Computational Argumentation – Part I (Tutorial)

Basics of Scientific Presentation

Henning Wachsmuth
henningw@upb.de

April 16, 2019

partly based on slides of
Prof. Dr. Gregor Engels and
Prof. Dr. Steffen Becker

PADERBORN UNIVERSITY

Outline

- **Literature research**
  Types, quality, reading, acquisition, and organization

- **Oral presentations in general**
  Content, structure, style, talking, and timing

- **Lecture talks in particular**
  Important differences to ”normal“ oral presentations

- **Written presentations left out here**
  Content, structure, style, citations, and plagiarism

Literature research

Doing literature research

- **Literature research**
  - Fundamental task in science
  - Time-intensive and tedious — but necessary
  - Often, the first task to be done

- **Literature research in general**
  - Obtain all information relevant to the scope of the problem
  - Obtain background information
  - Obtain evidence for your or others‘ claims

  ... and similar reasons

- **Literature research in science**
  - Find out if your approach to a problem is new
  - Find alternative approaches or perspectives
  - You are rarely the first to work on a problem

    If you are, what does that tell you?
  - Don‘t reinvent the wheel

Selecting literature

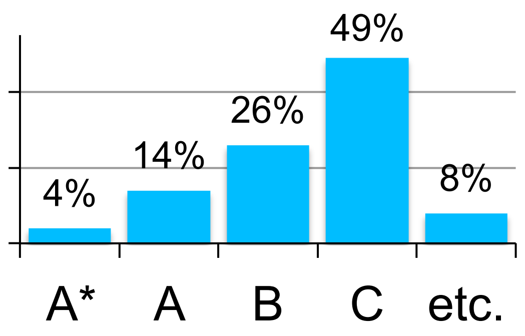
- **Types of literature (and similar)**
  1. Books. Theory, basics, approved techniques
  2. Scientific journal papers. Completed research lines
  3. Conference papers. State-of-the-art research
     In our field, major publication type
  4. Workshop papers. New ideas, ongoing research
  5. Conference/Online tutorials. Easy access to basics and techniques
  6. Popular science magazines. Easy access to research lines
  7. Other websites. Anything

- **What type to prefer (in our field)**
  - Generally, literature should be peer-reviewed
    Most literature of types 1–4 is peer-reviewed, but not all
  - Rule of thumb: books > journals > conferences > workshops > tutorials > magazines > websites > other
  - But, for example: top conferences > average journals

Assessing quality of literature

- **Conference and journal rankings**
  - Top tier ranked A$^+$/A* or A; B still good
  - Unranked conferences/journals may be doubtful
    No ranking achieves complete coverage, though
  - One of the most reputable rankings is CORE
    core.edu.au/conference-portal

CORE ranks



- **Number of citations**
  - Roughly indicates importance
  - Rather for *relative* comparisons within a topic
  - Notice: Newer papers naturally tend to have less citations
  - Good resource for citation numbers is Google Scholar scholar.google.de
    Journals also have so called impact factors derived from citation numbers

- **Disclaimer**
  - Good and bad research appears at all places
  - Often, only reading helps

Reading and finding literature

- **Reading papers efficiently**
  1. Read abstract, introduction, and conclusion
  2. Look at figures and tables
  3. Decide whether worth reading everything
  4. Read goal-driven
     Specify questions to be answered during reading

https://pixabay.com

- **Finding the next paper**
  - Follow promising references at the end of a paper
  - Find promising papers citing a paper
  - Learn to identify the best search terms
    Rule of thumb: As specific as possible, but as abstract as needed

- **Getting started in the course**
  1. First read the literature that we provide
  2. Then find further literature

Acquiring literature

https://pixabay.com

- **Obtaining papers**
  - Many papers freely available online
  - Others might be free from a university network
  - If neither, maybe your advisors can help

- **Important sources**
  - ACL Anthology for computational linguistics papers aclweb.org/anthology
  - ACM Digital Library for many important computer science papers dl.acm.org
  - dblp for any literature related to computer science dblp.dagstuhl.de
  - Google Scholar for any scientific literature scholar.google.de

