x all titles listed on IMDb)
- This relentless expansion means that every new search engine query competes with a background of billions of potential matches: a simple search for “ford”, for example, can return more than two billion results. Determining which documents actually satisfy a user's intent is becoming increasingly difficult even for AI based ranking systems.
- The velocity of data creation compounds the challenge. Real-time sources such as news or social media posts generate information that may be relevant for only a short time. News can trend globally and then fade in relevance within an hour. Retrieval systems must therefore index and rank new content close to its creation, or risk becoming obsolete. Traditional pipelines designed for nightly or weekly index updates cannot keep up with this pace.



- Meeting these demands has required dramatic advances in both storage and computation. Physical limits on read and write speeds illustrate the scale of the problem.
 - **Storage** has become much cheaper and more efficient, turning a once scarce resource into an abundant one. Magnetic hard drives steadily increased capacity while lowering cost per gigabyte, followed by optical discs that provided durable, low-cost backups. Solid-state drives improved performance with fast, reliable flash memory and no moving parts. At the same time, cloud storage removed the need to own hardware by pooling vast arrays of magnetic and flash devices across global data centers.
 - **Compute:**
 - A pivotal step came with Nvidia's introduction of CUDA, a parallel computing platform that made GPUs practical for general-purpose computing. Combined with distributed frameworks such as MapReduce, CUDA enabled researchers to process enormous datasets by distributing work across thousands or even millions of machines.
 - As of March 2023, the AMD EPYC 9654 has 96 cores and 192 threads, can perform nearly 1 trillion integer operations per second, and delivers about 550 gigaflops of floating-point performance at 400 watts.
- Even with these capabilities, training a state-of-the-art large language model remains difficult. A single epoch can take several days on a single CPU. Nvidia's RTX 4090 (October 2022), built on a 5-nanometer process, delivers up to 100 teraflops with CUDA at about the same power use, enabling training speeds roughly 200 times faster than a single CPU.



- Illustration of price and space compression of storage



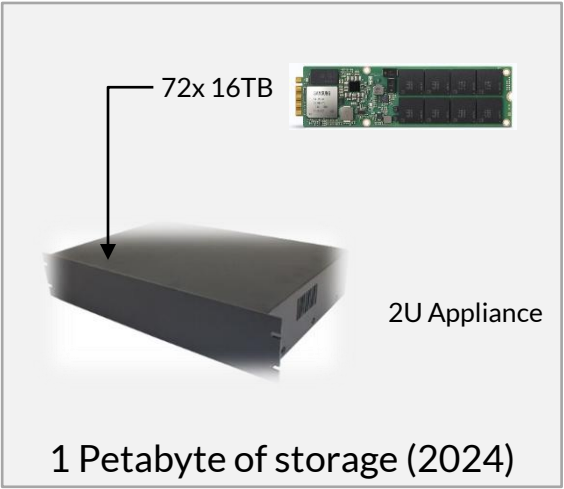
source: <https://ourworldindata.org/>

In the past decades, we have seen price drops of 50% every 14 months. Every 4 years, the costs decreased by an order of magnitude. On the other side, firms still spend the same amount of \$ to increase and replace their storage real estate. As a consequence, the amount of managed storage also grew exponentially and makes it ever more difficult to find relevant information.

Year	1 TB disk	1 TB memory
1960	\$3,600,000,000	\$5,240,000,000,000
1970	\$259,700,000	\$734,000,000,000
1980	\$95,000,000	\$6,480,000,000
1990	\$3,270,000	\$46,000,000
2000	\$4,070	\$700,000
2010	\$45	\$5,100
2020	\$16	\$2,600



21x smaller in 3 years



- So, how long does it take to read 1 Petabyte? All data points as of 2025:
 - Fastest HDD have about 550MB/s read rate
 - Fastest SATA SSD have about 550MB/s read rate
 - Fastest PCI-E 5.0 NVMe SSD have about 14,000MB/s read rate
 - USB 3.2 can handle up to 2,000MB/s transfer rate
 - USB 4.0 can handle up to 40,000MB/s transfer rate
 - PCI-E 7.0 can handle up to 256GB/s in one direction (512GB/s bi-directional)
 - Ethernet (400GbE) can handle up to 50GB/s transfer rate
 - Fibre Channel 256GFC can handle up to 50GB/s (per direction)
 - The largest Internet exchange point (DE-CIX) operates at up to 3TB/s

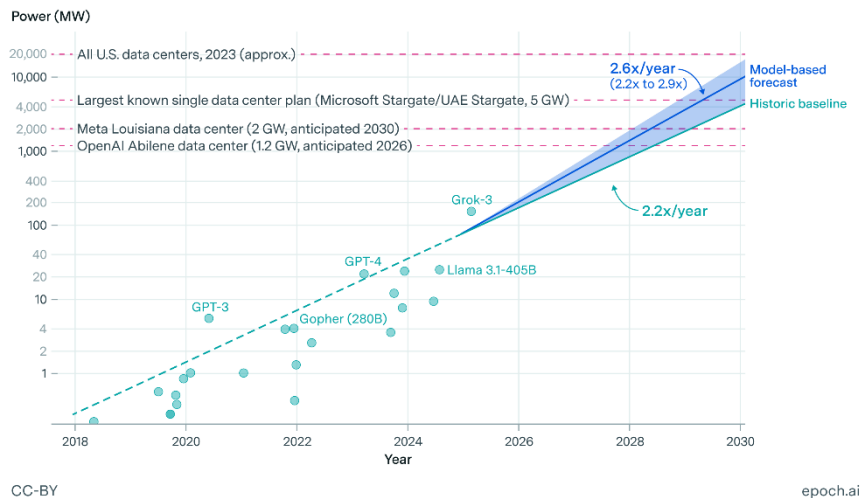
Device / Channel	GB/s	Time to read	1 GB	1 TB	1 PB	1 EB	1 ZB
HDD	0.55		1.8s	30m	21d	58y	57,000y
SATA SSD	0.55		1.8s	30m	21d	58y	57,000y
NVMe SSD	14.00		71ms	1.2m	20h	2.3y	2,263y
USB 3.0	40.00		25ms	25s	6.9h	9.5mo	792y
PCI-E 7.0	256.00		3.9ms	3.9s	1.1h	45d	123y
Ethernet 400GbE	50.00		20ms	20s	5.6h	7.6mo	633y
Fibre 256GFC	50.00		20ms	20s	5.6h	7.6mo	633y
DE-CIX	3,125.00		0.3ms	0.3s	5.3m	3.7d	10y

