UNIVERSITÄT BASEL

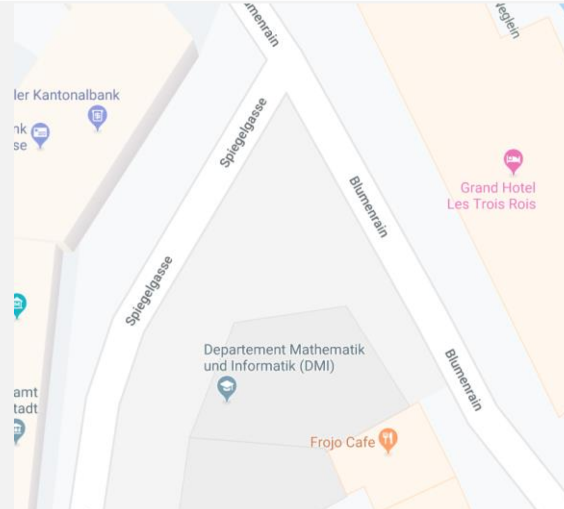
Computer Science / 15731-01 / 2022

Multimedia Retrieval

Chapter 0: Introduction

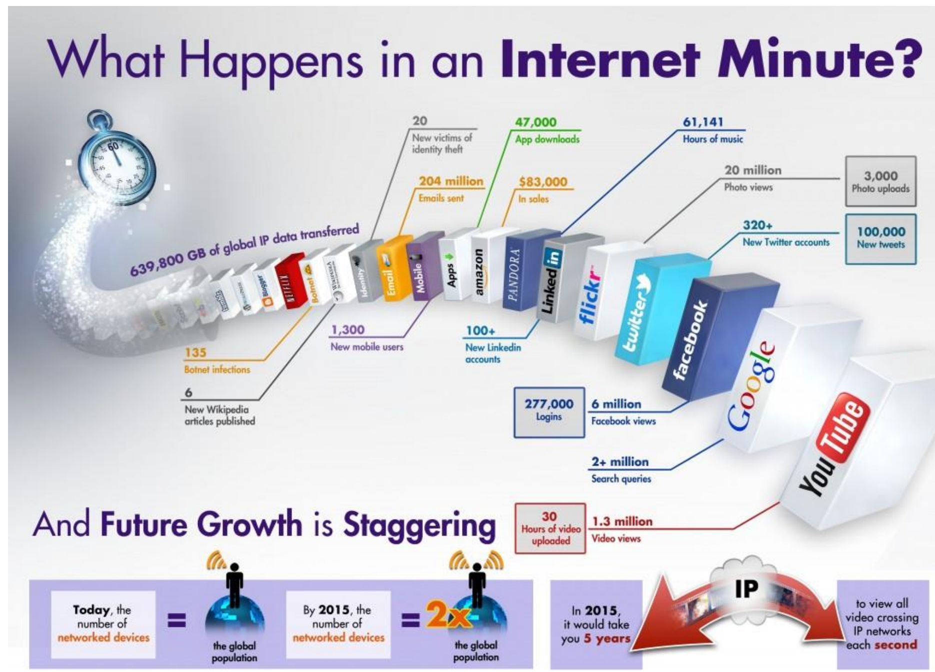
Dr. Roger Weber, roger.weber@gmail.com

Course ID	15731-01
Lecturer	Dr. Roger Weber, roger.weber@gmail.com
Time	Friday 15:15 - 18:00 Note: changes are announced on web site and / or per e-mail ahead of lectures
Location	<u>Physical presence:</u> Seminarraum 00.003, Spiegelgasse 1 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. https://unibas.zoom.us/j/66256788017?pwd=TDZmckxZb01QOHl0ZXRkOCtBcHd5Zz09
Prerequisites	Basics of programming 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 past and current information retrieval techniques and search algorithms.
Exam	Oral exam (30 minutes) on January 6, 13, 27 (15.00 – 18.00)
Credit Points	6
Grades	From 1 to 6 with 0.5 steps. 4.0 or higher required to pass exam.
Homepage	WEB: https://dmi.unibas.ch/de/studium/computer-science-informatik/lehrangebot-hs22/lecture-multimedia-retrieval/ ADAM: https://adam.unibas.ch/goto_adam_crs_1345357.html All materials are published in advance. Practical exercises to be submitted to ADAM



