Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

CAIDAS

WüNLP

ALGORITHMS IN AI & DATA SCIENCE 1 (AKIDS 1)

Course Introduction

Prof. Dr. Goran Glavaš

23.10.2023

Course title

- Algorithms?

- Artificial Intelligence?

- Data Science?

Algorithm: Definitions

A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

Oxford dictionary

A finite sequence of rigorous instructions, typically used to solve a class of specific problems or to perform a computation.

Wikipedia

Any well-defined computational procedure that takes some (set of) value(s) as input and produces some (set of) value(s) output.

Cormen et al.

What are algorithms built from?

- **Building blocks** of algorithms
  - Elementary operations
  - Sequential processing (**one** processing line)
  - Parallel processing (**multiple** processing lines)
  - Conditions (conditioned execution)
  - Loops (repetition)
  - Subprograms (modular construction of an algorithm)
  - Recursion (more later in the course)

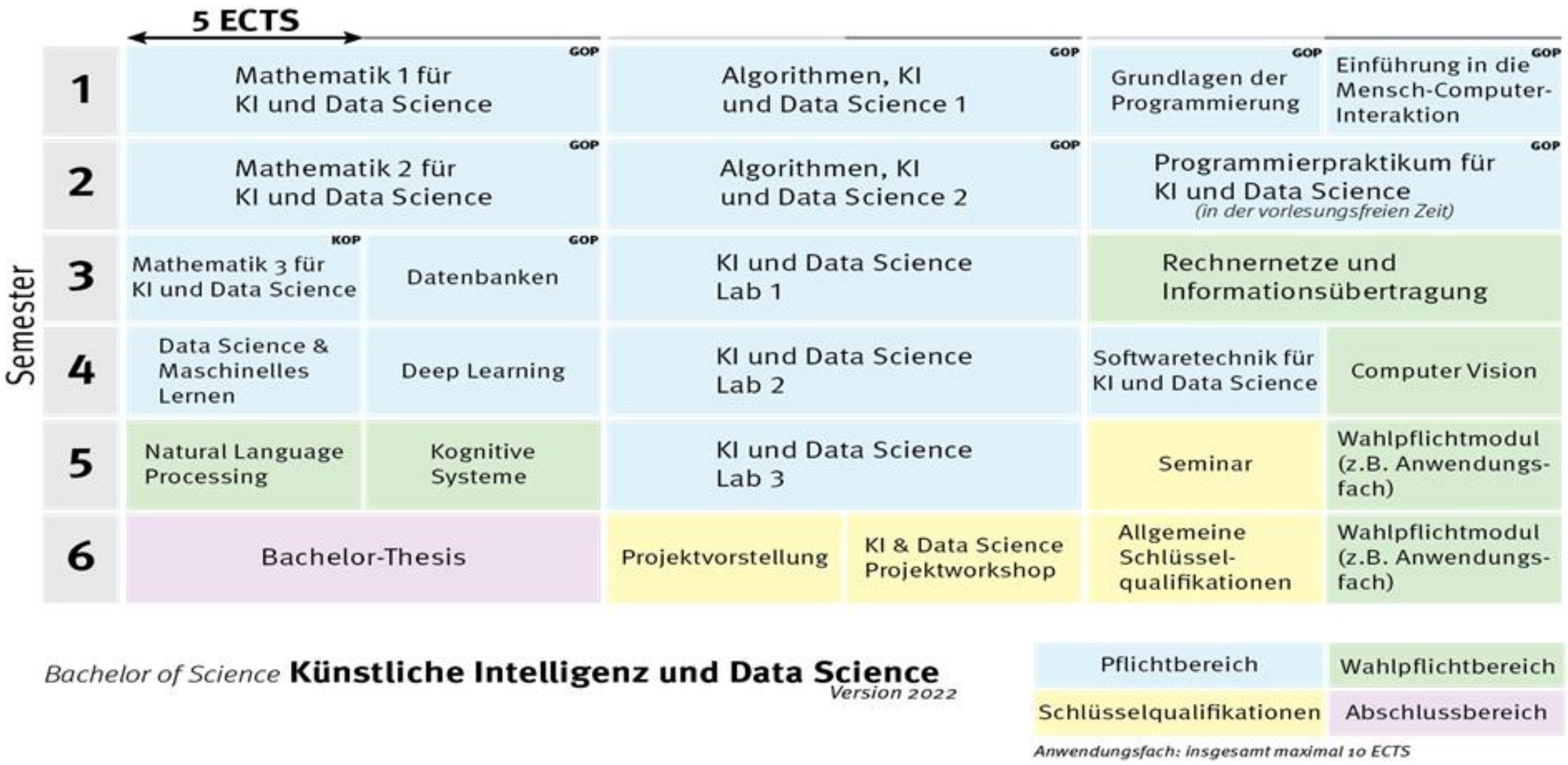
Let’s build an algorithm

- **Algorithms** are not strictly tied to computers and programming
  - All around us: you’re doing many things **„algorithmically”**
  - Loosely, it’s any formalized process of how to do something
    - How to conver inputs to outputs

Task: enroll into WueCampus courses of all ”Veranstaltungen” recommended by the „Studienaufbau” for a given semester

Let’s build an algorithm

Task: enroll into WueCampus courses of all ”Veranstaltungen” recommended by the „Studienverlaufsplan” for a given semester



- **Input** (Eingabe)? **Output** (Ausgabe)?