- Meeting these demands has required dramatic advances in both storage and computation. Physical limits on read and write speeds illustrate the scale of the problem. Engineers have relied on massive parallelism and distributed architectures to keep pace.

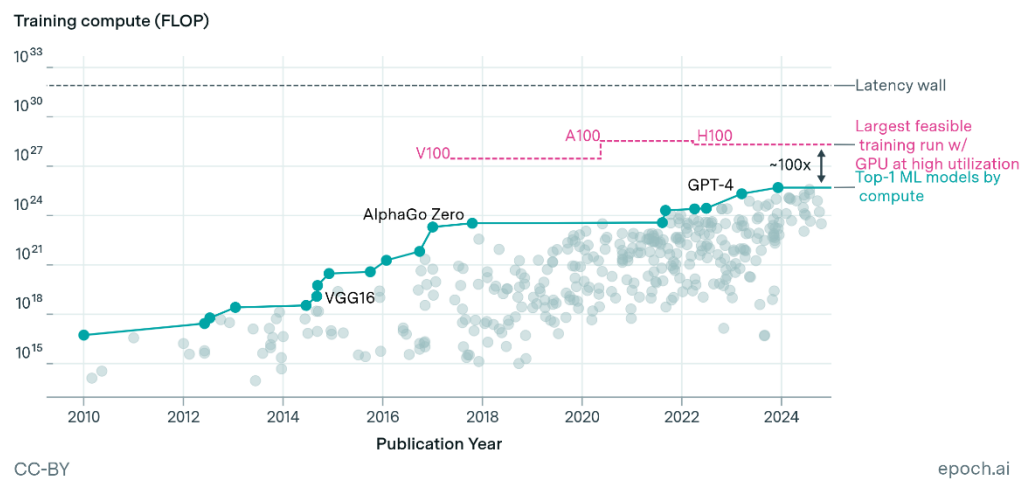
Distributed Computing and the Rise of Modern Foundational Models:

- Distributed computing is essential for foundational models, especially large language models (LLMs). These models can exceed 500 billion parameters and require far more computing power than a single machine can provide. Training OpenAI's GPT-3 is estimated to have required about 3.14×10^{23} floating-point operations (FLOPS). Using the theoretical throughput of a single Nvidia V100 GPU, performing that many operations would have taken many years.
 - **Hardware:** OpenAI used 10,000 V100 GPUs in a high-bandwidth cluster to train GPT-3.
 - **Time:** Based on the V100's theoretical FLOPS, training would take about 13 days on that specific 10,000-GPU cluster using half-precision, or hundreds of years on a single GPU.
 - **Cost:** Estimates suggest the compute cost alone for a single training run was in the millions of dollars.
- Distributed computing also enables parallelism in multiple dimensions. Model parallelism splits the parameters of a single network across devices, while data parallelism processes different portions of the training dataset simultaneously. Pipeline parallelism and sharding techniques further optimize memory usage and throughput. By leveraging these approaches, modern LLMs not only become trainable but can also be fine-tuned, evaluated, and iteratively improved on enormous datasets.

Projected power growth for frontier AI training



Data movement bottlenecks constrain AI scaling



The Future of Information Retrieval: Context, Agency, and Multi-Modal Intelligence

- The next phase of information retrieval will focus on deeper contextual understanding, integration across data types, and agentic capabilities. Multi-modal retrieval, which combines text, images, audio, and video into unified representations, is becoming a major research area. Its use will enable systems to handle complex queries that involve multiple kinds of data and produce more relevant, insightful results than traditional keyword or vector searches.
- Contextual search and user-centered retrieval will become increasingly important. Agentic systems that adapt dynamically to user goals, integrate multi-step reasoning, and combine information from diverse sources will act as intelligent collaborators rather than passive tools. These systems will mediate between users and vast information landscapes, prioritizing relevance, coherence, and user intent.
- At the same time, computing and energy needs are becoming major constraints. Training and running large language models, especially multi-modal ones, requires large clusters of GPUs or specialized chips and uses a lot of power. This creates financial and environmental limits and is driving the development of more efficient architectures, low-precision training, and distributed or federated learning.
- The increasing autonomy of retrieval systems also raises concerns about the role of humans in evaluating information. Language models can generate, locate, and summarize content with minimal human intervention, but the risk is that critical assessment may diminish and the value of information may decline. Systems may present content that is accurate yet contextually irrelevant, creating a perception of understanding without genuine insight.