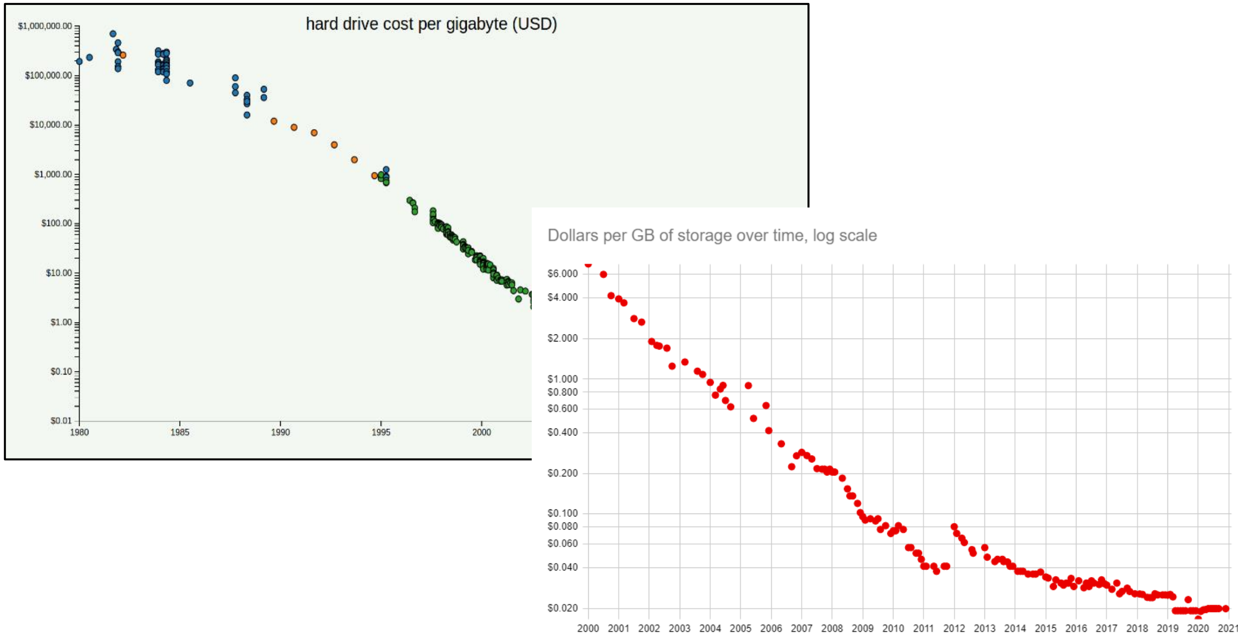
0.1 Motivation: Why Information Retrieval?

- We are living in a digital world with exponential information growth. Every day, we consume and produce enormous amounts of data:
  - ~400 Exabyte of IP traffic per month (Cisco prediction for 2022)
  - ~100 Zettabyte of data, doubling every 18-24 months (Statista, 2021)
  - ~100,000 Google queries per second (8.5bn searches per day, Internet Live Stats, 2022)
  - ~2bn websites on the Internet (Statista, 2021)
- This enormous growth was possible due to continuous improvements of how we store, transport, and manipulate data. In 2017, the cost per media was as follows:
  - Tape Drive: $6*/15 per TB (* compressed)
  - Hard Drive: $25 per TB
  - SSD Drive: $250 per TB
- Compared with past prices:
  - 1990: $10M per TB
  - 2000: $10K per TB
  - 2010: $100 per TB
- So 10 Zettabyte ($10 \cdot 10^{21}$) roughly costs $250B which is about 0.25% of the world wide gross domestic product, or 10 times less than the world wide spend on oil. In other words, data is cheap!
- **What can we do with so much information and how can we find "stuff"?**



source: Intel

- Illustration of price and space compression of storage



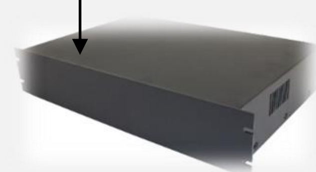
In the past decades, we had price drops of 50% every 14 months (!). Every 4 years, the costs decreased by an order of magnitude. However, we see that firms still spend the same amount of $ each year to increase and replace their storage real estate. As a consequence, the amount of managed storage grows exponentially and imposes ever larger problems to find relevant information. A financial institute, as an example, is now required to answer regulatory inquiries (court calls) over >100PB of online and offline data (GDPR)