Let’s build an algorithm

Task: enroll into WueCampus courses of all ”Veranstaltungen” recommended by the „Studienaufbau” for a given semester

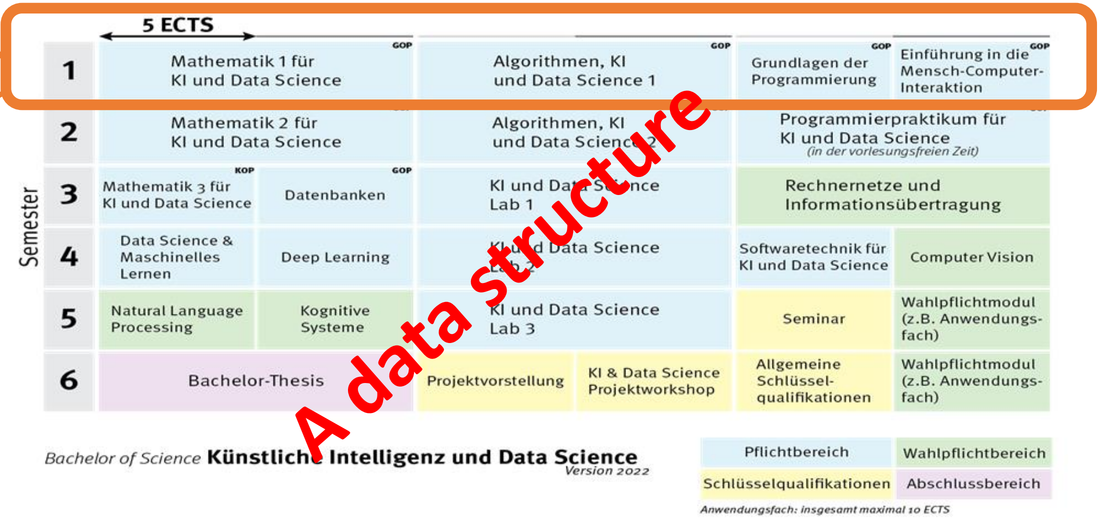
- **Input:** semester number, student identification
- **Output:** list of WueCampus courses with successful enrollment

- **Data (structures)**
  - Information has to be stored/written/recorded somewhere, somehow

Let’s build an algorithm

Task: enroll into WueCampus courses of all ”Veranstaltungen” recommended by the „Studienaufbau” for a given semester

1. **step:** get the recommended courses for the **input** semester
    - **Input**: 1. semester
    - **Output:**
        - Grundlagen d. Programm.
        - Einführung MCI
        - AKIDS 1
        - Mathe für KI/DS1



Let’s build an algorithm

Task: enroll into WueCampus courses of all ”Veranstaltungen” recommended by the „Studienaufbau” for a given semester

2. step: for each course,

2a) find the corresponding WueCampus course

- We need also a **data structure** that maps („translates“) courses from „Studienaufbau“ data structure into WueCampus pages

Veranstaltung	WueCampus Kurs
AKIDS1	https://wuecampus2.uni-wuerzburg.de/moodle/enrol/index.php?id=56119
Grundlagen d. Programm.	https://wuecampus2.uni-wuerzburg.de/moodle/enrol/index.php?id=54727
…	…

Let’s build an algorithm

Task: enroll into WueCampus courses of all ”Veranstaltungen” recommended by the „Studienaufbau” for a given semester

2. **step: for each course,**

2b) enroll the student into the WueCampus course

- Request enrollment of student *s* in WueCampus course *c*
- Get back the information whether the enrollment was successful

Let’s build an algorithm: **pseudocode**

Pseudocode is an artificial and informal language that helps us **develop algorithms**. It can be seen as a "text-based” tool for designing algorithms.