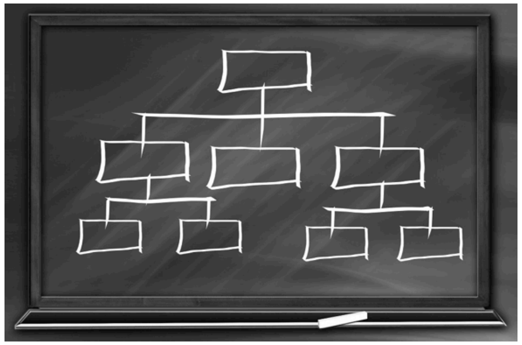
    ... and general web search, of course

- **Accessing books**
  - Check if available in the library
  - Some accessible online, for example, on Google Books books.google.de

    Purchasing books can make sense when of continuous importance for you

Organizing literature

- **Literature organization**
  - Maintain overview, start from the beginning
  - ”Extra“ effort will pay off



- **Create logical folder structure**
  - Build your own view of the field
  - Logically subdivide topics, but don‘t over-engineer

    For instance ./literature/computational-argumentation/argument-mining/ — but maybe not deeper
- **Rename all literature consistently**
  - Simplifies browsing and finding
  - We use <1stauthor><2digityear>-<full-title-lower-case-no-special-chars>.pdf

    For example: ajjour17-unit-segmentation-of-argumentative-texts.pdf
- **Organizing meta-information**
  - Bibliographical information needed when citing literature
  - Store bibtex of literature whenever available

    Learn more on en.wikipedia.org/wiki/BibTeX; many pages such as dblp provide bibtex‘s

Oral presentations in general

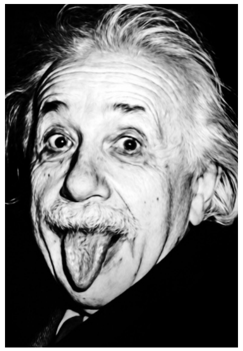
Content of your talk

- **Scientific presentation is storytelling**
  - Tell a coherent story with a central theme
  - Plan what points to make and how to get there
  - Make it exciting, show importance
  - Don‘t be complete, be selective
    Somewhat different for lectures and articles (see below)
  - Avoid surprise: Clarify why you tell something

- **Science needs to be understood**
  - Adjust complexity to audience
  - Leave out formal things, unless needed
    May be different in lectures and articles (see below)
  - Be precise and clear
  - Introduce terms, use consistently
  - Figures and examples help

"Sometimes **reality** is too complex. **Stories** give it a form."

Jean-Luc Godard



"Everything should be as **simple** as possible, but **not simpler**."

Figures

- **Figures**
  - Charts, diagrams, graphs, pictures, drawings, ...
  - Slides are visual
  - Rule of thumb: (Almost) No slide without figure
- **What to use figures for**
  - Primary. Replace text; visually explain concepts, ...
  - Secondary. Support your message with pictures
    (as often done in this presentation)
- **Formatting**
  - Vector graphics whenever possible
  - Other figures: Optimize sharpness, scale down smartly
    Never scale > 100%; 50% is better than 53% — why?
  - Never squeeze or stretch the aspect ratio
    If needed, cut figures on any side instead
  - Check readability of included text

"a **picture** is worth
a **1000 words** "

https://pixabay.com

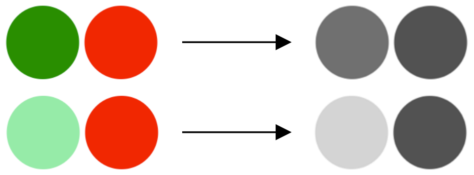
"**unsharpness**
is the mistake that even
lay persons see"
Herbert Kania

Colors

- **Colors in general**
  - Presentations are visual, make use of colors
  - Less colors create a more clear style
  - But natural colors have an appeal, too
- **Font colors for highlighting important points**
  - Use colors consistently
  - Not too colorful
  - I use dark blue here for regular highlighting

    And a cyan-like color for quotes
- **Support your messages**
  - Always the same color for the same concept
  - Can create connections even across slides
- **Color vs. brightness**
  - Think of color blind people — contrast helps

„Everything should be as **simple** as possible, but **not simpler**. "



Tables (and matrices)

- **Tables for what?**
  - Presenting numerical results
  - Comparing alternative ideas, approaches, or similar
  - Listing attribute values of multiple instances