42U Rack

1 Petabyte of storage (2015)

21x smaller in 3 years

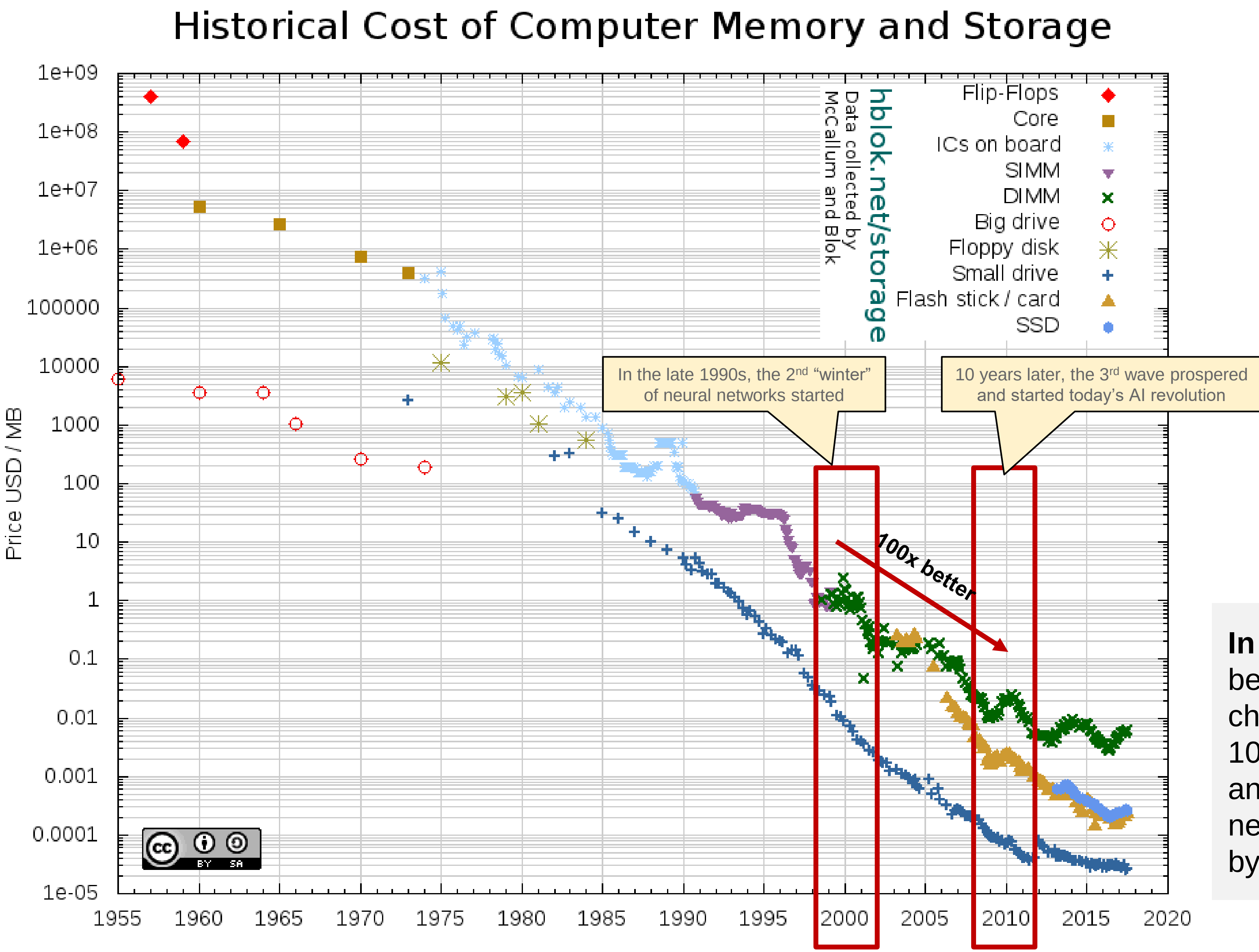
72x 16TB



2U Appliance

1 Petabyte of storage (2018)

- Another view on the price developments for memory and storage:



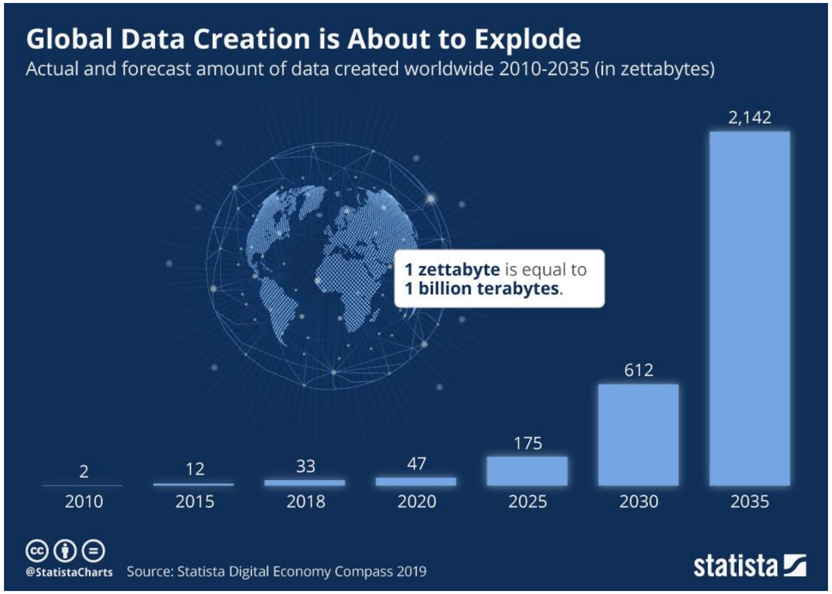
In 10 years: memory became a 100 times cheaper, disk got a 100 times cheaper and compute for neural nets improved by a 1000 times