- Programming languages (e.g., Python, Java, C++) are artificial (as opposed to **natural** human languages), but formal

```
enroll_student_into_all_semester_courses
Input: stud_ID, sem_no
Data: studien_verlaufs_plan # maps semesters to courses
      wuecamp # maps courses to WueCampus courses/URLs

courses <- look into studien_verlaufs_plan for sem_no # our „step” 1
enrolled <- [] # empty list
for each course c in courses # step 2, iterating over all obtained courses
      wcc <- look into wuecamp for c # our „step” 2a
      success <- enroll(stud_ID, wcc)
      if success = True
         add wcc to enrolled

Output: enrolled
```

Let’s build an algorithm: **pseudocode**

Pseudocode is an artificial and informal language that helps us **develop algorithms**. It can be seen as a "text-based” tool for designing algorithms.

- **Building blocks** of algorithms
  –

```
enroll_student_into_all_semester_courses
Input: stud_ID, sem_no
Data: studien_aufbau # maps semesters to courses
      wuecamp # maps courses to WueCampus courses/URLs

courses <- look into studien_aufbau for sem_no # our „step” 1
enrolled <- [] # empty list
for each course c in courses # step 2, iterating over all obtained courses
      wcc <- look into wuecamp for c # our „step” 2a
      success <- enroll(stud_ID, wcc)
      if success = True
        add wcc to enrolled

Output: enrolled
```

Elementary (atomic) operations

- Cannot be broken into smaller suboperations
- Basiselemente eines Algorithmus, die nicht näher aufgeschlüsselt werden

Let’s build an algorithm: **pseudocode**

Pseudocode is an artificial and informal language that helps us **develop algorithms**. It can be seen as a "text-based” tool for designing algorithms.

- **Building blocks** of algorithms
  –

```
enroll_student_into_all_semester_courses
Input: stud_ID, sem_no
Data: studien_aufbau # maps semesters to courses
      wuecamp # maps courses to WueCampus courses/URLs

courses <- look into studien_aufbau for sem_no # our „step” 1
enrolled <- [] # empty list
for each course c in courses # step 2, iterating over all obtained courses
      wcc <- look into wuecamp for c # our „step” 2a
      success <- enroll(stud_ID, wcc)
      if success = True
         add wcc to enrolled

Output: enrolled
```

Sequential processing

- We execute steps one after another
- Often we need some intermediate results from a previous step in the next step

Let’s build an algorithm: **pseudocode**

Pseudocode is an artificial and informal language that helps us **develop algorithms**. It can be seen as a "text-based” tool for designing algorithms.

- **Building blocks** of algorithms
  –

```
enroll_student_into_all_semester_courses
Input: stud_ID, sem_no
Data: studien_aufbau # maps semesters to courses
      wuecamp # maps courses to WueCampus courses/URLs

courses <- look into studien_aufbau for sem_no # our „step” 1
enrolled <- [] # empty list
for each course c in courses # step 2, iterating over all obtained courses
      wcc <- look into wuecamp for c # our „step” 2a
      success <- enroll(stud_ID, wcc)
      if success = True
         add wcc to enrolled

Output: enrolled
```

Conditions

- We execute step(s) only **if** a condition is satisifed
- **falls** ***Bedingung*** **dann** Schritt(e) **sonst** Alternativschritt(e)

Let’s build an algorithm: **pseudocode**

Pseudocode is an artificial and informal language that helps us **develop algorithms**. It can be seen as a "text-based” tool for designing algorithms.

- **Building blocks** of algorithms
  –

```
enroll_student_into_all_semester_courses
Input: stud_ID, sem_no
Data: studien_aufbau # maps semesters to courses
      wuecamp # maps courses to WueCampus courses/URLs

courses <- look into studien_aufbau for sem_no # our „step” 1
enrolled <- [] # empty list
for each course c in courses # step 2, iterating over all obtained courses
      wcc <- look into wuecamp for c # our „step” 2a
      success <- enroll(stud_ID, wcc)
      if success = True
         add wcc to enrolled

Output: enrolled
```

Loops (Schleife, repetition)

- We repeat the execution of some steps
- **Repetition**:
  - **FOR**: Fixed number of times
  - **WHILE**: as long as some condition is satisfied

Let’s build an algorithm: **pseudocode**

Pseudocode is an artificial and informal language that helps us **develop algorithms**. It can be seen as a "text-based” tool for designing algorithms.

- **Building blocks** of algorithms
  –

```
enroll_student_into_all_semester_courses
Input: stud_ID, sem_no
Data: studien_aufbau # maps semesters to courses
      wuecamp # maps courses to WueCampus courses/URLs

courses <- look into studien_aufbau for sem_no # our „step” 1
enrolled <- [] # empty list
for each course c in courses # step 2, iterating over all obtained courses
      wcc <- look into wuecamp for c # our „step” 2a
      success <- enroll(stud_ID, wcc)
      if success = True
         add wcc to enrolled

Output: enrolled
```

Subprograms / modules

- Functionality (sequences of steps) that need to be reused in different contexts (different programs)
- Modularized into subprograms
  - „Functions” or „Methods”

Course title

- Algorithms?
- Artificial Intelligence?
- Data Science?

