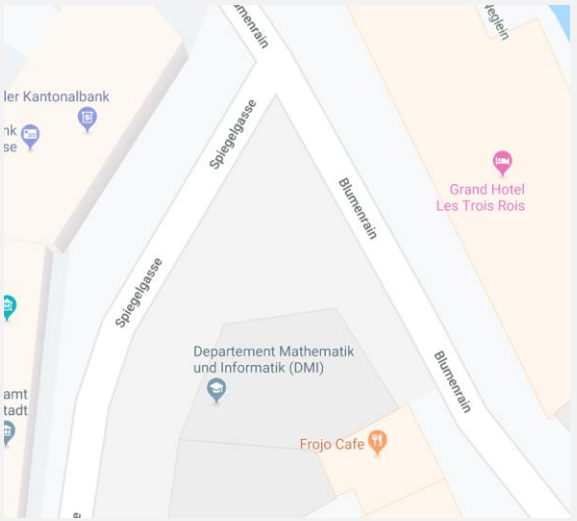
Multimedia Retrieval

Chapter 0: Introduction

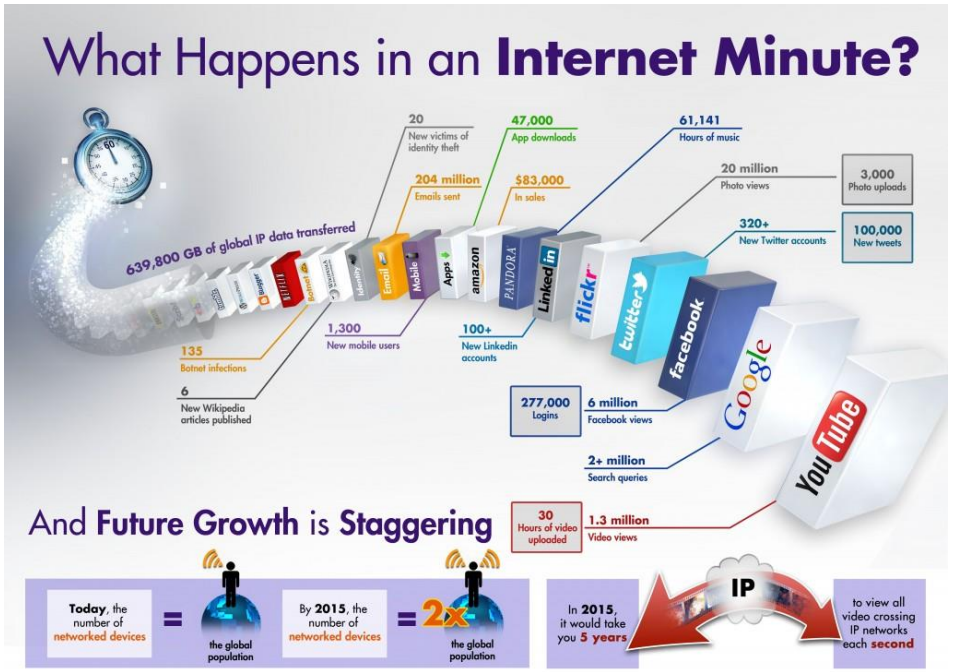
Dr. Roger Weber, roger.weber@ubs.com

Course ID	15731-01
Lecturer	Dr. Roger Weber
Time	Wed 14:15 - 17:00 Note: changes are announced on web site and / or per e-mail ahead of lectures
Location	Spiegelgasse 1, room 00.003
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 Jan 08 & 14, 2020
Credit Points	6
Grades	From 1 to 6 with 0.5 steps. 4.0 or higher required to pass exam.
Modules	Doktorat Informatik: Empfehlungen (PF Informatik) Modul Applications of Distributed Systems (Master Computer Science 16) Modul Applications of Machine Intelligence (Master Computer Science 16) Modul Concepts of Distributed Systems (MSF - Computer Science)
Homepage	https://dmi.unibas.ch/de/studium/computer-science-informatik/lehrangebot-hs19/lecture-multimedia-retrieval/ All materials are published in advance. Practical exercises submitted to “courses” site