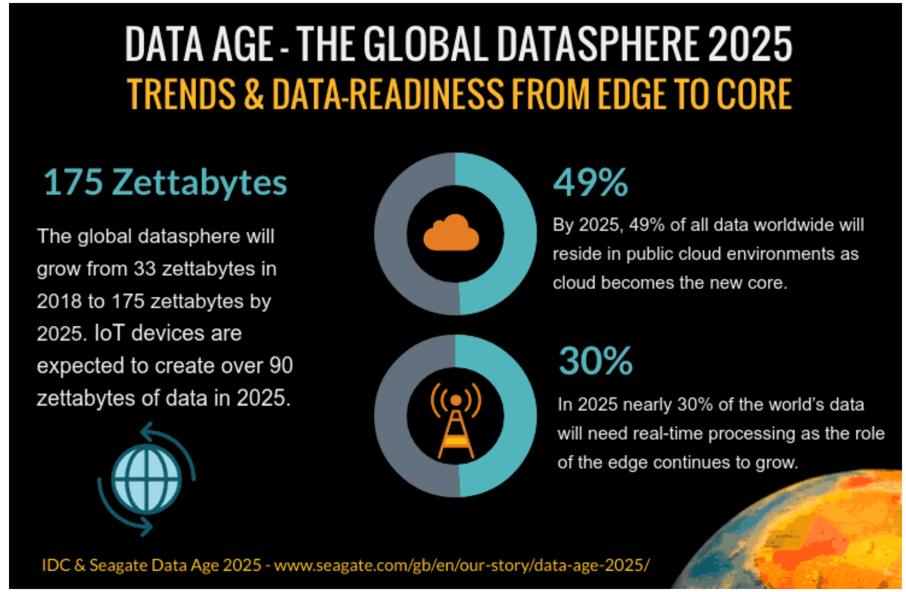
source: https://hblok.net/blog/storage/

- What can we do with so much information? A few examples:
  - **Byte [B]:**
    - 1 B = a single ASCII character (Unicode: 2 bytes)
    - 4 B = single precision floating point (32 bits)
    - 8 B = double precision floating point (64 bits)
  - **Kilobyte [10^3 B]:** (note: the IEC defined kibibyte as the binary equivalent, i.e., 1024B)
    - 10 KB = a page in a encyclopedia; size of an average web page
    - 64 KB = L1 cache size per core in CPU
  - **Megabyte [10^6 B]:**
    - 1 MB = a novel
    - 1-8 MB = a digital image
    - 5 MB = all written pieces by Shakespeare; a typical MP3 file
    - 100 MB = 1 meter of books in a shelf
  - **Gigabyte [10^9 B]:**
    - 1 GB = a small transporter full of paper; a movie in TV quality (not HD)
    - 2 GB = 20 meters of books in a shelf
    - 20 GB = all pieces by Beethoven stored as audio files; a movie in HD quality
    - 500 GB = largest FTP server
  - **Terabyte [10^12 B]:**
    - 1 TB = all x-rays in a large hospital; data produced by the EOS system in a single day
    - 2 TB = all books in the Uni Basel library; all emails exchanged in a day
    - 60 TB = largest SSD in 2016 (Seagate, 3.5" form factor; $10,000)
    - 300 TB = data released by CERN in April 2016 for latest run of the large Hadron Collider
    - 400 TB = database of the "National Climatic Data Center (NOAA)"

- **Petabyte [10^15 B]:**
  - 1 PB = data produced by the EOS system over the last year
  - 2 PB = DVD material of all movies listed in IMDB; all CDs registered in CDDB
  - 10 PB = data per hour from Square Kilometer Array (SKA) telescope (2016)
  - 20 PB = Storage owned by large banks (2009, annual growth rate at 30%)
  - 100 PB = size of Google's index (2016, 60 trillion pages, $60 \cdot 10^{12}$)
  - 200 PB = all pieces of information ever printed worldwide
- **Exabyte [10^18 B]:**
  - 1 EB = monthly traffic in Internet in 2004
  - 5 EB = all words ever spoken by a human being (stored as text)
  - 16 EB = address space of a 64-bit processor
  - 500 EB = digital information in Internet as of 2009 (2007: 281 EB)
  - 667 EB = annual global IP traffic estimated for 2013 (>1000EB in 2016)
- **Zettabyte [10^21 B]**
  - 1 ZB = 12,288 million 4K videos
  - 4.8 ZB = global data traffic via the Internet (~80% for video content; Cisco prediction for 2022)
  - 97 ZB = total amount of data created, captured, copied, and consumed globally (Statista 2022)
- **Yottabyte [10^24 B = 1,000,000,000,000,000,000,000,000 B]**
  - 1 YB requires 412'500m^3 of 400GB microSDXC cards (~165 Olympic size pools)
  - Expected amount of data generated in 2030 (Huawai, 2021)
- **Brontobyte [10^27 B = 1,000,000,000,000,000,000,000,000,000 B]**
  - Place your bets: when do we reach this next milestone?

- We are drowning in what has become the Big Data Lake (or Swamp)
  - And to make things worse, the lake is growing faster and faster with no end in sight.
  - How long would it take to read 1 Petabyte of information? how long for 1 Zettabyte?
  - How can we process or search through such enormous amounts of data?