1.3 How Retrieval Systems Find Answers

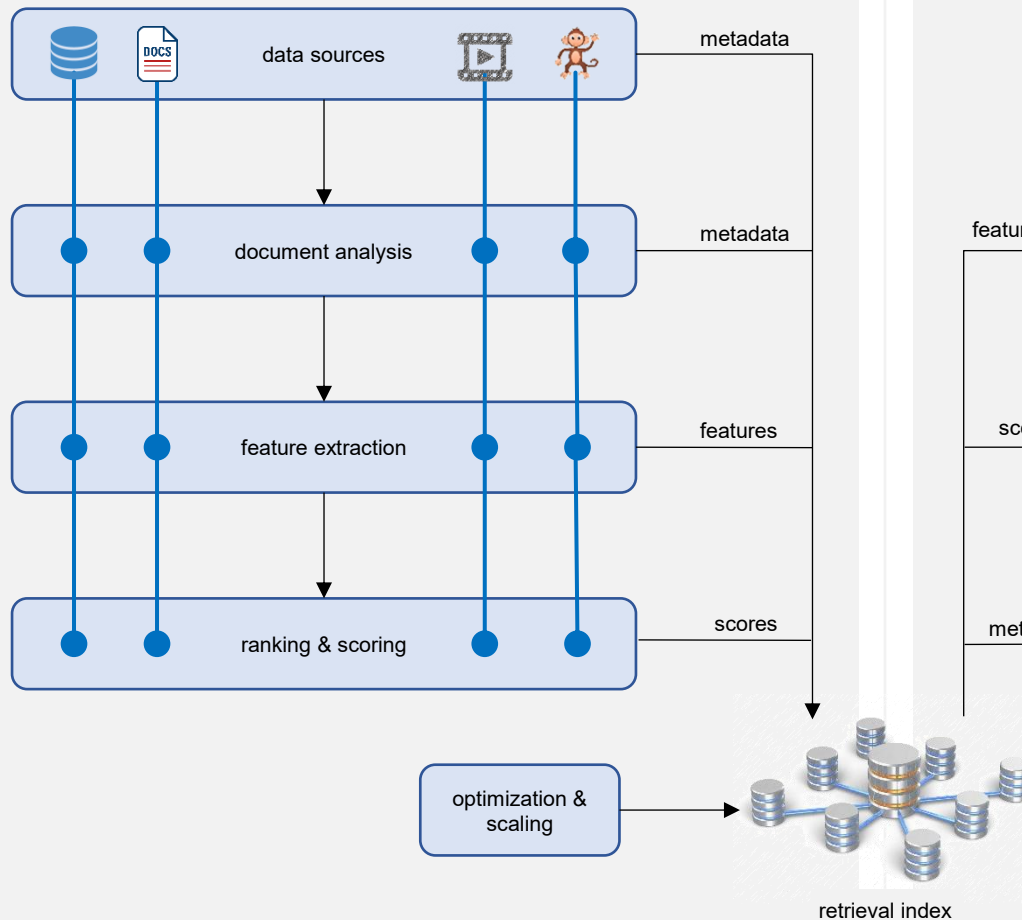
- A retrieval system addresses the following fundamental problem:

Given a set of N documents D_0 to D_{N-1} and a query Q , find a set of documents D_{i_j} with $0 \leq j < k$ that are relevant for the query Q in the context of query originator. Rank the documents such that D_{i_0} is the most relevant document and $D_{i_{k-1}}$ is the least relevant document for query Q in the context of the query originator.

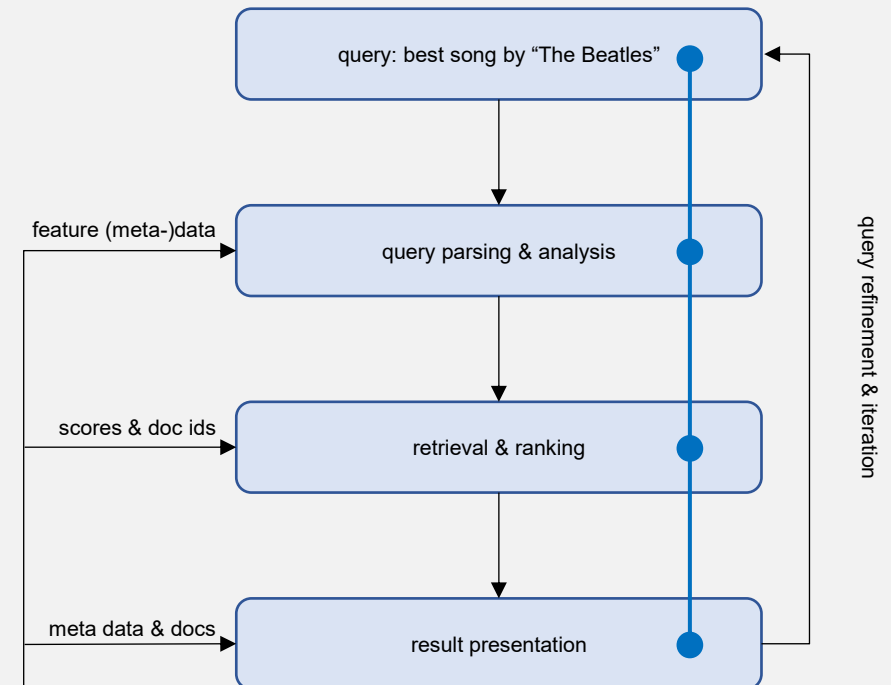
- First, what do we mean with “relevant for query Q in the context of the query originator”
 - Relevancy is the degree to which retrieved information matches a query and meets the query originator's needs or intent. For example, if someone asks "Where can I get a pizza tonight?", the user's location is crucial to providing a relevant result. A restaurant that makes excellent pizza will not be relevant if it is too far away. Social media often presents information without an explicit query; users still expect content that fits their interests rather than random posts. Relevance is also shaped by the user's objective: when searching for a product, it matters whether the user wants to evaluate the item or find the best place to buy it.
 - Objective relevance means factual correctness. If someone searches for the capital of France, Paris is objectively relevant because it is the capital. Subjective relevance depends on individual preferences. When a user asks for “movies to watch,” what counts as relevant will vary with taste and with personal definitions of a good movie. Search engines use feedback from users to improve relevance. Clicks and longer time spent on a result signal that it was useful, and ranking algorithms learn from those signals to show results that better satisfy users.
 - Query-less search is an approach where users discover content without entering an explicit query. Platforms such as Instagram Reels and TikTok analyze user preferences, behavior, and trends to recommend videos that match inferred interests, enabling discovery without typed or spoken queries. Relevance is defined implicitly by how well the recommended content matches the user's inferred interests and intent, even though the user never specifies a query. Because there is no explicit question to measure against, the system estimates relevance from behavioral signals and contextual data. Content is considered relevant when it aligns with these inferred preferences and keeps the user engaged, for example, when the user consistently watches similar videos to completion or interacts positively with them. In other words, relevance is not judged by textual similarity to a query but by predicted satisfaction and engagement based on the user's past behavior and the patterns the system detects.

