FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

Writing for Computer Science

Prof. Dr. Viktor Leis

Professur für Datenbanken und Informationssysteme

Motivation

“When you understand that nobody wants to read your shit, your mind becomes powerfully concentrated. You begin to understand that writing/reading is, above all, a transaction. The reader donates his time and attention, which are supremely valuable commodities. In return, you the writer must give him something worthy of his gift to you.”
Steven Pressfield

Key Skills for Computer Scientists

1. programming
2. reading
3. giving talks
4. writing

These skills are not independent, but complementary.

Types of Technical Writing

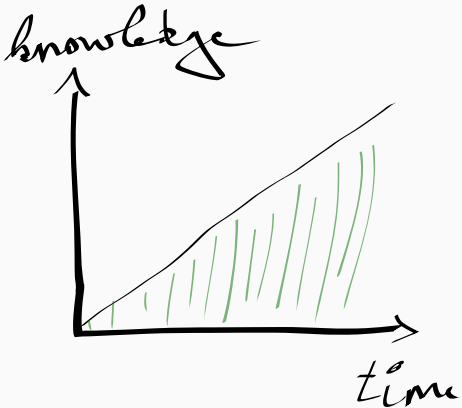
- thesis: Bachelor, Seminar, Master, PhD
- research: research paper, project proposal
- industry: documentation, design document, white paper, website

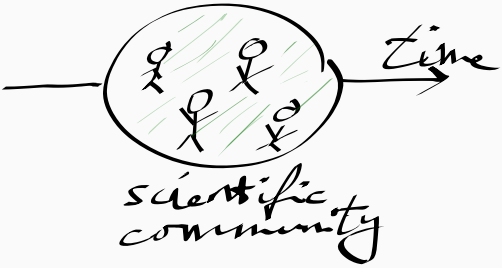
Writing: An Intellectual Superpower

- forces one to make thoughts concrete, organizes thought
- helps debugging ideas
- helps truly understanding concepts
- process of writing shapes ideas and generates new ones
- “Science is science writing; science writing is science”[1]

[1] https://statmodeling.stat.columbia.edu/2019/12/29/science-is-science-writing-science-writing-is-science/

Models Of Scientific Progress





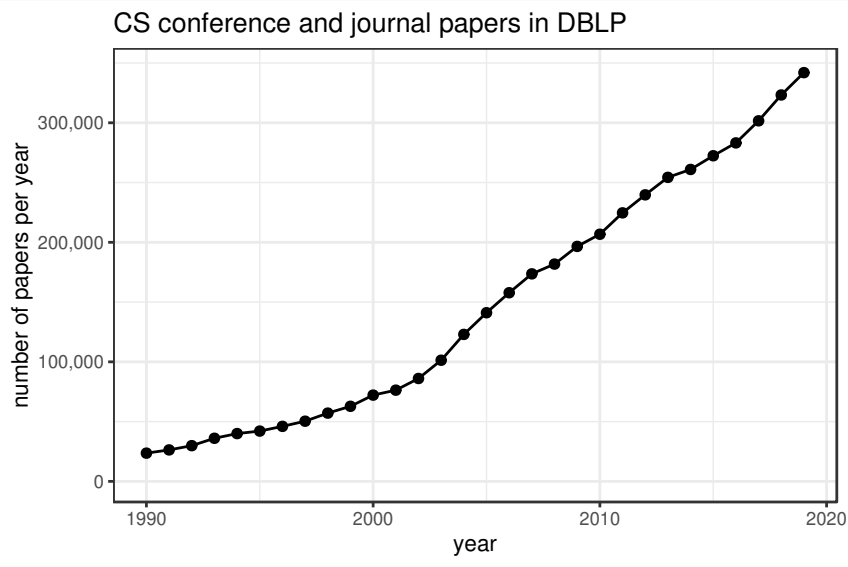
The Role of Good/Bad Writing in Science

- everyone is busy: attention is extremely valuable
- even very good ideas are ignored if they are not written up clearly
- will be reinvented by someone who writes well
- badly-written papers are either or ignored
- non-selective conferences and journals have little impact

Effects of Bad Writing on Readers

1. slow down, reread
2. do not understand
3. get angry
4. stop reading (reject)