Artificial Intelligence: Definitions

Theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages.

Oxford dictionary

Intelligence demonstrated by machines, as opposed to the natural intelligence displayed by animals and humans.

Wikipedia

The field of study of intelligent agents, which refers to any „rational agent" that perceives its environment and takes actions that maximize its chance of achieving its goals.

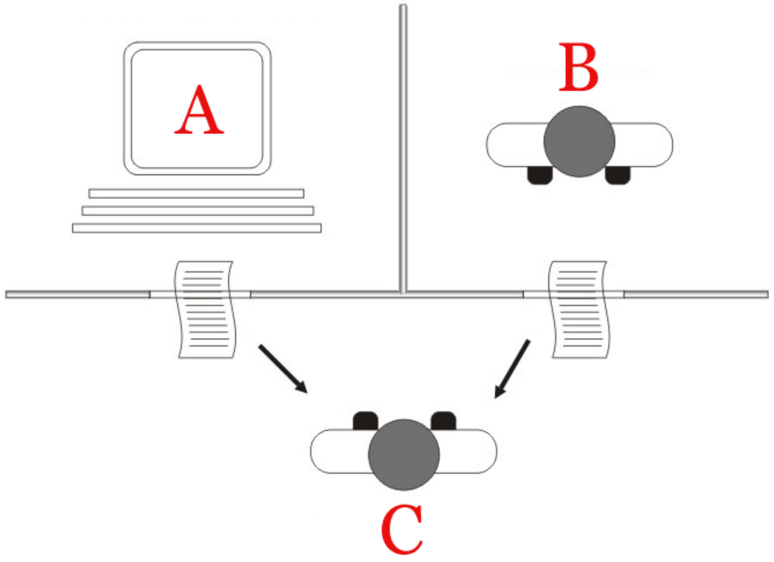
Russell & Norvig

The Turing Test

Theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages.

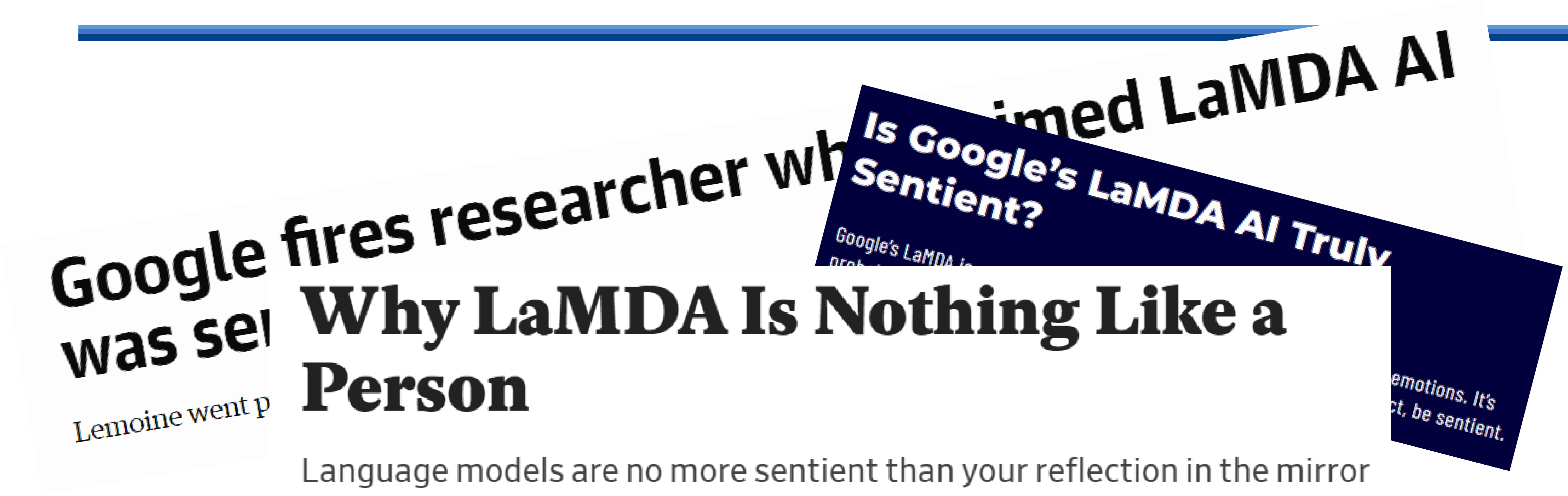
Oxford dictionary

- Computer systems that act like humans
- Machine behaviour indistinguishable from human?
- **Turing test**: „the imitation game”
  - Natural language (text) communication
  - Can C successfully guess (better than chance) whether the answers come from A (machine) or B (human)?