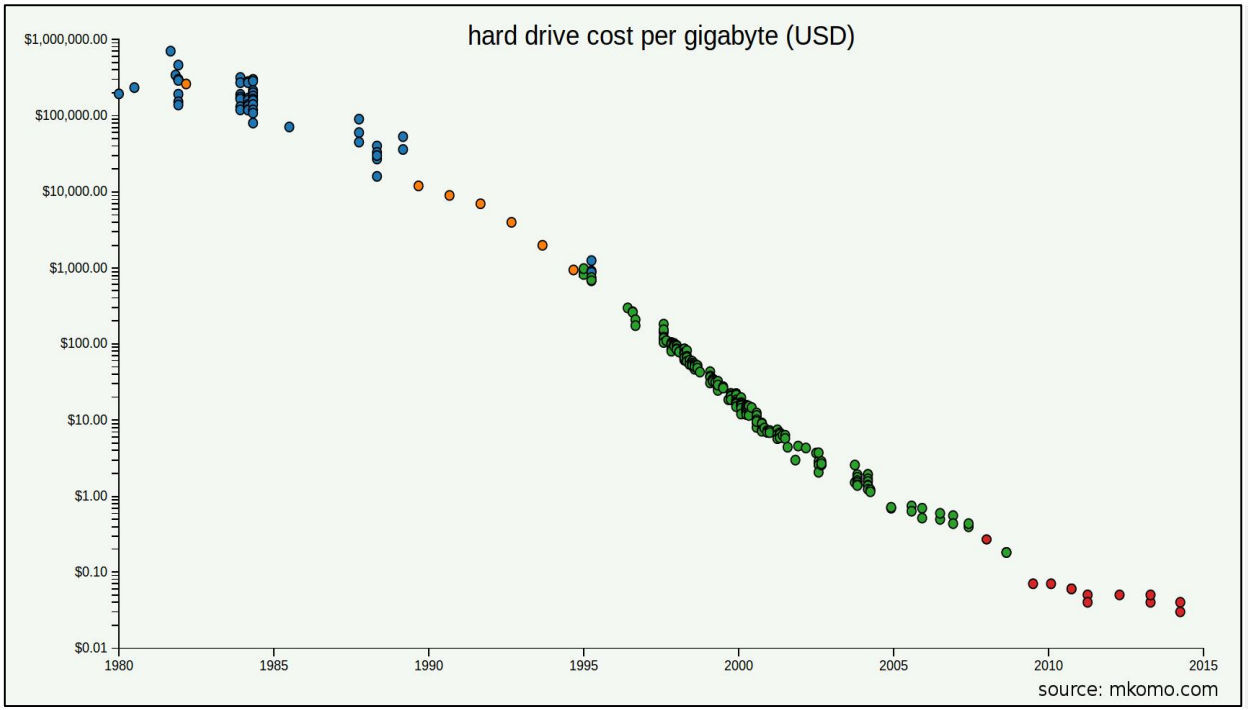
0.1 Motivation: Why Information Retrieval?

- We are living in a digital world with exponential information growth. Every day, we consume and produce enormous amounts of data:
  - 122 Exabyte of IP traffic per month
  - 10 Zettabyte of data, doubling every 18-24 months
  - 40,000 Google queries per second
  - 1 billion websites on the Internet
- This enormous growth was possible due to extreme improvements to store, transport, and manipulate data. In 2017, the costs per media were 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!
- **But 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.

