

# **Presenting your Bachelor Project**

Siegfried Nijssen

# Schedule

- April 20
- May 4
- May 18
- June 1

# What

- Oral presentation of 10 minutes
- 5 minutes for questions
- Parallel sessions
- Your supervisor is preferably present
- Indicate on the website of the bachelor class which day suits you and your supervisor

<http://goo.gl/forms/B81hqlo3W7>

# Outline





- Title, authors
- Introduction: context, motivations, research question
- Methodology
- Results
- Conclusions, next steps

# Content

- Do: Motivate, motivate, motivate and don't forget to motivate.
  - Why this problem is interesting
  - Why this problem is hard
  - Why this solution makes sense
  - Why you made the choices that you made
  - Why an experiment is insightful

# Content

- A good presentation  $\neq$  A good scientific result

	<b>great scientific results</b>	<b>poor scientific results</b>
<b>great communication</b>		
<b>poor communication</b>		

# Pitfalls

- 10 minutes is very short; don't:
  - Try to put many formulas or algorithms in your presentation
  - Try to show off
  - Try to put a lot of full text on your slides
  - Try to describe everything that you did
- Rather:
  - Provide intuitions, for instance using examples
  - Try to understand what your audience is capable of understanding; don't use jargon
  - Use images
  - Make choices – only the most important elements of your work can be in your presentation