- **Natural language communication and language understanding** represent one of the main pillars of human inteligence

The Turing Test: Large Language Models?



In June 2022 the Google LaMDA (Language Model for Dialog Applications) chatbot received widespread coverage regarding claims about it having achieved sentience. Initially in an article in *The Economist* Google Research Fellow Blaise Agüera y Arcas said the chatbot had demonstrated a degree of understanding of social relationships.[99] Several days later, Google engineer Blake Lemoine claimed in an interview with the *Washington Post* that LaMDA had achieved sentience. Lemoine had been placed on leave by Google for internal assertions to this effect. Agüera y Arcas (a Google Vice President) and Jen Gennai (head of Responsible Innovation) had investigated the claims but dismissed them.[100] Lemoine's assertion was roundly rejected by other experts in the field, pointing out that a language model appearing to mimic human conversation does not indicate that any intelligence is present behind it,[101] despite seeming to pass the Turing test. Widespread discussion from proponents for and against the claim that LaMDA has reached sentience has sparked discussion across social-media platforms, to include defining the meaning of sentience as well as what it means to be human.

(Very) Brief History of AI

- Various disciplines – Philosophy, Mathematics, Psychology, Linguistics, and Computer Science – have all
    - Posed questions about machine intelligence and links to human intelligence
    - Developed AI methods and results

- AI is as old as Computer Science
    - Origins of AI: **1943-1956**
    - **Development of first computers**
        - „Can computers copy the humans (or human mind)"?

(Very) Brief History of AI

- **Dartmouth summer research project on Artificial Intelligence**
  - Summer **1956**
  - Considered the birthplace of AI as a discipline
  - Leading American mathematicians & CS minds aiming to solve a range of problems
    - John McCarthy (Dartmouth)
    - Marvin Minsky (MIT)
    - Nathaniel Rochester (IBM)
    - Claude Shannon (Bell Laboratories)
    - …

(Very) Brief History of AI

- **Symbolic AI**: 1956-1974
  - „Reasoning as search" (state space search)
  - Natural language processing
    - Daniel Bobrow's **STUDENT** program solving high-school algrebra word problems
    - **Semantic net** connecting concepts (e.g., „house" has „door") by Roger Schank
    - **ELIZA**, the first chatbot, by Joseph Weizenbaum

> Schank developed the model to represent knowledge for natural language input into computers. [...] His goal was to make the meaning independent of the words used in the input, i.e. two sentences identical in meaning, would have a single representation. The system was also intended to draw logical inferences
>
> **Source: Wikipedia**

  - Micro-worlds by Minsky and Papert
  - Automata and robotics (in Japan at Waseda University, WABOT-1 in 1972)

(Very) Brief History of AI

- Promise of Artificial General Intelligence (AGI)
  - Dartmouth workshop and subsequent activities generated a lot of **optimism** (one might say **„hype”**) about AI potential

„Within ten years a digital computer will be the world's chess champion" and "within ten years a digital computer will discover and prove an important new mathematical theorem”

Herbert Simon & Allen Newell, 1958

Machines will be capable, within twenty years, of doing any work a man can do.”

Herbert Simon, 1965

Within a generation ... the problem of creating 'artificial intelligence' will substantially be solved.

Marvin Minsky, 1967

In from three to eight years we will have a machine with the **general intelligence** of an average human being.

Marvin Minsky, 1970

(Very) Brief History of AI

- **First AI Winter**: 1974-1980
  - Criticism and financial setback
  - Primarily due to underestimating the difficulty of the problems
  - Minsky's devastating criticism of perceptron (simplest neural network)

  - Limited computational power:
    - Not enough memory & compute
  - Limited data
    - Some domains – like (computer) vision or natural language communication – require an incredible amount of information about the world

(Very) Brief History of AI

- **The 80s Boom**: 1980-1987
  - The rise of **expert systems**
    - Organizing and encoding background knowledge and knowledge about the world as the backbone of reasoning
    - The „knowledge revolution”
  - Progress in **connectionism** (i.e., neural nets)
    - Backpropagation – an effective method for training multi-layer neural networks popularized by Geoffrey Hinton and David Rumelhart
    - Hopfield net – John Hopfield proved that a form of a neural net learns and processed information in a new way

(Very) Brief History of AI

- **Second AI Winter**: 1987-1993
  - Dissapointment with the limitations of the expert systems
  - Collapse of the market for AI specialized hardware in 1987
  - 300+ companies shut down or gone bankrupt
    - End of first commercial wave of AI

- **Stabilization period**: 1993-2011
  - Incremental, stable improvements, primarily because of better hardware
  - Machine learning advances, e.g., Support Vector Machines in 1995
  - Deep Blue beats the world chess champion (Garry Kasparov) in 1997
  - However, no hype, no big promises