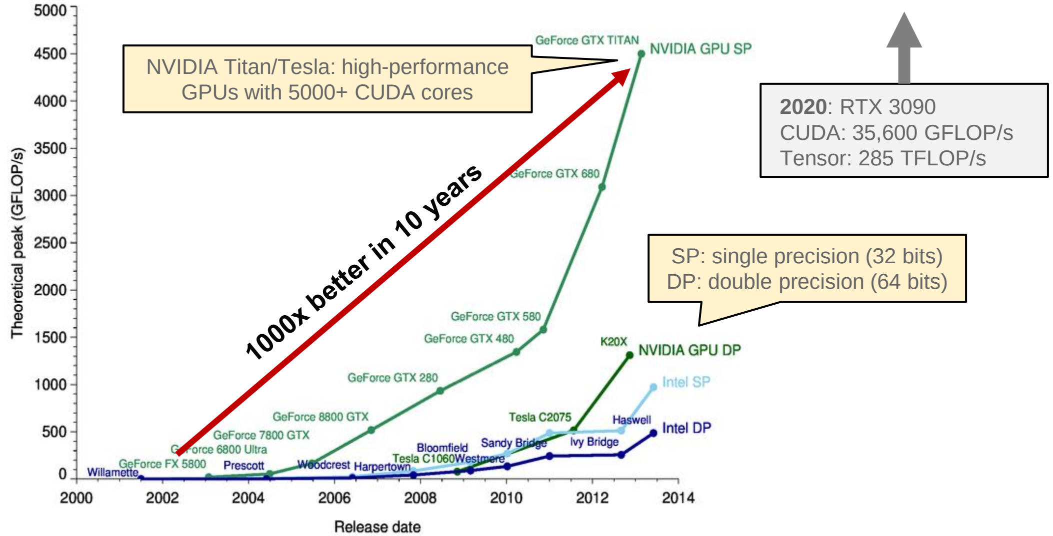


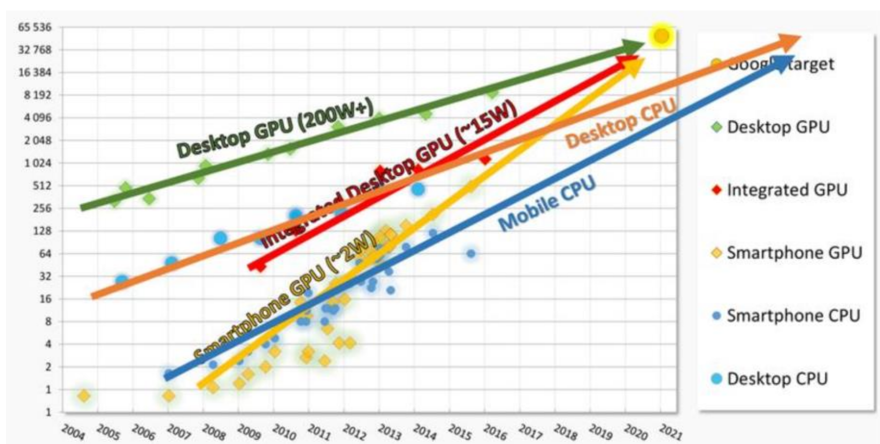


The Internet is growing at a rate of 14% a year. Every 5.32 years, the number of domains doubles (see figures above). Google's index contains more than 35 billion pages. In 2008, Google software engineers announced that they discovered one trillion unique URLs.

In the age of Big Data, growth rates have further accelerated! Social media surpassed enterprise data and VoIP data. Sensors & devices (Internet of Things) have doubled the generated data volumes. In 2022, an estimated data volume of 100 ZB (!) is created, copied and consumed in a single year.

- The advances in recent years also improved our ability to organize and analyze data. For example, deep learning still operates with similar paradigms than 20 years ago. But with the performance boosts, mostly through GPUs and TPUs, it is now possible to train very large networks with brute-force methods. Similarly, such brute-force methods (aka Big Data) have drastically improved response times in data processing to extract meaningful facts from ever growing “data lakes”.





The biggest improvement over the past ten years was the creation of CUDA, an extreme parallel computing platform created by Nvidia. In combination with new neural network algorithms and the advent of map/reduce as a generic distributed computing paradigm, enormous amounts of data became processable through the sheer brute force of 1000s of connected machines. Going forward, we will see highly specialized chips (like Google’s TPUs) and cloud compute hardware (like HPEs ‘The Machine’) further accelerating the hunt in ever larger data lakes.

Fun fact: the next gen game consoles have more than 10,000 GFLOP/s

- So, how long does it take to read 1 Petabyte? All data points as of 2017:
  - The fastest hard disk have about 200MB/s read rate (and almost same write rate)
  - The fastest solid state disk have about 550MB/s read rate (and 10% smaller write rate)
  - The fastest M.2 flash drives have about 3500MB/s read rate (and 2100MB/s write rate)
  - USB 3.0 can handle up to 640MB/s transfer rate
  - PCI-E 2.0 can handle up to 4GB/s transfer rate
  - 100GB Ethernet can handle up to 10GB/s transfer rate
  - Fibre full duplex 128GFC can handle up to 13GB/s (per direction)
  - The largest Internet exchange point (DE-CIX) operates at up to 700GB/s (average is 430GB/s)

Device / Channel	GB/s
Hard Disk	0.2
Solid State Disk	0.55
M.2 Flash Drive	3.5
USB 3.0	0.64
PCI-E 2.0	4
Ethernet 100GB	10
Fibre 128GFC	13
DE-CIX	700

Time to read

1 GB	1 TB	1 PB	1 EB	1 ZB
5s	83m	58d	158y	158’000y
1.8s	30m	21d	58y	57’000y
0.3s	5m	80h	9y	9’000y
1.6s	26m	18d	50y	50’000y
0.3s	4m	70h	8y	7’900y
0.1s	100s	28h	3y	3’200y
0.08s	77s	22h	2.5y	2’400y
1.4ms	1.4s	24m	16d	45y

- We can’t beat physics…but we can apply brute force with extreme scale and parallelism. A million machines can help to shorten times if the algorithm does scale. Today, large compute clouds have between 1-5 million servers.