- Because scanning billions of documents at query time is infeasible given the user's expectation for near-instant responses, retrieval is split into offline processing and online query answering:

Offline processing: concerned with analyzing documents in advance, extracting features, and organizing these features into indexes that allow for fast retrieval. Some systems also adjust their (objective) relevance ranking during this phase (see IDF or Google's PageRank later in the course)



Online query answering: parse the query and extract features like for the documents (e.g., embeddings), retrieve relevant documents, score and rank them, and present to the query originator. Some systems allow users to refine the query and iterate to improve results.



- Searching images, audio, and video is harder than searching text because of the so-called semantic gap. Users typically enter queries as keywords. Text search can match those keywords directly because queries and documents use the same representation. By contrast, images cannot be matched to keywords at the pixel level. For example, there is no clear, fixed relation between the keyword “cat” and the many ways a cat can appear in an image.
- When queries and media use different forms, the retrieval system must translate between them. This mismatch is called the semantic gap:

The semantic gap refers to the disparity between low-level features extracted from multimedia data and the high-level semantics that humans associate with that data.

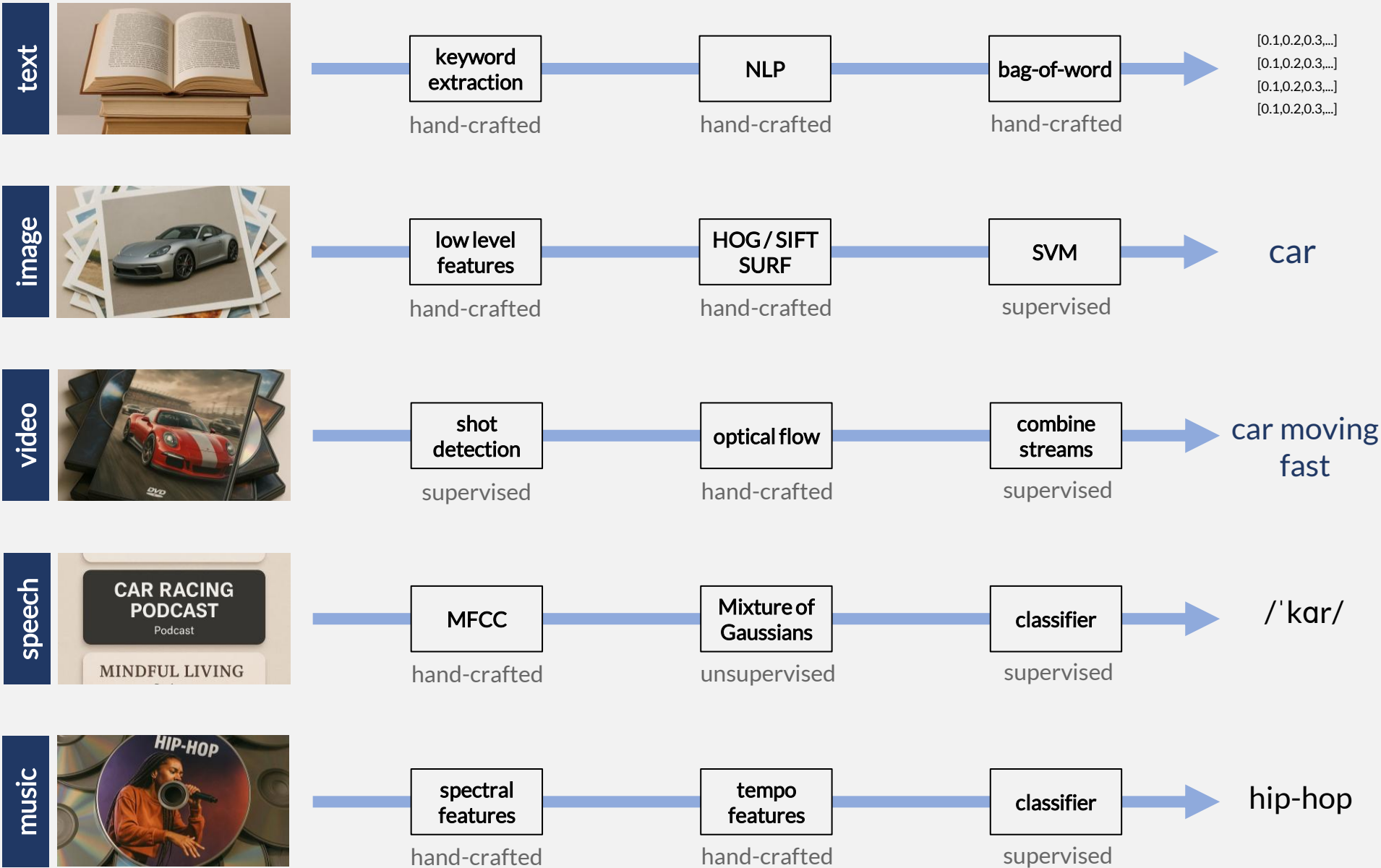
• How to overcome?

- From the 1960s to the 1990s the standard solution was manual tagging and keyword annotation, sometimes implemented as inversion of queries into dedicated folders for prominent items.
- With the web, large, annotated archives of media and metadata emerged through collaborative efforts; examples include IMDb, AllMusic, and MusicBrainz. The metadata was high quality but limited to standard representations of media.
- Since the 2010s, AI technologies have automated keyword generation and context analysis to improve relevance ranking, especially using multi-modal transformer models. What began as simple classification has evolved into producing detailed descriptions of media content.