(Very) Brief History of AI

- **Deep Learning revolution**: from 2011 and **ongoing**
  - AI had solid results in closed, simple domains and problems
  - Severe limitations in two crucial „open-ended" disciplines
    - **Computer Vision** (intelligent machines must be able to interpret their visual environment)
    - **Natural Language Processing** (intelligent machines must be able to „communicate" with humans in their preferred way of communication)
    - „Anything" can appear in an image or be said/written in language

  - **Deeper (neural) models** trained on more and more data
    - Models invented long before now proven useful with sufficient compute and data
      - Convolutional networks: LeCun, 1989
      - Long-Short-Term-Memory recurrent networks: Schmidhuber, 1997
    - Attention-based networks (Transformers): Vaswani, 2017; Devlin 2018

Course title

- Algorithms?
- Artificial Intelligence?
- Data Science?

Data Science: Definitions

Interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract or extrapolate knowledge and insights from noisy, structured and unstructured data, and apply knowledge from data across a broad range of **application domains**.

Wikipedia

The use of scientific methods to obtain useful information from (large amounts of) computer data.

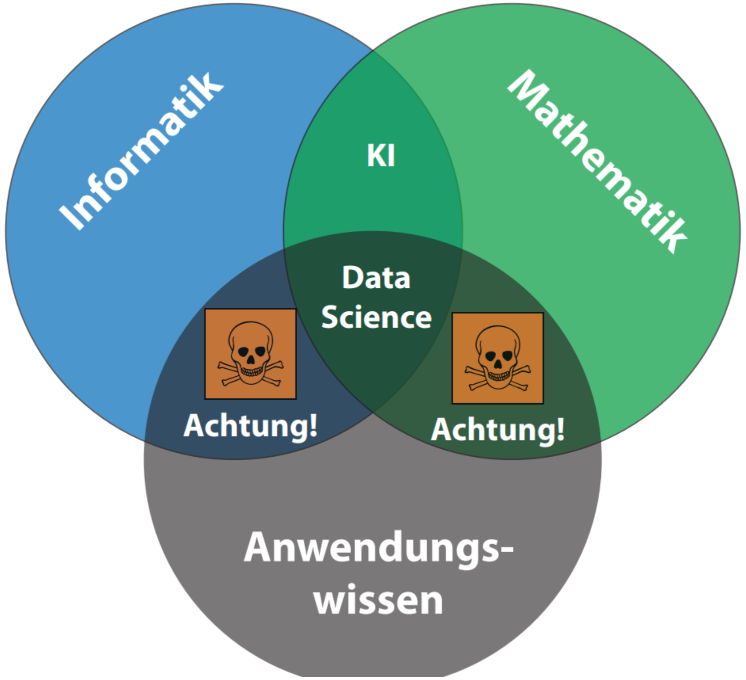
Cambridge dictionary

A field that deals with **advanced data analytics** and modeling, using mathematics, statistics, programming, and machine learning to extract valuable, often predictive information from large data sets.

Dictionary.com

Data Science

- **Data science** = **AI methods** + data from an **application area**
  - Yields insights/knowledge/information in the application area



Course Organization

Brought to you by: WüNLP

https://www.informatik.uni-wuerzburg.de/nlp

WÜNLP | NEWS | TEACHING | RESEARCH | TEAM

NATURAL LANGUAGE PROCESSING

We at the Chair for Natural Language Processing (Computer Science XII) try to make **machines understand human language**! In fact, we try to make them understand very many different human languages. We primarily focus on written text (after all, speech can always be transcribed to text). Methodologically, the work of the group focuses on **deep learning and representation learning methods for semantic modeling of natural language** (that is, precise modeling of meaning of natural language statements and text documents), with the special focus on multilingual representation learning and cross-language transfer of models for concrete NLP tasks.

Driven by deep learning advances, NLP has lately seen substantial progress, primarily due to the technical ability to (pre)train ever larger neural models on ever more text. Such progress can be exclusive as its benefits are beyond reach for most of the world's population (e.g., speakers of low-resource languages, anyone who lacks computational resources needed to train or use these models). Moreover, training ever larger language models based on complex neural architectures (for example, the popular Transformer) has a large carbon footprint and such models tend to encode a wide range of negative societal stereotypes and biases (e.g., sexism, racism). At WüNLP we specifically address these challenges and aim to **democratize state-of-the-art language technology**. To this end, we pursue three research threads that we hope will lead to **equitable, societally fair, and sustainable language technology**: *(i) sustainable, modular, and sample-efficient NLP models, (ii) fair and ethical (i.e., unbiased) NLP,* and *(iii) truly multilingual NLP, with special focus on low-resource languages.*

Text data is all around – besides the core methodological NLP work, we also work on **interdisciplinary projects** where we apply cutting-edge NLP methods to interesting problems from other disciplines, most prominently in the area of Computational Social Science (and so far most often in collaboration with political scientists).