  ... and similar

best results for each ranking approach

#	Dimension	τ	best	worst
1	PageRank	**0.28**	**15**	3
2	Number	0.19	6	**1**
3	Sentiment	0.12	12	4
4	Frequency	0.10	11	9
5	Similarity	0.02	9	10
6	Random	0.00	8	7

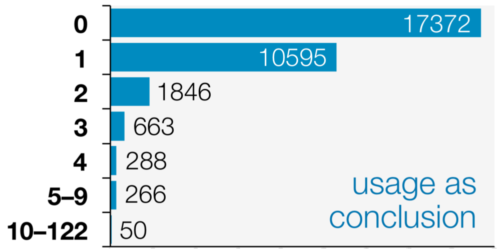
- **Table style**
  - Amount. Show only the most important values, in order to keep it manageable

    In lectures and articles, comprehensiveness may be preferred, though
  - Alignment. Text left, numbers right
  - Lines. Recommended to use only horizontal lines

    Except for matrices

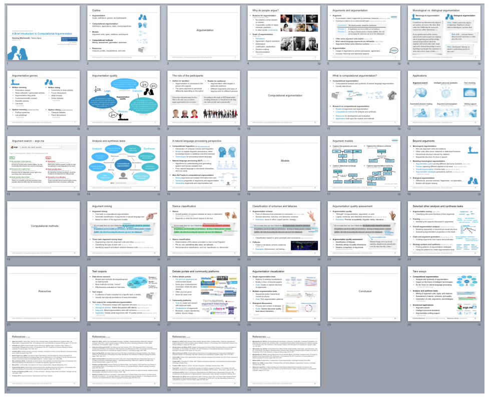
	true	**false**
true	TP	FP
false	FN	TN

- **Tables vs. charts**
  - Prefer tables if exact numbers are important
  - Prefer charts if relative differences should be stressed

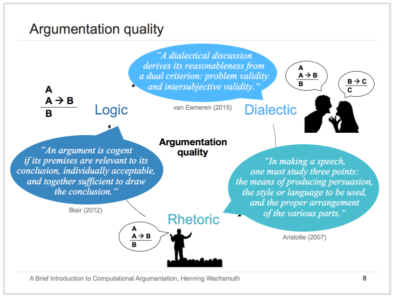


Structure of your slides

- **Overall stucture of presentation**
  - Title slide. Title, authors, maybe date
  - Outline slides. Only for longer talks (as of ~20 slides)
  - Content slides. Your story
  - Conclusion slide(s). Always! Take aways, future work
  - References. Prepare, but only show when asked for



- **Structure of content slides**
  - Header. Clear unique title, should match content of body
    Notice: Titles often not read by the audience
  - Body. Bullet points, figures, tables, etc.
  - Footer. Title, presenter, no date, always page no./progress



Argumentation quality

"A dialectical discussion derives its reasonableness from a dual criterion: problem validity and intersubjective validity." van Eemeren (2015)

"An argument is cogent if its premises are relevant to its conclusion, individually acceptable, and together sufficient to draw the conclusion." Blair (2012)

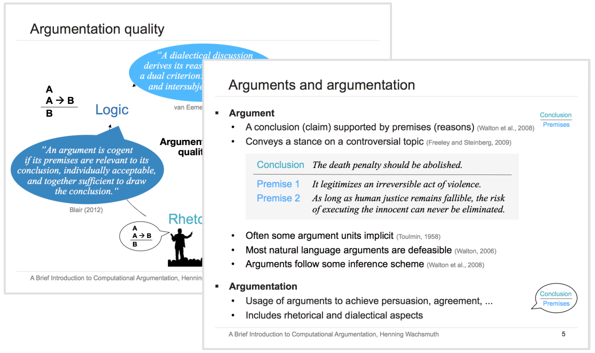
"In making a speech, one must study three points: the means of producing persuasion, the style or language to be used, and the proper arrangement of the various parts." Aristotle (2007)

A Brief Introduction to Computational Argumentation, Henning Wachsmuth 8

- **Space for separation**
  - Leave space between different slide parts
  - Leave some space to slide borders
    Harder to read there + border sometimes clipped

Style of your slides

- **General slide style**
  - Decide what to put on slide and what to say
  - Vary slides to maintain attention
  - Animations only when useful, use consistently
    Avoid playful ones, unless they match your message
  - Clarify what is from you and what from others
    Also see notes on citations below



Argumentation quality

A
A → B
B

Logic

"A dialectical discussion derives its reas a dual criterion: and intersubje

van Eeme

"An argument is cogent if its premises are relevant to its conclusion, individually acceptable, and together sufficient to draw the conclusion."