Papers, Papers, Papers

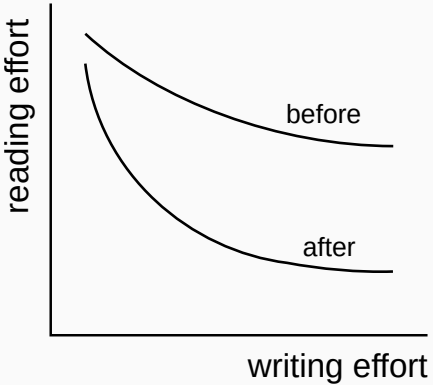


Algorithm For Learning Writing

1. decide that writing matters
2. read a lot (both fiction and non-fiction) to develop an ear for good writing
3. write a lot
4. go to 2.

Why Should One Take This Course?

- writing is hard
- becomes easier with practice, but never effortless
- the goal of this course is to speed up this process



The Process of Writing

Algorithm:

```
function write()
    draft = writeDraft()
    while (quality(draft) < threshold)
        draft = revise(draft)
        ... // if possible, do something else
    return draft
```

Writing The First Draft

- easy to become stuck (“writers block”)
- one approach is to create a first draft quickly
- this first draft is only for you
- to avoid distractions of formatting, it may be beneficial to use a plain text editor
- at this stage, perfection is the enemy of the good

Revising

- early draft should be written quickly
- revising it is the key to get a good result
- most of time should be spent revising
- revising requires extreme concentration, is even more exhausting than programming
- it makes sense to do it every day for a short amount of time, e.g., every morning for 1 hour
- at the end of the process, the paper will likely look totally different
- don’t fall victim to the sunk cost fallacy, “kill your darlings”

Feedback

- the first draft is for oneself
- later drafts can be shown to others to obtain feedback
- most useful feedback will be on first reading
- if reader tells you that he did not understand some part
  - the natural reaction is to explain
  - there is no point in doing that
  - think about how to improve the text based on the input

Know the Reader

- no one cases about you, what you did, your ideas, your feeling
- you must provide **value** to the reader
- to do this you have to know what the reader knows/believes

Plan for the Course (may change)

- The Mechanics of Writing: how to write clear sentences, paragraphs and sections
- Bird's Eye Perspective: how to structure technical material
- tips for English texts (commas, common errors)
- how to create figures, graphs
- tools, workflow: LaTex, git, ggplot, inkscape

Learning By Doing

- like most difficult skills, can only be learned by doing it
- writing and revision homework
- we will discuss solutions
- if you have anything to write this semester, you can use it
- no exam: hand in a text at the end

English vs. German in CS

- examples will be in English
- 80% of this course is applicable to German writing too
- Deutsch als Wissenschaftssprache ist tot
- in cutting-edge companies all writing is in English too

General Writing Guides

- Joseph Williams, Style: Toward Clarity and Grace, Univ. of Chicago Press, 1990
  Actionable advice. Parts of the course are based on this book.
- Steven Pinker, Sense of Style: The Thinking Person's Guide to Writing in the 21st Century, Penguin, 2014
  Another good writing guide from a well-known non-fiction writer.
- William Strunk and E. White, The Elements of Style, Fourth Edition, 1918
  Classic, highly influential style guide, emphasis on brevity, a bit outdated.
- Francis-Noël Thomas and Mark Turner, Clear and Simple as the Truth: Writing Classic Prose, Princeton University Press, 2011 *Essay arguing for "classic style".*
- William Zinsser, On Writing Well: The Classic Guide to Writing Nonfiction, Harper Perennial, 2012

CS-Specific Writing Guides

- Lyn Dupre, BUGS in Writing, Addison-Wesley, 1998
  Easily-digestible tips.
- Justin Zobel, Writing for Computer Science, Springer, Third Edition, 2014
  The title says it all.
- Charles Ling and Qiang Yang, Crafting Your Research Future: A Guide to Successful Master's and Ph.D. Degrees in Science & Engineering, Morgan & Claypool, 2012
  Not just writing, but how to do science. Some questionable advice too.

Larry McEnerney, The Craft of Writing Effectively:
`https://www.youtube.com/watch?v=vtIzMaLkCaM&t=30s`