42U Rack

1 Petabyte of storage (2015)

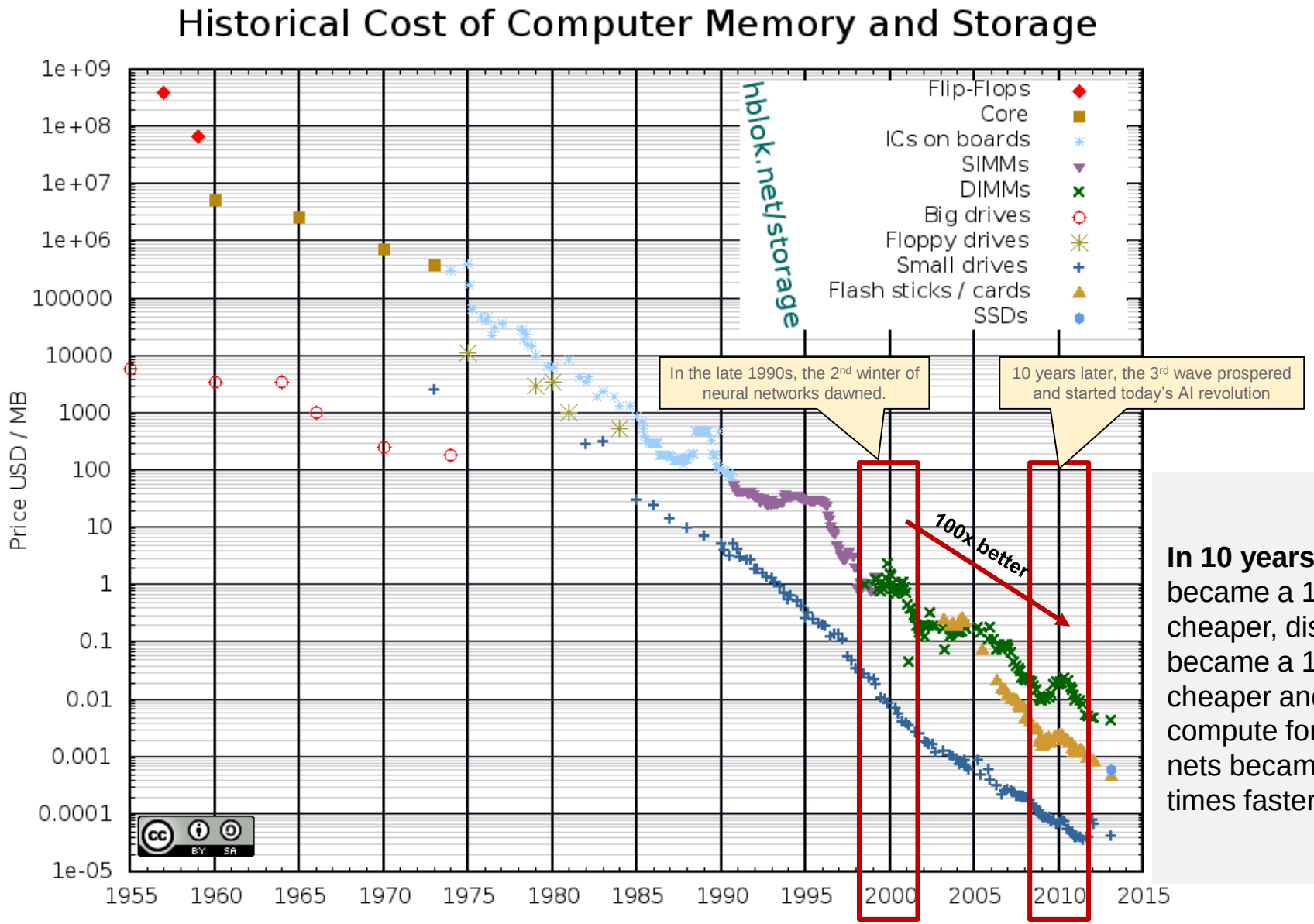
21x smaller in 3 years

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 became a 100 times cheaper and compute for neural nets became 1000 times faster

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
  - 3 EB = digital information in CH in 2007 (equals 580 tons of books per person)
  - 5 EB = all words ever spoken by a human being (stored as text)
  - 10 EB = estimated size of disk storage @ Google (R. Munroe, 2013)
  - 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 = 36,000 years of high-definition video
  - 1.5 ZB = Worldwide data center storage capacity in 2018 (estimates by Statista)
  - 10 ZB = data generated worldwide in 2017
  - 42 ZB = all words ever spoken by a human being (16 kHz 16-bit audio)
- **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)