Blair (2012)

Argumen quali

Rheto

A
A → B
B

A Brief Introduction to Computational Argumentation, Hennin

Arguments and argumentation

- **Argument**
  - A conclusion (claim) supported by premises (reasons) (Walton et al., 2008)
  - Conveys a stance on a controversial topic (Freeley and Steinberg, 2009)

Conclusion	*The death penalty should be abolished.*
Premise 1	*It legitimizes an irreversible act of violence.*
Premise 2	*As long as human justice remains fallible, the risk of executing the innocent can never be eliminated.*

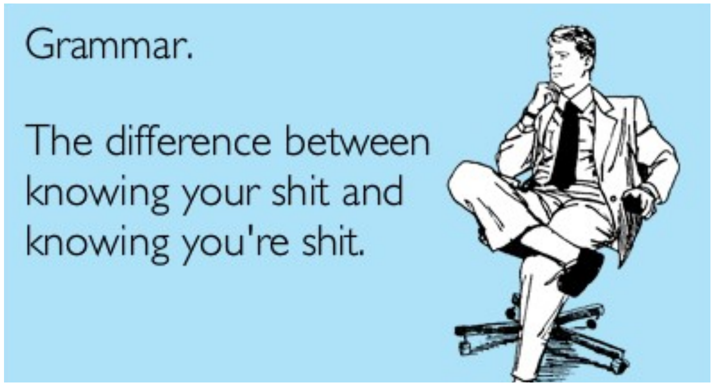
  - Often some argument units implicit (Toulmin, 1958)
  - Most natural language arguments are defeasible (Walton, 2006)
  - Arguments follow some inference scheme (Walton et al., 2008)
- **Argumentation**
  - Usage of arguments to achieve persuasion, agreement, ...
  - Includes rhetorical and dialectical aspects

Conclusion
Premises

Conclusion
Premises

A Brief Introduction to Computational Argumentation, Henning Wachsmuth 5

- **Text style**
  - Avoid grammar and spelling errors
  - Avoid full sentences, rather key points
    Different in lecture talks! (see below)
  - AIA & AUA
    Always introduce acronyms & Avoid unnecessary acronyms


Grammar.
The difference between knowing your shit and knowing you're shit.

https://flickr.com

- **Amount of text**
  - Some say 7x7 — maximum 7 bullet points per slide, 7 words per point
  - I'd rather say 3x3 — 3 top-level points with 3 sub-points

Fonts

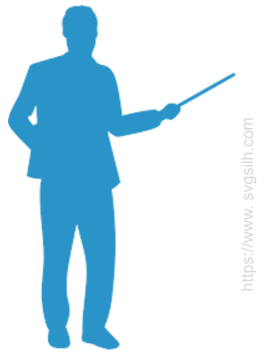
- **Fonts**
  - Sans-serif fonts (Arial, Verdana, ...) much more readable on slides
    Ambiguity ("Ill") speaks against Arial... but Arial available on all machines
  - *Serif* fonts (Times, Garamond, ...) are made for printing
    I use them on slides for example texts only
  - Prefer simple fonts
  - Don't use too narrow fonts just to save space
- **Font size**
  - This text is written in 26 pt — for titles and stressing
  - This text is written in 24 pt
  - This text is written in 21 pt
  - This text is written in 18 pt — minimum for text that should be read
  - This text is written in 16 pt
  - This text is written in 14 pt
  - This text is written in 12 pt — minimum for extra information that may be skipped
  - This text is written in 10 pt
  - This text is written in 8 pt
  - This text is written in 6 pt — maybe for texts that should on purpose not be readable

Talking and timing

- **Giving a talk**
  - Match words on slides, but complement them
  - No pre-phrased sentences
  - Look at audience, speak to everybody
  - Don‘t be *too* formal, but be serious, avoid slang