Our Chair has **international prominence and visibility**. We regularly publish our research results at the very competitive **top-tier NLP conferences** (**ACL**, **EMNLP**, **NAACL**, **EACL**). Further, Prof. Glavaš served as an **Editor-in-Chief** for the **ACL Rolling Review**, the centralized reviewing service of the **Association for Computational Linguistics**. We have established numerous research collaborations, most prominently with the Language Technology Group of the University of Cambridge., CIS at LMU München, and UKP at TU Darmstadt.

News

› Three papers accepted at EMNLP 2022

WüNLP will have three papers in the Main Conference Program of Empirical Methods in Natural Language Processing, one of the most prestigious venues in

› Welcome Fabian David Schmidt!

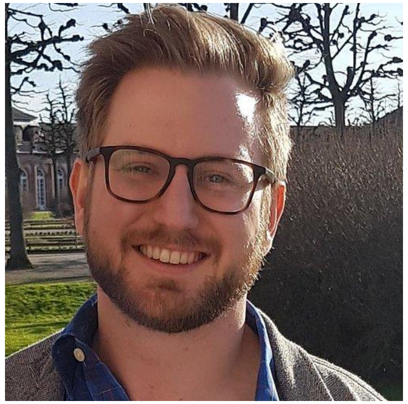
Fabian David Schmidt will join WüNLP from July!
› Mehr

› Two papers accepted for NAACL 2022

WüNLP will have two papers in the Main Conference Program of the Conference of the North-American Chapter of the Association for Computational

The Team

Prof. Dr. Goran Glavaš
Lectures



Fabian David Schmidt
Exercises

Benedikt Ebing
Exercises

Content

Block I: Algorithms and Data Structures Fundamentals

- **L1**: Introduction to algorithms, AI, and Data Science.
- **L2**: Algorithms: fundamentals
- **L3**: Data structures
- **L4:** Algorithmic complexity
- **L5:** Sorting algorithms
- **L6:** Priority queue and Heapsort
- **L7:** Hashing
- **L8:** Binary search trees
- **L9:** Balanced trees
- **L10:** Graphs
- **L11:** Graph Algorithms
- **L12 & L13**: Dynamic programming (with examples)

Content (planned!)

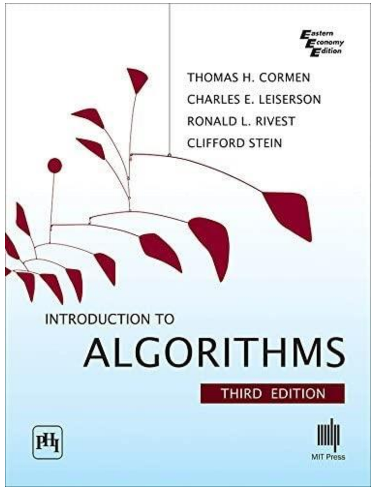
Block II: Artificial Intelligence and Data Science Algorithms

- **L14:** Uninformed state space search
- **L15:** Informed search: A* algorithm
- **L16**: Adversarial search
- **L17:** Metaheuristic search
- **L18:** Constrained optimization (backtracking)
- **L19:** Rule-based reasoning: expert systems
- **L20**: Numerical optimization
- **L21**: Introduction to Machine Learning
- **L22**: Parametric Classification
- **L23**: Non-Parametric Classification
- **L24**: Clustering

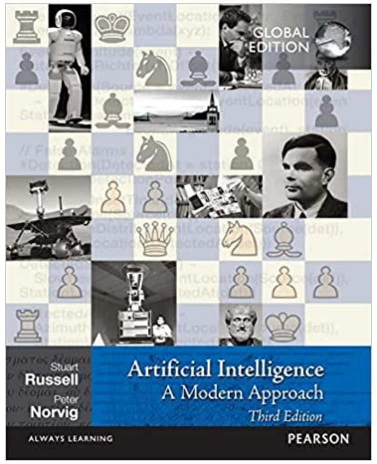
Content (planned!)

- Possible adjustments in our schedule/plan along the way

Literature



- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to Algorithms*. MIT press.



- Russell, S. J., Norvig., P. *Artificial intelligence: A Modern Approach*. Pearson Education, Inc.