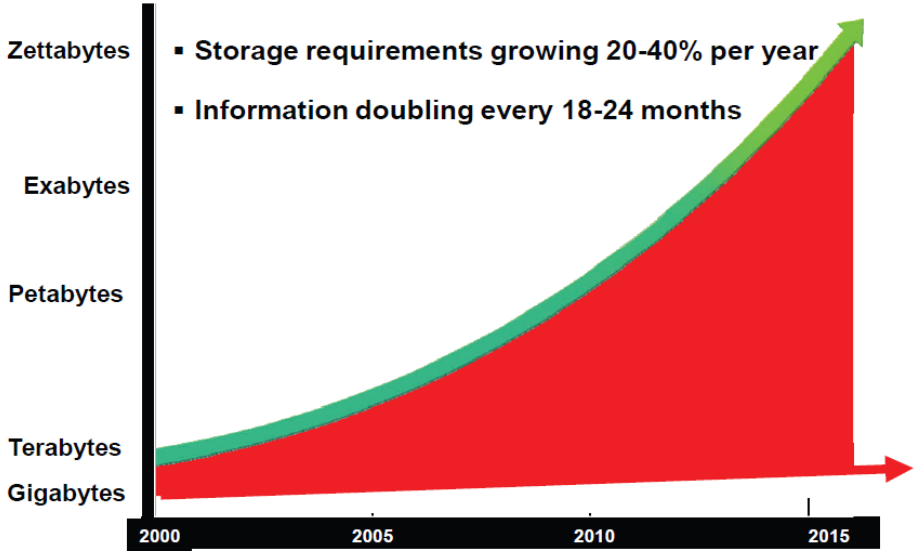
Over the past years, due to

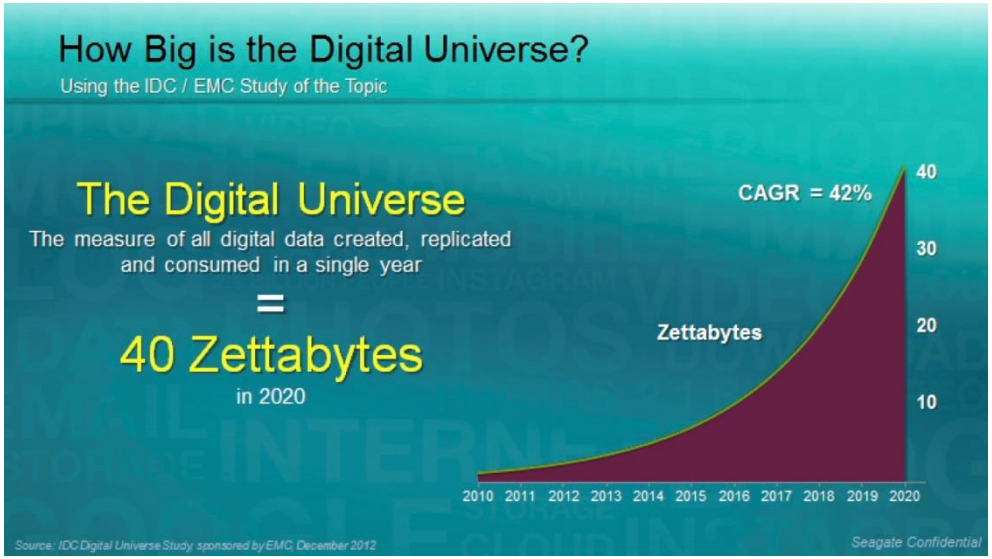
- enormously improved computational power at a constant price level,
- significantly smaller costs per memory unit,
- digitalization, ubiquitous media capturing, HD videos, and
- new and improved sensors (scanners, satellites, recorders)

extreme amounts of digital documents were created and archived. Only a small fraction of these archives are currently economically operated, i.e., in most cases documents are just archived but seldom organized in a way to generate profit. A few examples:

- **Internet** contains around 500 EB of data but only a fraction is easy to find
- **Companies** operate numerous information systems but there rarely exists a consistent access across the diverse systems; duplication of data makes things even harder; regulatory requirements ask for archival of all (relevant) information for 5-10 years
- **Media** archives encompass large amounts of photos, articles, videos but mostly writers browse through the archives "by hand"; upload features like on youtube inflate repositories with new material. Often, videos become "viral" way after they have been uploaded
- **Internet of Things** produces terabytes of data per day through numerous sensors (cameras, microphones, smart devices, satellites); most of the data is directly moved into an archive without being considered by a human being beforehand ("data graveyards")
- **Entertainment** companies are selling numerous movies and music songs but do we actually find what we are looking for? Do I find the music/movie that I like in iTunes?
- **People** store lots of data in various devices (computers, mobile phones, at work) and various formats (PDF, PowerPoint, MP3, …). Are you always finding your documents? Some own more than 50'000 still images taken with HD digital cameras (500+ GB of raw data). Storage became so cheap that purging old files is no longer required.

- 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.

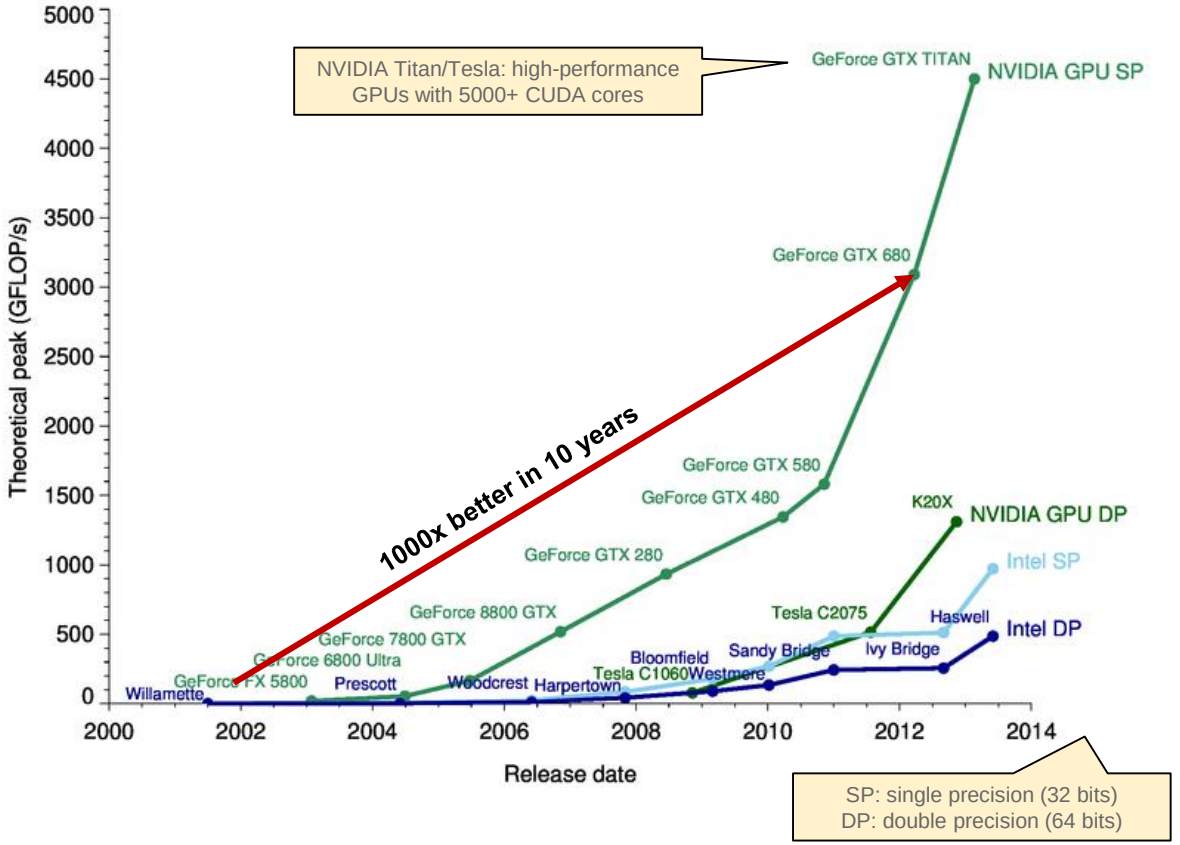
In the age of Big Data, growth rates have further accelerated! Social media surpassed enterprise data and VoIP data. Sensors & devices (Internet of Things) have doubled the generated data volumes. In 2020, an estimated data volume of 40 ZB (!) is created, replicated and consumed in a single year.