    Jokes may be nice if you know how to use them
- **Timing**
  - Use your time, but stick with time limit
  - Expect ≥ 2 minutes per (animated) content slide
  - Rule of thumb: Audience can read slide twice
  - Leave time for questions, discussion and breaks
- **Practice your complete talk!**
  - How much time do you need?
  - Does your story work?
  - Can you explain everything well?

max.
135 min.
(+ break)

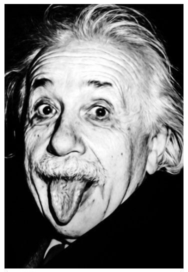


Lecture talks in particular

Specific characteristics of lecture talks

- **Most general hints on talks also hold for lectures**
  See above!
  - Science is storytelling
    Seminar: No scientific break-through expected, rather summarize and discuss
  - Science needs to be understood

https://de.wikipedia.org


https://publicdomainpictures.net

- **What‘s the difference?**
  - A lecture should teach students all basics needed to understand the respective topic.
  - In computer science, slides often replace a ”real“ script.
  - Tutorials are used to practice application or similar.

https://www. svgsilh.com

- **Consequences**
  - Lectures more complete
  - Lectures should be more interactive
  - Lecture slides should be more sound

Style of lectures

- **Lectures more complete**
  - Tell the whole story, avoid gaps in argumentation
  - But: Include only relevant content
  - Don‘t expect too much prior knowledge
  - But: No details on knowledge that can be presupposed
- **Lectures more interactive**
  - Ask for questions from time to time, for example after each ”section“
  - Include interactive parts, to raise attention and interest
  - Proactively check the students' understanding
  - Double-check whether people have understood you
- **Puzzled faces should alert you**
  - Try different ways of explaining
  - Give more details or examples

”**Don‘t** make me **think.**“

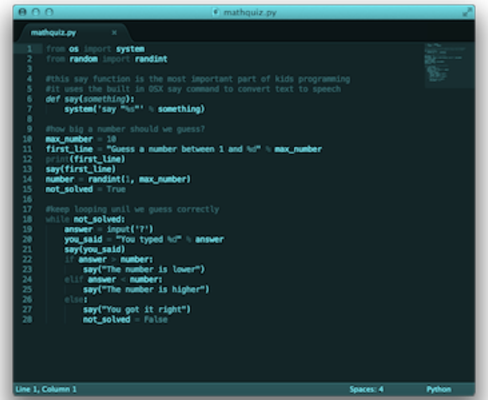
Steve Krug

Style of lecture slides

- **Lecture slides more sound**
  - They need to be more precise than in other talks
  - Formalize concepts where it is needed/helpful for full understanding
  - But: Don‘t make things complex (common misunderstanding)
- **Lecture slides serve as a script**
  - Full sentences often make it easier to avoid misunderstandings
    Still: Keep text short, one statement per sentence.
  - Use explicit discourse markers, such as ”because“
  - Blurring is non-scientific, such as ”It could be...“
- **Technical terms**
  - Introduce terms where needed, but don‘t overformalize
  - Use well-defined terms, AIA & AUA
  - Always use the same term for the same concept (no synonyms!)
    Reader is misled to check whether intentional differences exist.

Tutorials: Hands-on experience

- **Our tutorials are programming tutorials**
  - 90 minutes time. Students should learn to develop computational approaches themselves
  - Schedule. Instructions, programming, discussion
  - Rule of thumb. About 60 minutes for programming



```python
from os import system
from random import randint

#this say function is the most important part of kids programming
#it uses the built in OSX say command to convert text to speech
def say(something):
    system('say "%s"' % something)

#how big a number should we guess?
max_number = 10
first_line = "Guess a number between 1 and %d" % max_number
print(first_line)
say(first_line)
number = randint(1, max_number)
not_solved = True

#keep looping unil we guess correctly
while not_solved:
    answer = input('?')
    you_said = "You typed %d" % answer
    say(you_said)
    if answer > number:
        say("The number is lower")
    elif answer < number:
        say("The number is higher")
    else:
        say("You got it right")
        not_solved = False
```

- **Preparation**
  - Task. Should address some core ideas of the given topic
  - Code. Create a reasonable template, so students can achieve something
    Prepare one solution yourself that the students see/get afterwards
  - Libraries. Use where understanding is not in the focus (e.g., for machine learning)

- **In the tutorial**
  - Instructions. Give instructions in the beginning, explain what you prepared
  - Handouts. May help to give an overview of concepts or similar
  - Interaction. Proactively approach students, check progress and problems
    Give hints, but don't solve the task for them.