0.2 The Retrieval Problem

Given

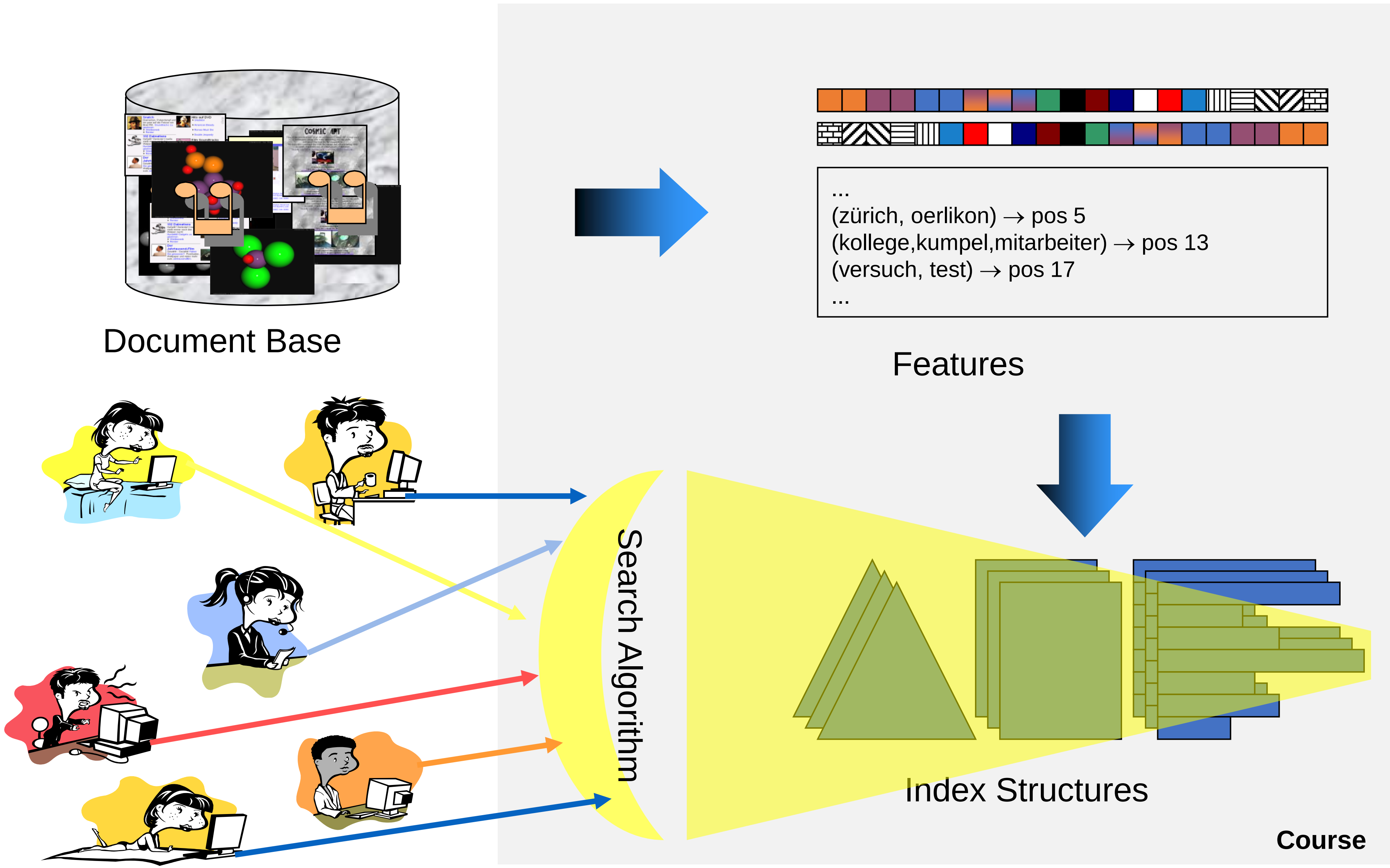
- N documents (D_0, ..., D_{N-1})
- Query Q of user

Problem

- Ranked list of k documents D_j ($0<j<N$) which match the query sufficiently well; ranking with respect to relevance of document to the query

- How do we measure ‘relevancy’ from a user’s perspective? Or: what am I actually looking for:
  - Extreme case 1) Find all relevant pieces of information
    - For instance: to avoid a patent clash, all relevant documents have to be retrieved. If a relevant one is missed, high costs may result due to law suits
  - Extreme case 2) Find quickly an answer to an information need (first result is enough)
    - For instance: when does the course “Multimedia Retrieval” at University Basel start
- How do we compare two retrieval systems based on ‘relevancy’ definition of the user and their returned results. How can we objectively determine whether system A (algorithm A) is working better than System B (algorithm B)
- How can we accelerate the search, i.e., avoid reading through petabytes of documents to find the results to a query? We will introduce different ways of extracting ‘meaning’ from documents (also called features), thereby compressing the amounts of data. In addition, we organize storage in such ways to process queries much faster than a ‘brute force’ scan through documents

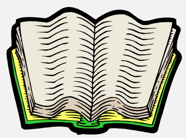
The principle ingredients of information retrieval



0.2.1 Content Modeling and Feature Extraction

- Due to the enormous amount of data stored in a collection, indexing of documents must be implemented in a fully automated way (no human interaction). But how can we describe the content of a document in a denser way to avoid reading through terabytes of data? Depending on the document type, different approaches exist:

Text Documents

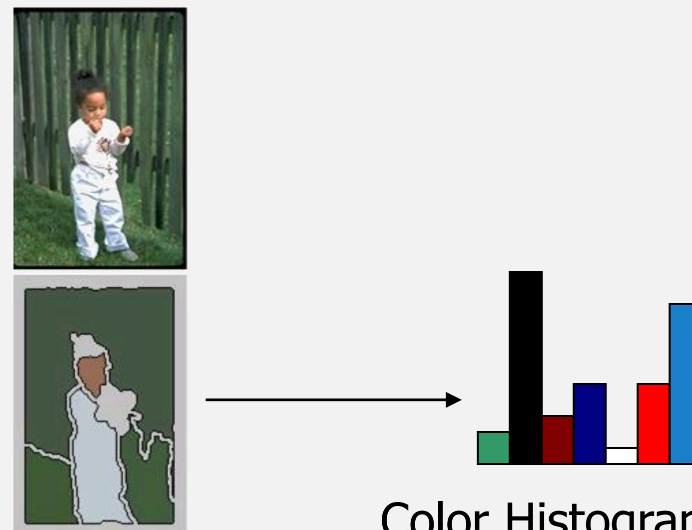


c

Feature Extraction

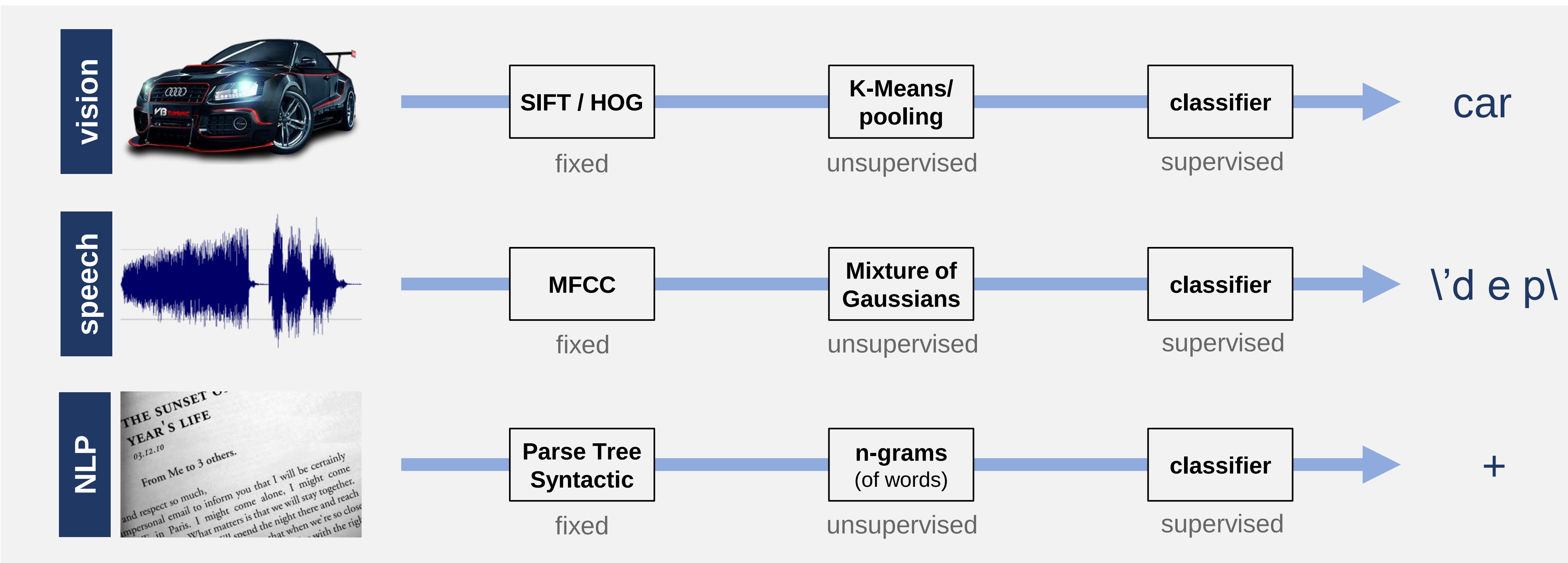
docID = doc10
dog → word 10, word 25
cat → word 13
home → word 2, word 27
...

Images



Color Histogram

- Higher level features (object recognition, motion detection, genres, moods,…)
  - Signal information is too low level and too noisy to allow for accurate recognition of higher-level features such as objects, genres, moods, or names. As an example, there are exceedingly many ways how a chair can be depicted in an image based on raw pixel information. Learning all combinations of pixels or pixel distributions is not a reasonable approach (also consider clipped chairs due to other objects in front of them).
  - Feature extraction based on machine learning abstracts lower level signal information in a series of transformations and learning steps as depicted below. The key ingredient of a learning approach is to eliminate noise, scale, and distortion through robust intermediate features and then cascade one or many learning algorithms to obtain higher and higher levels of abstractions.