- 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 7[illegible]B/s (average is 430GB/s)

Device / Channel	GB/s	Time to read	1 GB	1 TB	1 PB	1 EB	1 ZB
Hard Disk	0.2		5s	83m	58d	158y	158'000y
Solid State Disk	0.55		1.8s	30m	21d	58y	57'000y
M.2 Flash Drive	3.5		0.3s	5m	80h	9y	9'000y
USB 3.0	0.64		1.6s	26m	18d	50y	50'000y
PCI-E 2.0	4		0.3s	4m	70h	8y	7'900y
Ethernet 100GB	10		0.1s	100s	28h	3y	3'200y
Fibre 128GFC	13		0.08s	77s	22h	2.5y	2'400y
DE-CIX	700		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 help you to shorten times if the algorithm does scale nicely (almost like moving two cells to the left in above table). Large clouds have between 1-5 million servers these days.

- The advances in storage, networking, and compute not only increased the problem size, equally, they have improved our ability to organize the data and reason about its content. For instance, deep learning which is the basis for the recent advances in artificial intelligence, still operates with similar paradigms as 20 years ago when the research field got totally abandoned. With the extreme performance boosts, mostly through the use of GPUs and now through special chips, it has become possible to train extremely large networks with a brute-force method. Similarly in other areas where it was historically difficult to scale, brute-force methods (aka Big Data) have drastically improved response times and were able to deduce meaningful data out of the enormous "data lakes".



The biggest improvement over the past ten years was the creation of CUDA, a 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.

0.2 The Retrieval Problem

Given

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

Problem

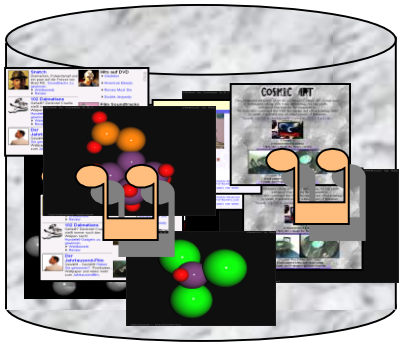
- Ranked list of k documents D_j (0<j<N) which match the query sufficiently well; ranking with respect to relevance of document to the query

Topics of this course

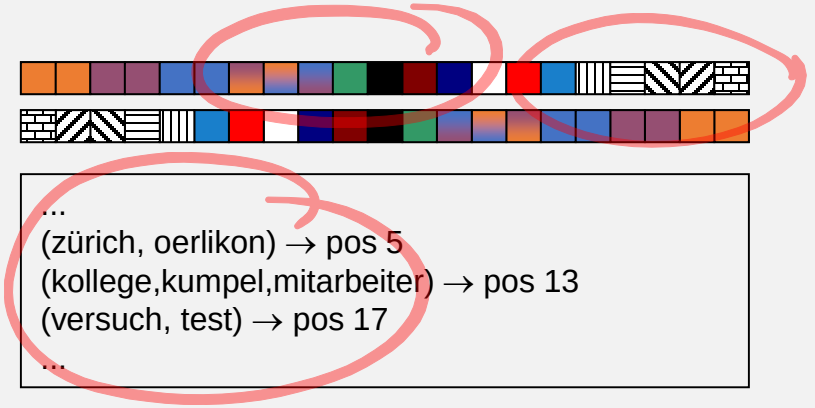
- Low level feature extraction (words, n-gram, color, pitch, tempo)
- Machine learning to obtain higher level features from content
- Retrieval models (Boolean retrieval, vector space retrieval, LSI, signatures, probabilistic retrieval, nearest neighbor search, meta data search)
- Index structures (inverted list, signature files, relational database, multi-dimensional index structures)
- Ranking of retrieved documents (RSV, link structure, phrases)
- Web retrieval and information hidden in a network
- Performance measurements (benchmarking)