Citation

- **Citation**
  - Reference to a bibliographic source
  - We use ACL-style: Author names + year

    Other communities use numbers ([1], ...) or acronyms ([ACW17], ...)

(Ajjour et al., 2017)
(Wachsmuth and Stein, 2017)
Ajjour et al. (2017)

- **What to cite**
  - Any reuse, paraphrase, summary, or translation of content from some source

    Content: Text, figures, and tables
  - Rule of thumb: Always clarify what is from you and what from others

    Also have to cite yourself if you use your own sources
  - Better one citation too much than one too less

- **How to cite**
  - Direct reuse. Always, put in quotes (possibly shorten with [...]), give source

    Example: As Ajjour et al. (2017) point out, unit segmentation is "[...] the splitting of a text into argumentative parts".
  - Other citations. Give source close-by

    Example: Ssegmentation is the first task of an argument mining pipeline (Ajjour et al. 2017)
  - Large text portions. Give source once in the beginning

    Example: These slides are partly based on slides of Prof. Engels and Prof. Becker.

References and Plagiarism

- **List of references**
  - Bibliographical information at end of slides
  - Exactly those references cited in the slides
  - Given information should be complete and consistent

Ajjour et al. (2017). Yamen Ajjour, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth, and Benno Stein. Unit Segmentation of Argumentative Texts. In Proceedings of the Fourth Workshop on Argument Mining, pages 118–128, 2017. http://aclweb.org/anthology/W17-5115

- **Plagiarism**
  - To sell another‘s ideas or expressions as one‘s own
    See en.wikipedia.org/wiki/Plagiarism
  - On purpose or due to lack of giving sources
  - Plagiarism *not* a trivial offense
    In some countries considered as crime
  - Proper citing avoids all plagiarism issues

https://commons.wikimedia.org

- **My former group**
  - Does research on plagiarism detection
  - See the tool picapica www.picapica.org

Sum up

Take aways

- **Literature research**
  - Fundamental part of scientific work
  - Literature varies in quality and suitability
  - Find, read, and organize literature efficiently

- **Scientific presentation**
  - Science is storytelling, needs to be understood
  - Several best practices for content, structure, and style
  - Proper citation is a must
  - Practice oral and written presentation early

- **For your lecture talks**
  - Consider hints in this presentation
  - Don‘t confuse a lecture talk with a seminar talk
  - Develop your own way of presenting

References

- **Several slides reuse content from:**

  - **Engels (2010).** Gregor Engels. Einführung in wissenschaftliches Schreiben und Präsentationstechniken. Presentation within the Seminar ”Information-Driven Software Engineering“. Paderborn, 2010. https://cs.uni-paderborn.de/fileadmin/informatik/fg/dbis/Lehre/ws10_11/PG_IDSE/Dokumente/2010-04-15_Schreiben_Praesentieren.pdf
  - **Becker (2012).** Steffen Becker. Scientific Working. Presentation within the Seminar ”Model Driven Software Engineering with Eclipse. Paderborn, 2010. www.hni.uni-paderborn.de/fileadmin/Fachgruppen/Softwaretechnik/Lehre/Proseminar_Model_Driven_Software_Engineering/ProSem_MDSD_Guidelines.pdf

- **Examples are taken from:**

  - **Ajjour et al. (2017)**. Yamen Ajjour, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth, and Benno Stein. Unit Segmentation of Argumentative Texts. In Proceedings of the Fourth Workshop on Argument Mining, pages 118–128, 2017. http://aclweb.org/anthology/W17-5115
  - **Wachsmuth et al. (2017a).** Henning Wachsmuth, Benno Stein, and Yamen Ajjour. ”PageRank“ for Argument Relevance. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 1116–1126, 2017. http://aclweb.org/anthology/E17-1105