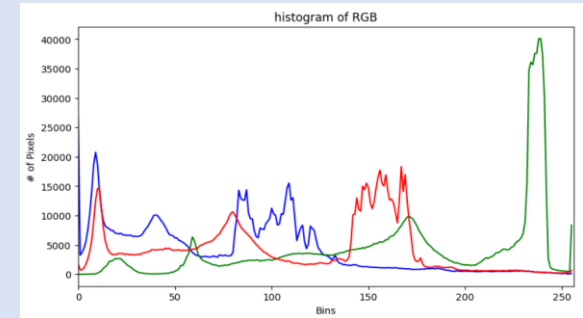
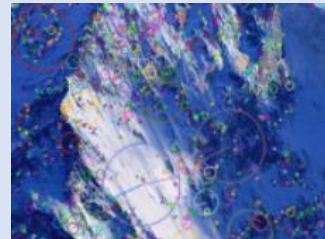
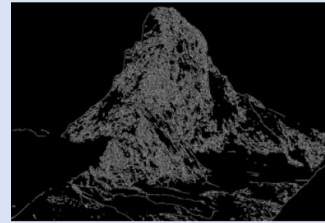
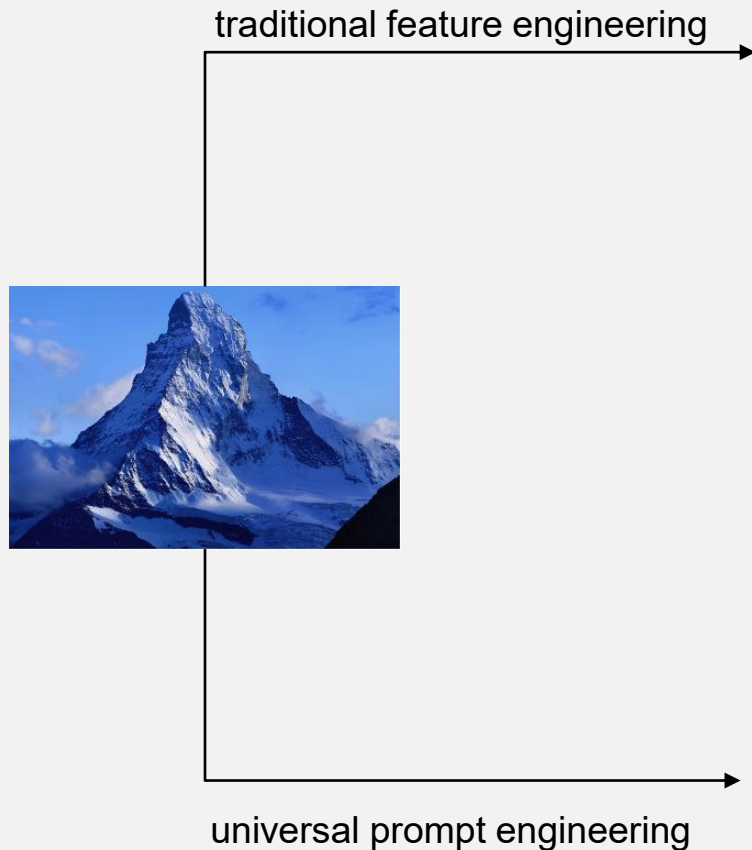


- **Traditional, mostly hand-crafted pipelines for feature extraction** (illustrations on the following pages)
 - Since the start of information retrieval, feature extraction has been a carefully designed, hand-crafted process for describing media content precisely and concisely so data can be retrieved quickly and with high relevance. In this course, we study many of these approaches, which remain useful despite the rise of improved methods based on large language models.
 - For text, common features include term vectors, bag-of-words, and n-grams. Natural language processing steps include stemming, handling synonyms, and resolving homonyms. Systems also extract embeddings and measure term discriminative power with methods such as inverse document frequency. Web and social retrieval add weights from metadata and link analytics like PageRank. Image retrieval once used simple features such as color, texture, and shape and now depends largely on neural network features from classification and representation models. Audio retrieval uses frequency and amplitude domain features, musical features such as pitch and tempo, and speech recognition. Video retrieval adds shot detection and motion analysis.
- **General-purpose feature extractors via prompting and large language models** (illustrations on the following pages)
 - The rise of generative AI has changed how people create content and how they extract useful information from it. Traditional analysis extracted features specific to each media type. Methods were often one-off and required new engineering for each domain and often for each use case scenario. For example, analyzing medical images required tailored techniques different from those used to detect faces in images.
 - Generative AI and multi-modal transformers shift the emphasis from manual feature engineering to prompt engineering. A single multi-modal model can be prompted with requests such as “Describe the key visual elements in this image”, and it will generate a summary of the most important parts of the image. The same approach can be adapted to different tasks and domains by changing prompts, and light fine-tuning methods such as LoRa (Low Rank Adaptation) can quickly adapt the model to domain-specific needs without retraining the entire model.
 - This universal approach has cut the cost of content analysis by about tenfold. Because the same core infrastructure can support many media types, teams can experiment faster: they can start with generic prompts and models to validate a use case and invest in fine-tuning only after the application proves valuable.
 - Prompts become the primary engineering artifact, which introduces new risks. Prompt execution is not strictly deterministic. The internal workings of large language models and multimodal transformers are not transparent, so one cannot fully explain how they generate outputs. Updates to underlying models can change behavior in subtle ways, affecting the applicability and stability of deployed solutions even when prompts do not change.