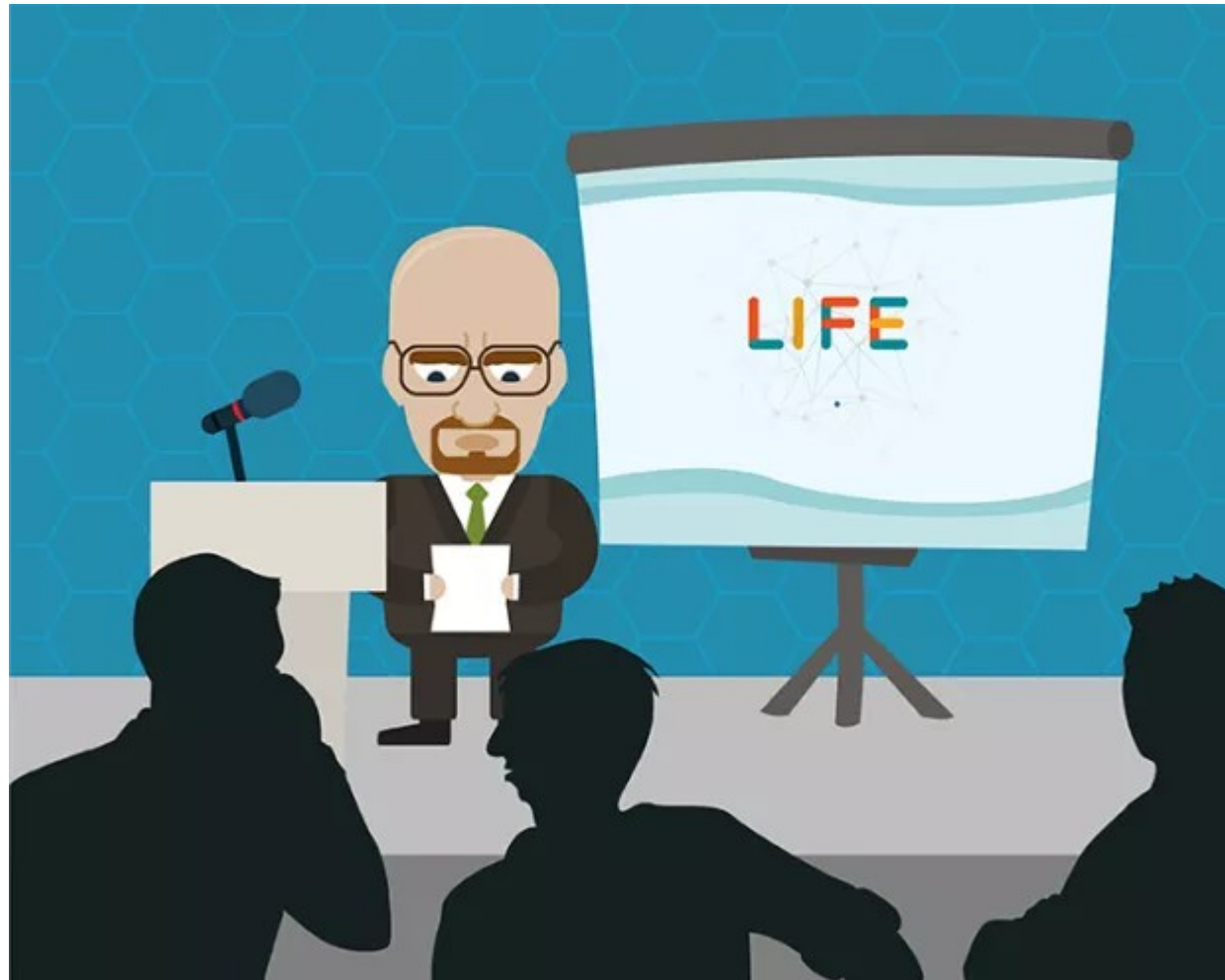
# Presenting

- Good presentation  $\neq$  Good slides
  - Think about what you want to say with your slides
  - Think about your intonation
  - Think about your body language



# Presenting

Make eye contact



# Presenting

Don't fidget



# Presenting

Don't show strange faces



# Presenting

Don't stand like a soldier



# **Writing a Bachelor Thesis in Computer Science**

Siegfried Nijssen

# Technical Matters

# LaTeX

- Recommended approach for writing a thesis
- Template files available on the website of the bachelorklas

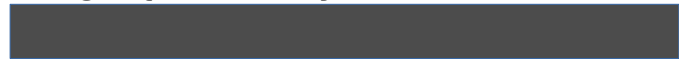
# LaTeX Structure



← (Obfuscated code)

```
\preface
```

```
\begin{abstract}
```



```
\end{abstract}
```

```
\begin{acknowledgements}
```



```
\end{acknowledgements}
```

```
\body
```

```
\listoftables
```

```
\listoffigures
```

```
\chapter{Introduction}
```

```
\section{Motivation}
```

```
\chapter{Definitions}
```

```
\section{Data}
```

```
\appendix
```

```
\bibliographystyle{...}
```

```
\bibliography{...}
```





# References

- There are many acceptable styles

Items are cited: *The L<sup>A</sup>T<sub>E</sub>X Companion* book [Goossens et al., 1993], the Einstein journal paper [Einstein, 1905], and The L<sup>A</sup>T<sub>E</sub>X related items are [Goossens et al., 1993, Knuth, ].

## References

[Einstein, 1905] Einstein, A. (1905). Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]. *Annalen der Physik*, 322(10):891–921.

[Goossens et al., 1993] Goossens, M., Mittelbach, F., and Samarin, A. (1993). *The L<sup>A</sup>T<sub>E</sub>X Companion*. Addison-Wesley, Reading, Massachusetts.

[Knuth, ] Knuth, D. Knuth: Computers and typesetting.

**Important: be consistent!**

# References in LaTeX

Het eerste `\LaTeX`-boek is van de ontwikkelaar Lamport `\cite{lampport94}`.

```
\begin{thebibliography}{XXX99}
\bibitem[Lam94]{lampport94}
Leslie Lamport, \emph{\LaTeX: A Document Preparation
System}. AddisonWesley, Massachusetts, 2nd Edition,
1994.
\end{thebibliography}
```

Source

Het eerste LATEX-boek is van de ontwikkelaar Lamport [Lam94].

## References

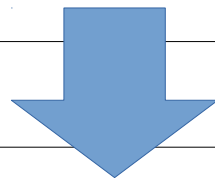
[Lam94] Leslie Lamport, *LATEX: A Document Preparation System*. AddisonWesley, Massachusetts, 2nd Edition, 1994.