- What are the actual goals of the user?
  - 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 due to law suits may result
  - Find quickly an answer to an information need
    - For instance: a student wants to retrieve documents on “information retrieval”. A good article or book is sufficient; she does not care if a relevant one is missing
  - This also leads to the important question of how to compare two retrieval systems based on the relevance of their results. How can one objectively determine whether system A (algorithm A) is working better than System B (algorithm B)
    - in the next chapter, we introduce different metrics to rate systems based on their precision, recall, and utility measure

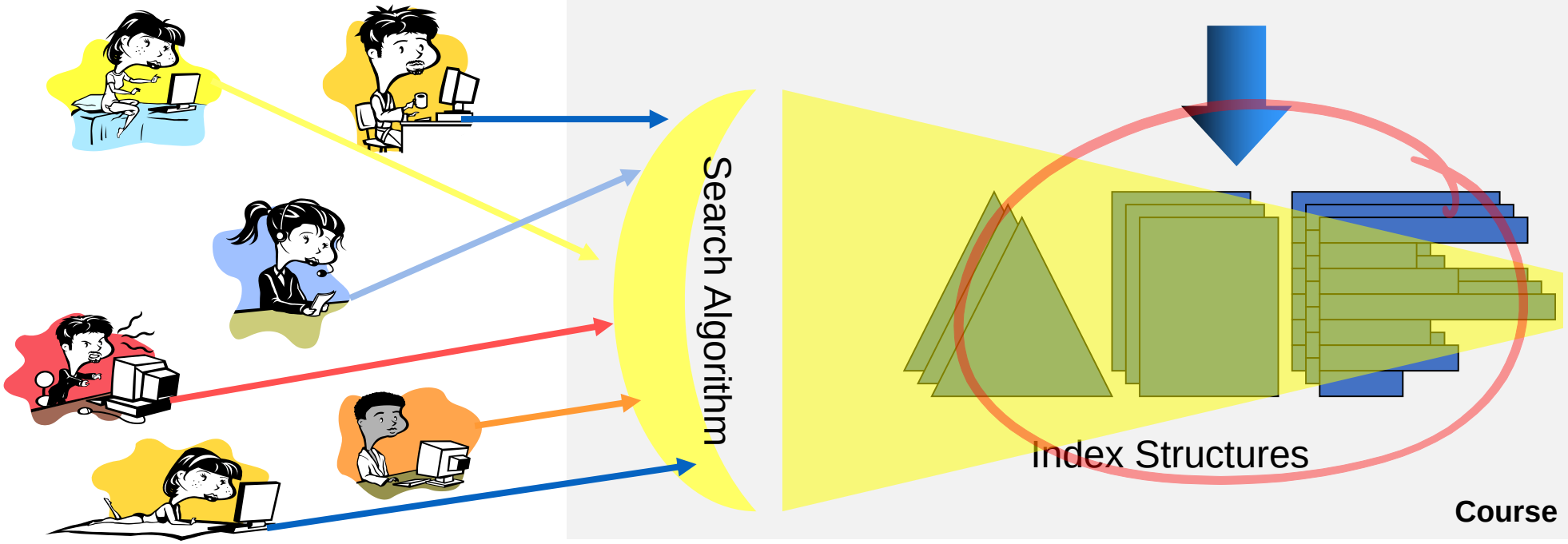
- The principle ingredients of information retrieval



Document Base



Features



0.2.1 Content Modeling and Feature Extraction

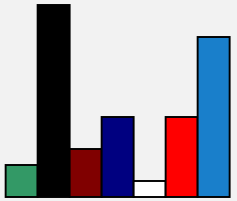
- Due to the enormous amount of data stored in documents of a collection, indexing of documents must be implemented in fully automated way (no human interaction). But how can we describe the content of a document in as dense as possible way? Depending on the document type, different approaches exist:

Text Documents

Feature Extraction

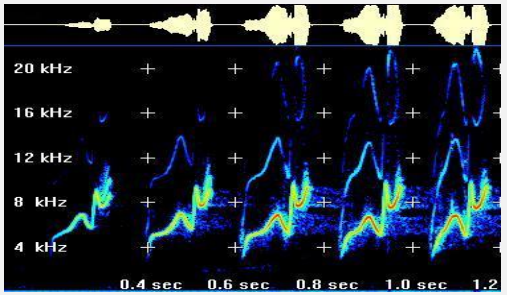
docID = doc10
dog → word 10, word 25
cat → word 13
home → word 2, word 27
...

Images



Color Histogram

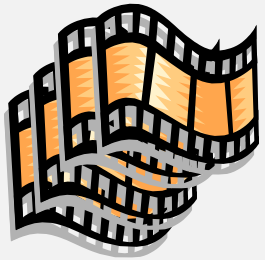
Audio Files



Phonemes: imnOrd@namfo:rmita:gs...
Text: Im Norden am Vormittag...