Traditional, mostly hand-crafted pipelines for feature extraction



General-purpose feature extractors via prompting and large language models



The main subject of this image is a majestic, snow-covered mountain peak, which appears to be the iconic Matterhorn in the Swiss Alps. The mountain dominates the frame, its distinctive pyramid shape rising dramatically against a clear blue sky. The setting is a high-altitude alpine environment, with the peak surrounded by other snow-capped mountains and glaciers visible in the lower portions of the image. The background is primarily composed of a vivid blue sky with a few wispy clouds. The colors in the image are striking, with the brilliant white of the snow contrasting sharply against the deep blue of the sky. The lighting appears to be natural sunlight, creating a play of light and shadow across the mountain's face that accentuates its rugged features and crevices. There are no visible people, animals, or man-made objects in the image. The focus is entirely on the natural grandeur of the mountain. The overall mood of the image is one of awe-inspiring beauty and serene majesty. There's a sense of isolation and pristine wilderness that the mountain embodies. Notable details include the jagged ridgelines of the mountain, the smooth snow fields on its flanks, and the wisps of cloud that cling to its lower slopes, suggesting high winds at the peak. The composition of the image is well-balanced, with the mountain placed slightly off-center, allowing the eye to follow its slopes from base to peak. The surrounding mountains and glaciers ...

- **Online query answering evolved rapidly in the past years**

- Early systems provided simple retrieval without relevance ranking, which was enough for file searches or basic catalog queries. As users asked for finer control and higher precision, systems added filtering and ranking, trading speed for accuracy. With the web's growth, large-scale retriever-ranker architectures became standard, offering high accuracy and low latency but higher infrastructure costs.
- Demand for question answering drove the next advance. Systems began combining retrieval with reading or generation to give direct answers instead of lists of documents. These hybrid models improved the user experience but required more computation and raised operating costs. For example, a web search for a factual question, such as "Who won the Formula 1 race this weekend", now returns a direct answer instead of a list of links that the user must click and read. Often, finding the answer on the linked pages took more time than the search itself.
- The field is now entering a period of rapid change after a long steady phase. Retrieval-augmented generation and agentic extensions combine planning, tool use, and iterative reasoning, improving accuracy and flexibility while challenging traditional cost and latency trade-offs. This faster research cycle reflects a wider range of use cases, from everyday search to complex multi-step problem solving, and points to rapid innovation in retrieval systems.
- Throughout this course, we will study models ranging from classical text retrieval approaches to modern methods that use agentic AI. In what follows, we describe the different retrieval types and explain what distinguishes them.

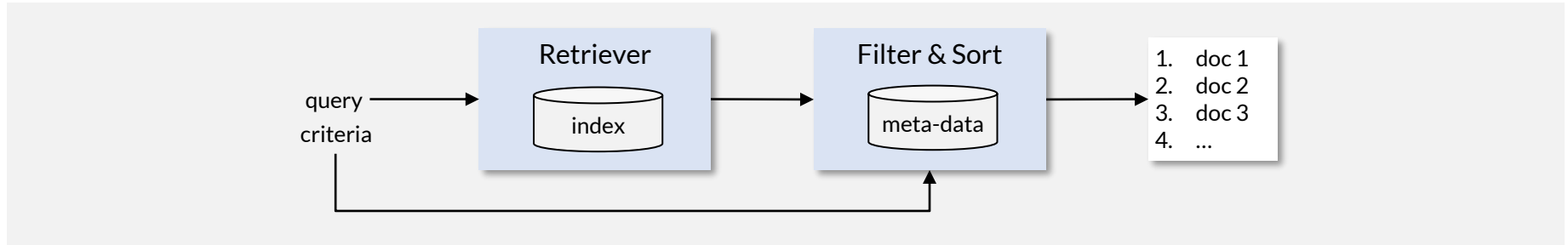
- **Retriever-only** systems use a retriever component to identify documents that match a query and present them to the user without an explicit relevance ranking. This basic search functionality is widely available in file search tools and simple web applications. Without ranking, filtering, and sorting, this approach works only for small data sets where queries usually narrow results to a few items which users can quickly assess for relevance.

→ Example: https://www.goodreads.com/search?q=agatha+christie&search_type=books



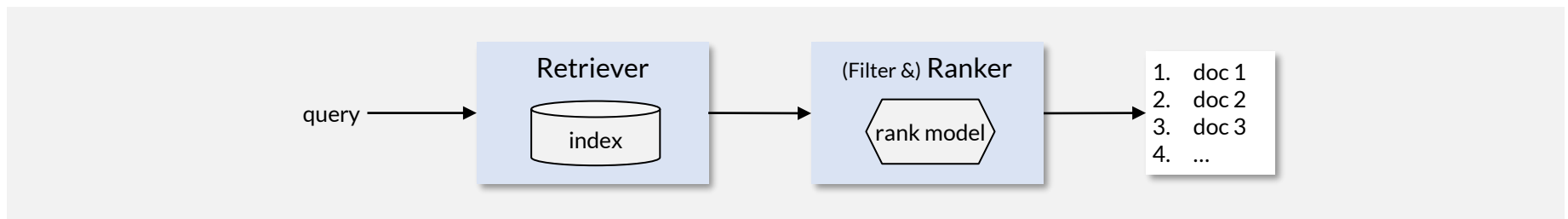
- **Retriever-Filter** systems are similar to retriever-only approaches but add a filtering and sorting stage to refine results before presentation. Filters let users narrow results by parameters such as year or rating, and sorting can be driven by attributes like popularity or price. Relevance may affect ordering, but other criteria often dominate. This search feature is common in e-commerce applications and is often enhanced with faceted search.