Rendered

# References in BibTeX

- BibTeX = textual database system that can be used to generate consistent references

```
@book{lampport94,  
  author      = {Leslie Lamport},  
  title       = {LaTeX: A Document Preparation System},  
  publisher   = {AddisonWesley},  
  year        = 1994,  
  address     = {Massachusetts},  
  edition     = 2  
}
```



Generates

```
\bibitem[Lam94]{lampport94}  
Leslie Lamport, \emph{\LaTeX: A Document Preparation  
System}. AddisonWesley, Massachusetts, 2nd Edition,  
1994.
```

# References in BibTeX

- Steps:
  - Put references in a .bib file (eg. thesis.bib)
  - Cite the references in a .tex file (eg. thesis.tex)
  - Run LaTeX (eg. `latex thesis`)
    - This will generate a PDF/DVI without references, but will also put the identifiers of cited references in a text file (.aux file)
  - Run BibTeX (eg. `bibtex thesis`)
    - This will generate the LaTeX code for the cited references (.bbl file), using the identifiers in the text file (.aux file)
  - Run LaTeX (eg. `latex thesis`)
    - This will generate the PDF/DVI with the references at the end; however, citations are not correct
  - Run LaTeX (eg. `latex thesis`)
    - This will generate the PDF/DVI with the reference at the end; citations will be correct as well

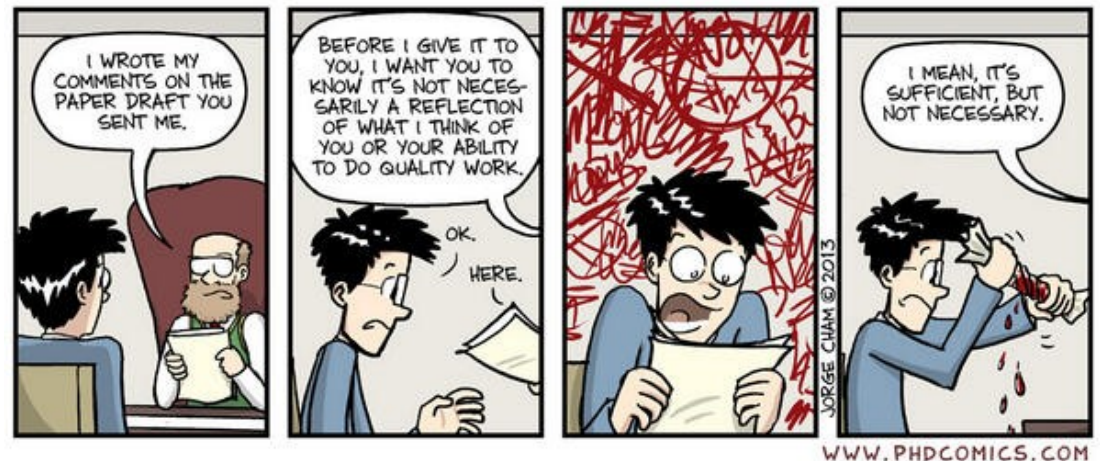
# Where to Get BibTeX Files

- DBLP
- ACM
- IEEE
- Bibliography management software:
  - Jabref
  - CiteULike
  - Mendeley