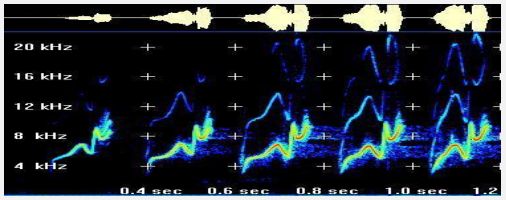
Video Files

Movie



Sequences

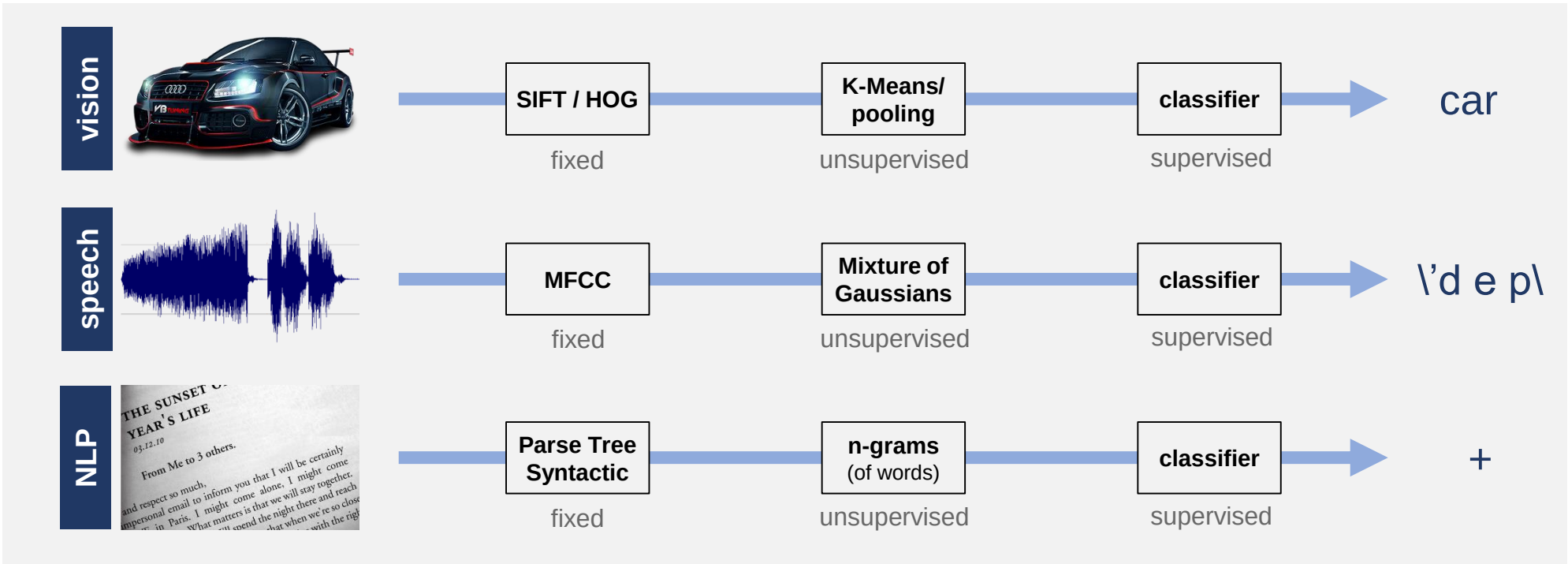
Key Frames



Audio Signal

Subtitle: [President] I never had ....

- 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 Meta Data

- Meta data contain further information about a document or its embedded objects. Such information is typically difficult or not possible to derive out of the signal data. Examples are:
  - Filename, Data Format, Size, Author, Owner, Date, Time, Price
  - Description, Categories, Key Words
  - Relationship to other Objects, Link Structure (Web)
- MPEG-7 is a standard established in 2002 by the Motion Picture Experts Group to model and organize meta data of multimedia documents (MPEG-7 also encompasses computed features). The meta data is stored in an XML format. An advantage of the approach is that the creator of the document himself adds respective meta data; this reduces efforts later on when indexing documents. However, this is also a drawback as the description and key words may be used in different ways (heterogeneity and granularity of description)
- Search over meta data often becomes a fuzzy retrieval task. We have partial information or inaccurate information and are looking for best matches in an efficient way. Signatures nad hashing are good ways to accelerate such searches. Edit distance (and similar metrics) are meaningful tools to look for close matches.

0.2.3 Document Types Considered in this Course

- The course addresses retrieval models for the following document types:
  - **Text**,
  - **Web Page,**
  - **XML**
  - **Image**
  - **Audio File**
  - **Video File**

  The course will describe methods to extract features, to organize features in an index structure, and to quickly retrieve relevant items for a query.