→ Example: <https://www.galaxus.ch/en/search?q=clothes+iron>



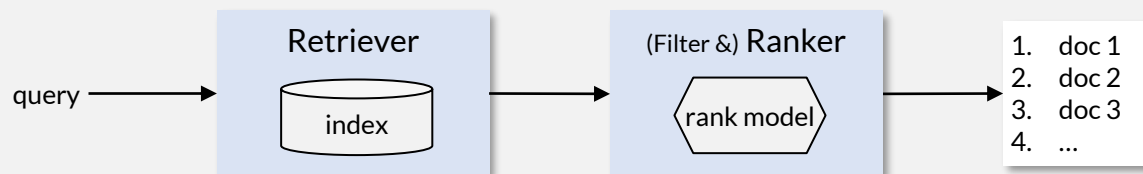
- **Retriever-Ranker** systems first select a pool of candidate documents using the retriever and then apply a ranker to assign a relevance score to each candidate, returning documents by score. This is a common architecture in both classical and modern retrieval systems and is frequently enhanced with semantic search and context-sensitive ranking such as user location, objective importance, and subjective importance. Web search engines typically use this model, combining text retrieval with web-specific ranking signals.

→ Example: <https://www.google.com/search?q=multimedia+retrieval+lecture> (change your location with a VPN client and submit again)



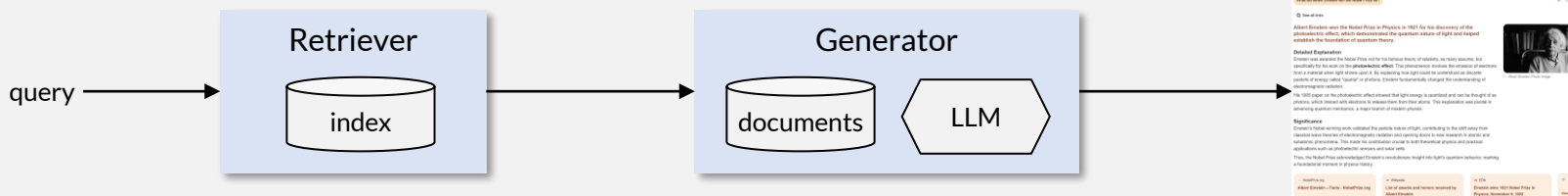
- **Retriever-Reader** systems are designed for question-style queries that ask for a specific answer. The retriever fetches relevant documents and the reader identifies one or more passages within those documents that answer the question, returning the passages rather than a list of documents. Readers often rely on language models to locate concise answers in the result documents.

→ **Example:** Google's 'featured snippet from the web' (this feature is now often replaced with a retriever-generator answer)
<https://www.google.com/search?q=what+is+the+main+ingredient+in+tylenol>
<https://www.google.com/search?q=What+did+Albert+Einstein+win+the+Nobel+Prize+for%3F>
 (this later query may provide an answer from the knowledge database; try "People also ask" for retriever-reader answers)



- **Retriever-Generator** systems, also known as Retrieval-Augmented Generation or RAG, combine the retriever with a generative language model. The retriever selects relevant documents or passages and those snippets are combined with the user query into a prompt template for a large language model. The model then generates a comprehensive answer rather than extracting a single passage.

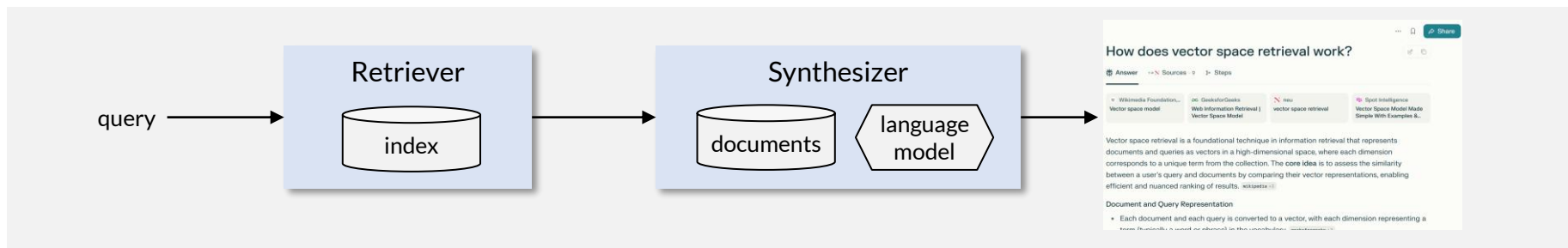
→ **Example:** Bing copilot search ("What did Albert Einstein win the Nobel Prize for?")
<https://www.bing.com/copilotsearch?q=What+did+Albert+Einstein+win+the+Nobel+Prize+for%3F>



- **Retriever-Synthesizer** systems fetch a set of relevant documents and then instruct a language model to synthesize a condensed summary rather than extracting a direct answer. This is especially useful for exploratory or conceptual queries, where users benefit from an overview of multiple sources rather than a single factoid. Unlike Retriever-Reader systems that target pinpoint answers, the summarizer model must integrate information, reconcile conflicting statements, and produce a coherent narrative.

→ **Example:** perplexity.ai with “How does vector space retrieval work?” (normal search)

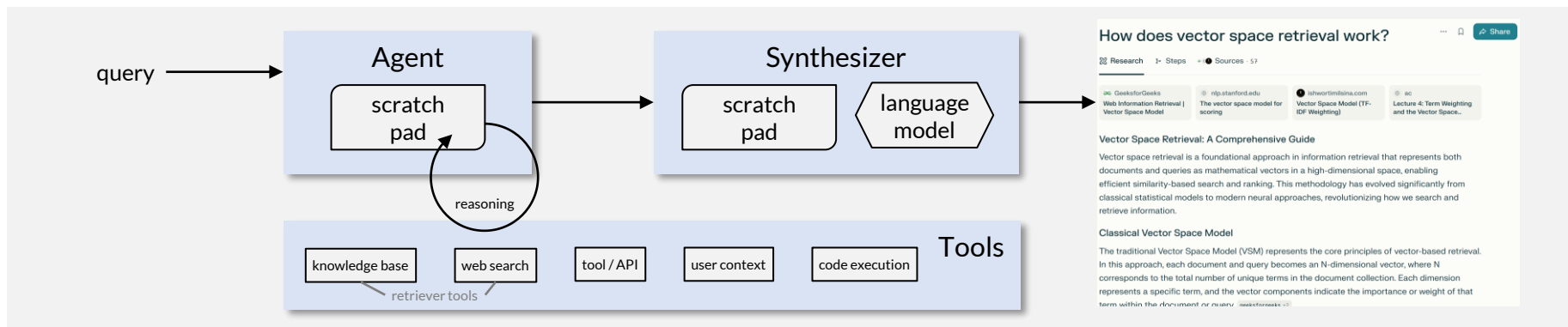
https://www.perplexity.ai/search/how-does-vector-space-retrieva-DuutsrURRgeeYmrJw_ObZg