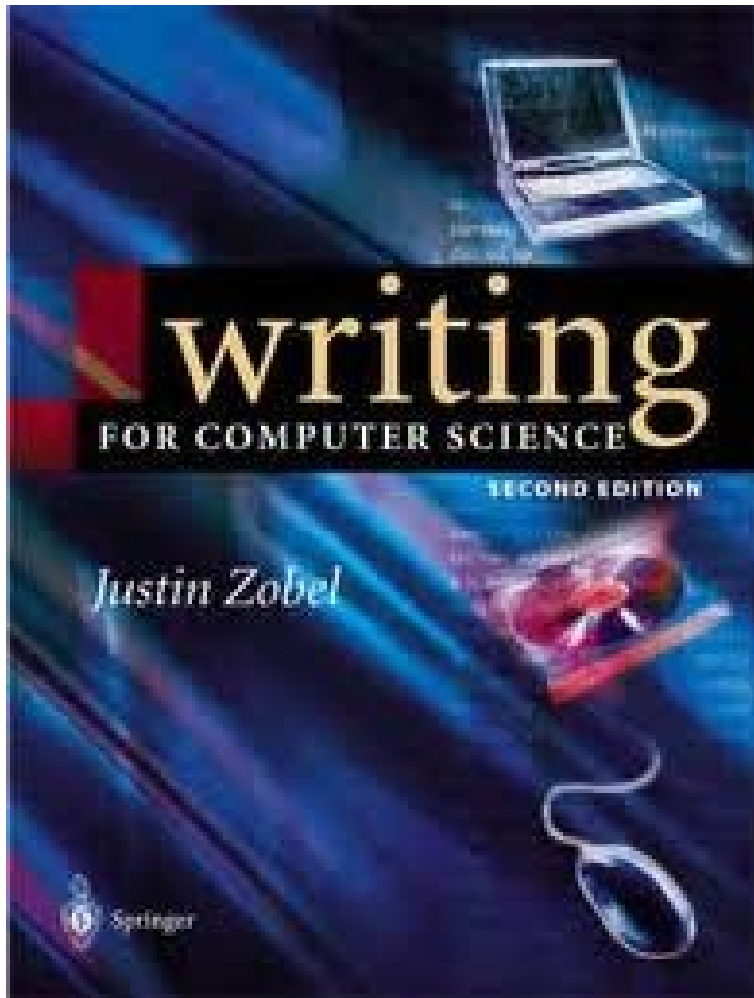
# Content

# How to Write

- Writing a text is not easy for many academics



# Bad Academic Writing is a Problem



## Dart-Throwing Chimp

*Thoughtful analysis or bloviation? Your call.*

### Why Is Academic Writing So Bad? A Brief Response to Stephen Walt

On his *Foreign Policy* blog, Stephen Walt picks up on a *Daily Dish* thread and asks, "Why is academic writing so bad?" He suggests a few reasons but concludes that, for the most part, scholars write poorly on purpose. In his view, bad writing is "a form of academic camouflage designed to shield the author from criticism."

Is this really such a mystery, though? Writing well is hard to do, and it depends in no small part on talent. Like all talents, the ability to write well is probably distributed normally across the population. Most people are mediocre at it, some are really bad, and

AUTHOR



ARCHIVES

Select Month

SUBSCRIBE VIA RSS

## Prospect

Blogs Politics Economics World Science Arts & Books



Prospector

So what, and what next?

### Why academics can't write

by Michael Billig / AUGUST 8, 2013 / 31 COMMENTS

#### Attack of the meaningless nouns



A couple or so years ago, the university where I teach introduced a new way for staff to record information about tutorials, grandly calling it the "Co-Tutor Student Management Relationship System." The name possesses the linguistic features that appeal to managers today: nouns strung together without intervening prepositions, with some of the nouns conveying generally desirable qualities ("relationship," "system"). Despite its pomposity the name



# Differences between Areas

## MODULAR FORMS, ELLIPTIC CURVES AND THE $ABC$ -CONJECTURE

DORIAN GOLDFELD\*

*Dedicated to Alan Baker on the  
occasion of his sixtieth birthday.*

### §1. The $ABC$ -Conjecture.

The  $ABC$ -conjecture was first formulated by David Masser and Joseph Oesterlé (see [Ost]) in 1985. Curiously, although this conjecture could have been formulated in the last century, its discovery was based on modern research in the theory of function fields and elliptic curves, which suggests that it is a statement about ramification in arithmetic algebraic geometry. The  $ABC$ -conjecture seems connected with many diverse and well known problems in number theory and always seems to lie on the boundary of what is known and what is unknown. We hope to elucidate the beautiful connections between elliptic curves, modular forms and the  $ABC$ -conjecture.