0.2.2 Search Paradigms and Retrieval Models

- In the course, we focus on the following basic query patterns:
  - **Keyword-based Search:** The most widely used approach is to enter a few keywords and to browse through result lists to find relevant documents. This especially works fine for text documents (or spoken text in audio files); all other media types, however, suffer from the so-called semantic gap between keywords and signal information (unless there is meta data)
  - **Similarity Search:** Instead of entering keywords, the user provides examples how the result should look like (“query by example”). The search engine looks for documents that best matches these examples. Similarity search works with text, images, audio files, and video files. The definition of what “similarity” actually means depend on the chosen models and features
  - **Combined Search:** Merges capabilities of attribute based queries (e.g., predicates), keywords, and examples to match. While the atomic parts follow the patterns, additional models and algorithms define an appropriate overall retrieval order and an evaluation algorithm.

0.2.3 Index Structures

- Feature extraction algorithms compress the contents of a document to a few vectors and values. These features have to be indexed such that relevant document for given queries can be efficiently retrieved. Over the last decades, numerous index structures were proposed for the problem of “information retrieval”. Often, an index structure heavily relies on a specific document type or query paradigm. The following provides a short overview of the index structures addressed in this course:
  - **Inverted Lists**: usually applied to text documents with a large dictionary (words used in the documents). The basic observation is that a document only uses a few words out of the dictionary. The course will also present a low cost implementation of inverted lists based on relational databases.
  - **High-Dimensional Index Structures:** many feature extraction algorithm compute high-dimensional vectors (e.g., LSI, color, texture, Fourier coefficients). Such vectors are usually maintained in special index structures (sequential file, VA-File, X-Tree) also supporting similarity searches.
  - **Brute Force:** given the hardware improvements of the past, brute force methods (search everything) have become more attractive for some of the most challenging task. Most Big Data algorithms and implementations work with map/reduce and distribute the workload of 100s or even 1000s of machines.

0.2.4 Ranking of Results

- Queries against large collections typically yield endless results. A web query often leads to more than 1 million documents. How can we rank these documents such that the most relevant ones appear at the top of the list?
- Ranking criteria depend on the document and feature type. A few examples about techniques presented in this course:
  - **Text Retrieval:** The so-called „retrieval status value“ (RSV) defines a metric for how well a document matches the query. Documents are ranked by decreasing RSV.
  - **Web Retrieval:** Web queries often contain only one or two words. This leads to poor ordering by RSV. Instead, search engines consider the proximity of words in the documents, the appearances in the document, attributes associated with terms, and the objective importance of a web page (PageRank by Google) or other linking information (authority, hubs).
  - **Image Retrieval:** a distance function in a (high-dimensional) feature space defines a measure for dissimilarity between two documents. The larger the distance the higher the lower the similarity and thus the rank of the image in the result list.
  - **Multimedia Retrieval** (e.g. Video, or Image & Text): due to the different underlying comparison and ranking functions (Image, Text, Audio), there is a need for special algorithms and combining functions to merge partial result lists to an overall result list.

0.3 Course Schedule

Chapter 0: Introduction   (1 hour)

Chapter 1: Performance Evaluation   (3 hours)

- Boolean Retrieval
- Retrieval with Ordering of Documents
- Machine Learning Basics
- Performance of Models

Chapter 2: Classical Text Retrieval   (3 hours)

- Term Extraction
- Models for Text Retrieval
- Index Structures

Chapter 3: Advanced Text Retrieval   (3 hours)

- Advanced Techniques for Text Processing
- Ordering Web Pages with the Example of Google
- Considering the Context of a Page (Hubs & Authorities, PageRank)

Chapter 4: Image Retrieval (6 hours)

- Basic understanding of perception
- Feature Extraction and Relevance Evaluation
- Advanced Methods for Image Recognition

Chapter 5: Audio Retrieval (4 hours)

- Basic understanding of perception
- Feature Extraction and Relevance Evaluation
- Advanced Methods for Audio Recognition

Chapter 6: Video Retrieval (2 hours)

- Keyframes and Shot Detection
- Motion Detection

Chapter 7: Similarity Search (2 hours)

- Index Structures for Similarity Search
- Evaluation of Complex Queries
- Relevance Feedback

Date	Time: 15.15 / 16.15	Time: 17.05	Where*
Sep 23	0. Introduction, 1. Evaluation	Q&A	University*
Sep 30	1. Evaluation	Ex 1, Q&A	University*
Oct 7	2. Classical Text	Q&A	tbd
Oct 14	2. Classical Text, 3. Advanced Text	Ex 2, Q&A	tbd
Oct 21	3. Advanced Text	Q&A	tbd
Oct 28	3. Advanced Text, 4. Image Retrieval	Ex 3, Q&A	tbd
Nov 4	4. Image Retrieval	Prep Exam	tbd
Nov 11	4. Image Retrieval	Ex 4, Q&A	tbd
Nov 18	4. Image Retrieval	Ex 4, Q&A	tbd
Nov 25	*No Lessons (Dies Academicus)*		
Dec 2	5. Audio Retrieval	Ex 5, Q&A	University*
Dec 9	5. Audio Retrieval	Ex 5, Q&A	University*
Dec 16	6. Video Retrieval	Q&A	University*
Dec 23	7. Similarity Search	Eval & Prep Exam	University*

* University: Spiegelgasse 1, Seminarraum 00.003, **no zoom available**, **no video uploads** after lecture
* Zoom: see meeting link on Web / in ADAM, video uploads after lecture

See course web site & ADAM for latest update of schedule