- **Agentic RAG** systems extend the concept of Retriever-Generator and Retriever-Synthesizer with agentic capabilities. In this setup, the agent receives the query and actively decides how to fetch information, which sources to query, and whether to iterate or reformulate queries. It can plan a sequence of retrieval actions based on intermediate results, not just a single retrieval pass. Once the agent has gathered and processed the necessary information, a generative language model synthesizes a comprehensive final answer.

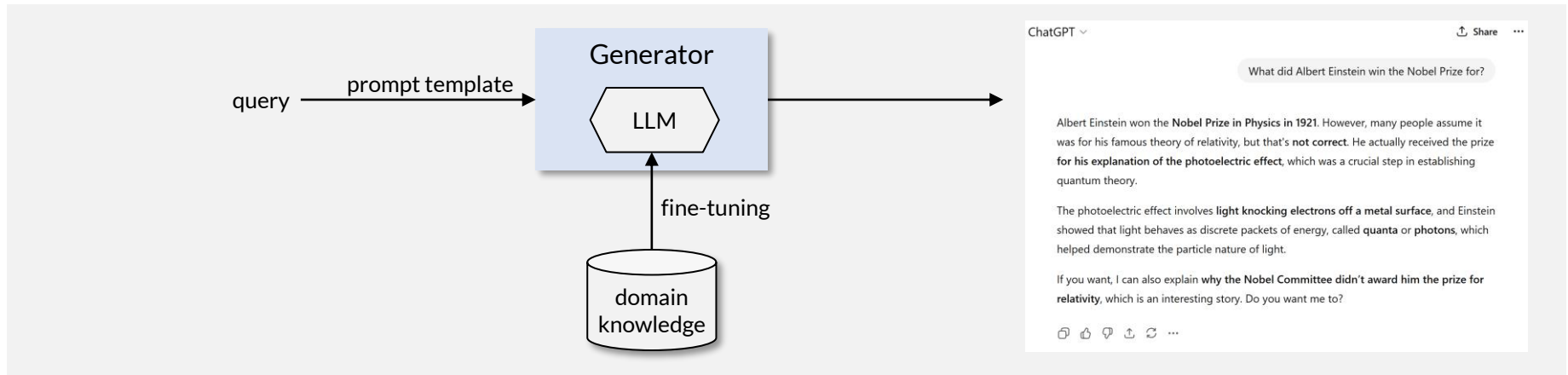
→ **Example:** perplexity.ai with “How does vector space retrieval work?” (pro version with multi-step reasoning)

<https://www.perplexity.ai/search/how-does-vector-space-retrieva-01jcYjFZQWmwzpaCVGdfRQ>



- **Generator-only** systems use only generative models and have no explicit retriever. They produce answers from knowledge stored in their training data. General-purpose models can handle many tasks with careful prompting, but fine-tuned models are needed for specialized or business queries. A major risk is hallucination, when the model gives plausible but incorrect information. This occurs because the model is trained to always answer and to satisfy the user, even when it lacks facts. Ambiguous questions, gaps in the training data, and the lack of retrieval grounding increase this risk. In short, generator-only systems are flexible but trade reliability for broad usefulness, making them vulnerable to mistakes in specialized or high-stakes situations.

→ **Example:** ChatGPT with “What did Albert Einstein win the Nobel Prize for?”
<https://chatgpt.com/share/68c57b5f-d174-8011-a6c1-14a0eb5078fa>



1.4 References & Links

- Statista is a reputable online portal for statistics, market research, and business intelligence. The volumes of data creates world wide, <https://www.statista.com/statistics/871513/worldwide-data-created/>
- The IBM Storage and Information Retrieval System (STAIRS), https://en.wikipedia.org/wiki/IBM_STAIRS
Also read the Computerworld article, 1975: https://books.google.com/books?id=X_3_D4RqzvIC&dq=IBM+STAIRS%2FVS&pg=PA14
- Semantic gap: the definition goes much beyond the scope of this introductory example, see [Wikipedia](#)
 - A. W. Smeulders, M. M. Worring, A. Gupta, and R. Jain, **Content-Based Image Retrieval at the End of the Early Years**, IEEE Trans. Pattern Anal. Machine Intell., vol.22 no.12, pp1349-1380, 2000. <https://doi.org/10.1109%2F34.895972>
 - B. Barz, J. Denzler, **Content-based Image Retrieval and the Semantic Gap in the Deep Learning Era**, CBIR workshop at ICPR 2020. <https://arxiv.org/abs/2011.06490>
- Google Research: Pathways Language Model (PaLM): **Scaling to 540 Billion Parameters for Breakthrough Performance** <https://ai.googleblog.com/2022/04/pathways-language-model-palm-scaling-to.html>
- Amazon AWS Machine Learning Blog, **Scaling Large Language Model (LLM) training with Amazon EC2 Trn1 UltraClusters** <https://aws.amazon.com/blogs/machine-learning/scaling-large-language-model-llm-training-with-amazon-ec2-trn1-ultraclusters/>
- Patrick Lewis et al., **Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks**. 2020, <https://doi.org/10.48550/arXiv.2005.11401>
- Epoch AI, **Publications**. <https://epochai.org/blog>