- The primary focus is on feature extraction, indexing and searching. The course will provide some mathematical backgrounds and the most basic idea of the algorithm, but we cannot go into further details of implementations (but we look at some frameworks like lucene and tensor flow).

0.2.4 Search Paradigms and Retrieval Models

- Typing in the URI of a document (or using a bookmark or link)

- If the URI of the document is not known, a user typically enters a query into the search form of a search service. There are a few differences how to enter queries and search for relevant documents:
  - **Browsing:** Yahoo! started with a huge catalog listing all web sites below a well-defined structure. Each web site was manually added to the catalogue. The user could then easily navigate through the structure to retrieve a list of relevant web sites. Most retrieval systems provide this type of search, however, maintaining the catalogue is rather expensive

    e.g. Animal -> Fish -> Perch

  - **Keyword-based Search**: Today, the typical approach is to enter a few keywords and to browse through huge result lists to find relevant documents. This especially works fine with text documents (or spoken text in audio files); all other media types, however, suffer from a semantic gap between keywords an signal information (unless there exists meta data). Furthermore, different models exist to evaluate keyword-based queries:
    - Boolean Retrieval: „white AND house“ „white ADJ house“
    - Vector Space Retrieval: retrieval with weighted terms

- Search Paradigms (contd):

Partially

  - **Meta Search**: The result lists of a number of search services are combined to retrieve better results (best of breeds). Documents that appear in all results at the top are considered to be more important than documents that are once or twice in the top of a result list. Documents can also appear in groups.
  - **Special Search Engines**: A few search engines focus on specific query types or query contexts: Ahoy was a home page searcher, CiteSeer indexes research papers, phone book search engines, or searches through train timetables.

Focus

  - **Similarity Search**: Instead of entering a few keywords, the user enters a few examples how the result should look like ("query by example"). The search engine then finds documents that match best to the patterns of the query. 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.
  - **Interaction & Visualization**: A main problem of a search is to find the right starting points (keywords, examples) such that the relevant documents appear at the top. Relevance feedback is an interactive technique to refine queries automatically based on user ratings on result entries (relevant, non-relevant). A final question addresses the problem of how to present result lists to the user. This is especially a problem with audio files and video files; but other document type may benefit from improved presentations as well.

0.2.5 Index Structures

- Feature extraction algorithms condense 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. In the following, a short overview of the index structures addressed in this course:
  - **Inverted List**: 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.
  - **Signatures**: applicable to meta data and test documents. Signatures contain a rather compact representation of the content and lend themselves perfectly to fuzzy retrieval (error robustness in the document and the query): e.g., the query "Maier" may also automatically retrieve documents with terms "Meier", "Meyer", or "Mayer".
  - **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.

0.2.6 Ranking of Results

- Queries against large collections typically lead numerous 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 in the result 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).
  - **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 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 (26 pages, 1 hours)

Chapter 1: Performance Evaluation (27 pages, 3 hours)

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

Chapter 2: Text Retrieval (67 pages, 7 hours)

- Term Extraction
- Models for Text Retrieval
- Index Structures

Chapter 3: Web Retrieval (31 pages, 4 hours)

- Size of the Web and its Coverage by Search Engines
- Ordering Web Pages with the Example of Google
- Considering the Context of a Page (Hubs & Authorities, PageRank)
- Architecture of a Search Engine and Crawler

Chapter 4: Basic Image, Audio, and Video Retrieval (127 pages, 10 hours)

- Feature Extraction and Relevance Evaluation (Similarity, Distance)
- Methods and Techniques for Images
- Methods and Techniques for Audio Files
- Methods and Techniques for Video Files

Chapter 5: High-level Features with Machine Learning (97 pages, 10 hours)

- Machine Learning Basics
- Data Preparation
- Methods and Applications

Chapter 6: Similarity Search (45 pages, 4 hours)

- Meta Data Search
- Index Structures for Similarity Search
- Evaluation of Complex Queries

Chapter 7: Relevance Feedback (15 pages, 1 hours) [usually at risk to be dropped]

- Feedback Models

Date	Time	Content		Exercise
Sep 17	14 – 17	Chapter 0:	Chapter 1:	
Sep 24	14 – 17	Chapter 1:	Chapter 2:	
Oct 1	14 – 17	Chapter 2:		Ex 1
Oct 8	14 – 17	Chapter 3:		
Oct 15	14 – 17	Chapter 3:	Chapter 4:	Ex 2
Oct 22	14 – 17	Chapter 4:		
Oct 29	14 – 17	Chapter 4:		
Nov 5	14 – 17	Chapter 4:		Preparation for Exam
Nov 12	14 – 17	Chapter 5:		
Nov 19	14 – 17	Chapter 5:		Evaluation Course
Nov 26	14 – 17	Chapter 6:		
Dec 3	14 – 17	Chapter 6:		
Dec 10	14 – 17	Chapter 6:		
Dec 17	14 – 17	Chapter 7:		Feedback Course

See course web site for latest update of schedule