**Conjecture ( $ABC$ ).** *Let  $A, B, C$  be non-zero, pairwise relatively prime, rational integers satisfying  $A + B + C = 0$ . Define*

$$N = \prod_{p|ABC} p$$

*to be the squarefree part of  $ABC$ . Then for every  $\epsilon > 0$ , there exists  $\kappa(\epsilon) > 0$  such that*

$$\max(|A|, |B|, |C|) < \kappa(\epsilon) N^{1+\epsilon}.$$

A weaker version of the  $ABC$ -conjecture (with the same notation as above) may be given as follows.

**Conjecture ( $ABC$ ) (weak).** *For every  $\epsilon > 0$ , there exists  $\kappa(\epsilon) > 0$  such that*

$$|ABC|^{\frac{1}{3}} < \kappa(\epsilon) N^{1+\epsilon}.$$

Oesterlé, [Ost] showed that if we define

$$\kappa(\epsilon) = \inf_{\substack{A+B+C=0 \\ (A,B)=1}} \frac{\max(|A|, |B|, |C|)}{N^{1+\epsilon}}$$

\* Supported in part by a grant from the NSF.

The author would like to thank Iris Anshel and Shu-Wu Zhang for many helpful conversations.

## Approximating the Crowd

Şeyda Ertekin · Cynthia Rudin · Haym Hirsh

Received: date / Accepted: date

**Abstract** The problem of “approximating the crowd” is that of estimating the crowd’s majority opinion by querying only a subset of it. Algorithms that approximate the crowd can intelligently stretch a limited budget for a crowdsourcing task. We present an algorithm, “CrowdSense,” that works in an online fashion where items come one at a time. CrowdSense dynamically samples subsets of the crowd based on an exploration/exploitation criterion. The algorithm produces a weighted combination of the subset’s votes that approximates the crowd’s opinion. We then introduce two variations of CrowdSense that make various distributional approximations to handle distinct crowd characteristics. In particular, the first algorithm makes a statistical independence approximation of the labelers for large crowds, whereas the second algorithm finds a lower bound on how often the current sub-crowd agrees with the crowd’s majority vote. Our experiments on CrowdSense and several baselines demonstrate that we can reliably approximate the entire crowd’s vote by collecting opinions from a representative subset of the crowd.

**Keywords** Crowdsourcing · Wisdom of crowds · Labeler quality estimation · Approximating the crowd · Aggregating opinions

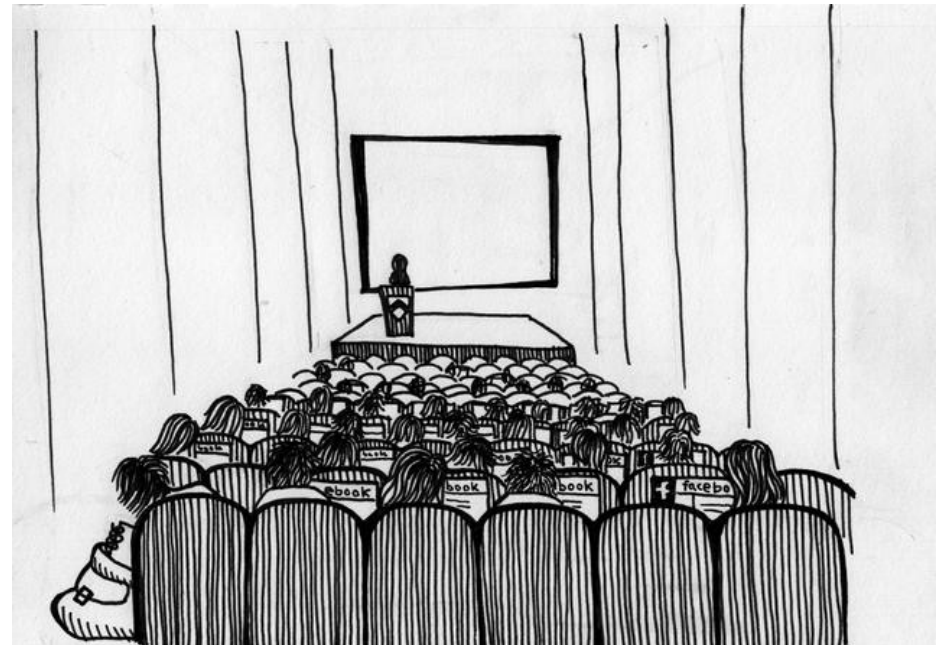
---

Şeyda Ertekin · Cynthia Rudin  
MIT CSAIL, Sloan School of Management, and Center for Collective Intelligence, Massachusetts Institute of Technology, Cambridge, MA  
E-mail: seyda@mit.edu, rudin@mit.edu