Activities and Time Slots

- **Lectures**
  - On-site: Turing-HS (in Computer Science building, Hubland Süd)
  - Two lectures per week:
    - **Monday**: 14.15-15.45
    - **Thursday**: 16.15-17.45
  - You are welcome to **interrupt and ask questions** at any point during the lecture**!!!**

- **Exercises**
  - On-site: SE II Physik building, ground floor)
  - One exercise session per week
    - **Friday**: 10.15-11.45
  - „On-the-paper" exercises & coding exercises
  - Exercises start this Friday, **27.10.**

Language

- **Lectures**: in English, **but**
  - Students welcome to **ask questions / for clarifications in German**
- **Exercises:** in German, **but**
  - Students welcome to **ask questions / for clarifications in English**
- **Other communication**
  - For example, emails
  - German or English, as preferred by the student: we will reply in the same language

The English language is sometimes described as the *lingua franca* of computing. In comparison to other sciences, where Latin and Greek are often the principal sources of vocabulary, computer science borrows more extensively from English.

A **lingua franca** (/ˌlɪŋɡwə ˈfræŋkə/; lit. 'Frankish tongue'; for plurals see § Usage notes),[1] also known as a bridge language, common language, trade language, auxiliary language, vehicular language, or link language, is a language systematically used to make communication possible between groups of people who do not share a native language or dialect, particularly when it is a third language that is distinct from both of the speakers' native languages.[2]

Homeworks & Bonus points

- **Exercise sheets** will be given **1 week** in advance
  - Shared after the corresponding lecture
  - Solved in groups of **3 students**
  - Submit the solutions per Moodle (WueCampus)
  - Correct solutions will be discussed in the exercise session

  - **Example**:
    - Exercise shared: 27.10.; Student solutions submitted: 3.11. in the morning
    - Exercise session: 3.11. in the afternoon

  - Submitted homeworks will be evaluated on a **3-grade scale**
    - **Insufficient**: 0 points; **Sufficient**: 1 point; **Good**: 2 points

Homeworks & Bonus points

- **12 exercise/homework sheets**
  - Maximum 24 bonus points
  - With 20+ bonus points you earn an **exam bonus**

- **Exam bonus**
  - If you pass (grade **4.0 or better**), exam bonus improves your grade by one level
  - Eine Notenstufe bei Bestehen der ersten Klausur nach Semesterende
  - Example: 2.0 -> 1.7

Exam / Klausur

- Termin und Ort **wird noch bekannt gegeben**
  - Irgendwann nach dem 09.02.2023
- Bonus (je eine Notenabstufung (+0.3/0.4) bei mind. 4.0)
  - Voraussetzung: erreichen von insgesamt 50% der Gesamtpunkte
  - Gilt sowohl für die erste Klausur als auch für die Nachklausur
- Anmeldung:
  - Muss im Anmeldezeitraum (bis Mitte Januar 2023) über WueStudy erfolgen
- WICHTIG
  - Nichterscheinen zur Klausur zählt als "nicht bestanden"!

How hard is this going to be?

- AKIDS 1 is a **fundamental** course
  - Basis for many other „Vertiefung" and specialization courses later in your study program
  - You should take it seriously
- **10 ECTS** = **250 to 300 hours** of work
  - Ca. 80 hours go to lectures and exercises
  - That still leaves **170-220 hours** of self-study

To „**Study**”: School vs. Uni

School	University
Your time is structured by others.	You manage your own time.
You can count on parents and teachers to remind you of your responsibilities and to guide you in setting priorities.	You must judge what your responsibilities are and set your own priorities. You might face moral and ethical decisions you have never faced before.
Each day you proceed from one class directly to another, spending 6 hours each day (30 hours a week) in class.	You often have hours between classes; class times vary throughout the day and evening and you spend only 12 to 16 hours each week in class.
Guiding principle: You will usually be told what to do.	Guiding principle: You are expected to take responsibility for what you do and don't do, as well as for the consequences of your decisions.

To „**Study**": School vs. Uni

School	**University**
You may study outside class as little as 0 to 2 hours a week, and this may be mostly last-minute test preparation.	For each paper, an 'expected' workload guide for a week is roughly 15 hours per week over a 12-week semester, but that includes contact time – lectures, labs etc as well as working on assignments.
You seldom need to read anything more than once, and sometimes listening in class is enough.	You need to review class notes and text material regularly, and you may have to read some material several times to fully grasp it.
You are expected to read short assignments that are then discussed, and often re-taught in class.	You are assigned substantial amounts of reading and writing which may not be directly addressed in class.
Guiding principle: You will usually be told in class what you need to learn from assigned readings.	Guiding principle: It's up to you to read and understand the assigned material; lectures and assignments proceed from the assumption that you've already done so.

All it Takes is Dedication and Effort!

Questions?

Turite klausimų?
Awọn ibeere?
ਸਵਾਲ?
Küsimusi?
Pitanja?
Sorusu olan?
Dúvidas?
質問は？
有问题吗？
Fragen?
¿Preguntas?
Questions?
Frågor?
Pytania?
Domande?
Питання?
Vragen?
Ερωτήσεις;
Porandukuéra?
คำถาม?
أسئلة؟