Haym Hirsh

# Differences between a Report and a Thesis

- A report is written for 1 or 2 people who know a topic very well
- A thesis is a public document that may be read by many people



# Overview

1. Structure

2. Language

# Structure

# Template for Papers in CS

- Abstract
- Introduction
- Background
  - Related work
  - Definitions and notation
- Main contributions
- Evaluation
  - Experiments
  - Related work
- Conclusions
- References
- Appendices

# ... in Data Mining

- Abstract
- Introduction
- Background
  - Related work
  - Definitions and notation → what kind of data, learning task
- Main contributions → new algorithm
- Evaluation
  - Experiments → speed, accuracy of algorithm
  - Related work → comparison with other algorithms
- Conclusions
- References
- Appendices

# ... in Fundamental CS

- Abstract
- Introduction
- Background
  - definitions earlier model (eg. petrinet)
  - Related work
  - Definitions and notation
- Main contributions
  - proposed changes to the model
- Evaluation
  - Experiments
  - Related work
    - proofs of properties of the new model
- Conclusions
- References
- Appendices

# Structure of a Thesis

- Can mimic that of a paper
  - Chapter 1: Introduction
  - Chapter 2: Background
  - Chapter 3: Definitions
  - ...
  - Chapter n: Conclusions
- Can be similar to a list of papers
  - Chapter 1: Introduction
  - Chapter 2: Basic approach  
introduction, background, definitions, evaluation, conclusions
  - Chapter 3: Extended approach  
introduction, background, definitions, main contributions, evaluation, conclusions
  - ...
  - Chapter n: Conclusions



# Structure of Text

Sections, paragraphs, sentences, ...

Always make sure to tell a story in CS by using this order:

1)WHY

2)WHAT

3)HOW

i.e., **never** describe “HOW” before describing “WHAT” or “WHY”.

# WHY: Motivations

WHY are you doing your project?

WHY is your project interesting and relevant?

WHY are assumptions sensible?

- Real world applications
- Earlier results and their weaknesses
- Motivating examples

# WHAT: Problem Description

- Formal definition of a problem using mathematics
- Specification of the desirable functionality a user interface should have
- Specification of the desirable functionality of a system
- Description of a dataset and the problems it poses

# HOW: Approach

- Description of an algorithm
- Description of a pipeline
- Description of the architecture of a system
- Description of a user interface
- Development of theory

# Repetitions of Why, What, How

- You develop a user interface for task X
- Level 1:
  - What: a user interface that makes it convenient to do X
  - How: iterate designs of a user interface
- Level 2:
  - What: a user interface according to specifications
  - How: description of libraries, etc., used

# Completeness

- A good thesis is self-contained: anyone with a bachelor in computer science should be able to understand it
- You should refer to the literature for any detail that is not strictly relevant to your text

*eg., you usually do not need to repeat a proof given in the literature, unless the reader needs to be familiar with the structure of the proof itself*

# Mode of Working

*Step 1:* Outline: which story you are going to tell?

*Step 2:* Write first text to “fill in” the outline. This text does **not** have to be perfect.

Subsequently: iterate, iterate, iterate, ...

**Language**



# Be Economical

*The volume of information has been rapidly increasing in the past few decades. While computer technology has played a significant role in encouraging the information growth, the latter has also had a great impact on the evolution of computer technology in processing data throughout the years. Historically, many different kinds of databases have been developed to handle information, including the early hierarchical and network models, the relational model, as well as the latest object-oriented and deductive databases. However, no matter how much these databases have improved, they still have their deficiencies. Much information is in textual format. This unstructured style of data, in contrast to the old structured record format data, cannot be managed properly by the traditional database models. Furthermore, since so much information is available, storage and indexing are not the only problems. We need to ensure that relevant information can be obtained upon querying the database.*

# Be Simple

In promulgating your esoteric cogitations, or articulating your superficial sentimentalities and amicable philosophical and psychological observations, beware of platitudinous ponderosities. Let your communications possess a clarified conciseness, a coefficient consistency and a concatenated cogency. Eschew conglomerations of flatulated garrulity, jejune babblement, and asinine affectations. Let your extemporaneous descantings and unpremeditated expatiations have intelligibility and veracious vivacity without rodomontade or thrasonical bombast. Sedulously avoid verbosity and vain vapidness either obscurant or apparent. Shun double entendre, prurient jocosity, and pestiferous profanity.<sup>11</sup>

# Simple Sentences

- One idea per sentence
- Short sentences with simple structure
  - ✗ When the kernel process takes over, that is when in the default state, the time that is required for the kernel to deliver a message from a sending application process to another application process and to recompute the importance levels of these two application processes to determine which one has the higher priority is assumed to be randomly distributed with a constant service rate  $R$ .
  - ✓ When the kernel process takes over, one of its activities is to deliver a message from a sending application process to another application process, and to then recompute the importance levels of these two application processes to determine which has the higher priority. The time required for this activity is assumed to be randomly distributed with a constant service rate  $R$ .

# Simple Sentences

- Do **not** concatenate independent sentences with commas; add words, or split sentences!
- × Our results show that the method scales well, it satisfies the requirements.
- ✓ Our results show that the method scales well. It satisfies the requirements.
- ✓ Our results show that the method scales well and satisfies the requirements.

# Rules for Simple Writing

- Logical organization
- One idea per sentence, paragraph, section
- Short sentences with simple structure
- Short paragraphs
- Short words
- Active tense, not passive tense
- No buzzwords or clichés
- No unnecessary material
- Be specific
- Do not break these rules unless there is a reason to

# Pitfalls

## Artistic impulses

We have already seen, in our consideration of *what is*, that the usual simplified assumptions lead inexorably to a representation that is desirable, because a solution is always desirable; but repugnant, because it is false. And we have presented *what should be*, assumptions whose nature is not susceptible to easy analysis but are the only tenable alternative to ignorance (absence of solution) or a false model (an incorrect solution). Our choice is then Hobson's choice, to make do with what material we have—viable assumptions—and to discover whether the intractable can be teased into a useful form.

# Pitfalls

Superiority complex: showing off

- Mention philosophers that your reader likely doesn't know
  - Use jargon as something the reader should know, while likely he doesn't
- X Analysis of this method is of course a straightforward application of tensor calculus.

# Pitfalls

## Obfuscation

- ✗ Experiments, with the improved version of algorithm as we have described, are the step that confirms our speculation that performance would improve. The previous version of the algorithm is rather slow on our test data and improvements lead to better performance.
- ✗ The status of the system is such that a number of components are now able to be operated.
- ✓ Several of the system's components are working.
- ✗ In respect to the relative costs, the features of memory mean that with regard to systems today disk has greater associated expense for the elapsed time requirements of tasks involving access to stored data.
- ✓ Memory can be accessed more quickly than disk.



# Simple $\neq$ Informal

- The explanation for these results remains an open question.
- We've looked for a good reason for these results, but couldn't think of any.
- We tried approaches X and Y; Y performed better.
- We first tried approach X, but this didn't work. We then tried approach Y, which worked better.

# Example 1

- Er werd voor gekozen om collaborative filtering te gebruiken.
- We hebben ervoor gekozen om collaborative filtering te gebruiken.
- We hebben voor collaborative filtering gekozen.

# Example 2

- Deze keuze leek ons de beste omdat ...
- Deze keuze heeft de volgende voordelen: ...

# Example 3

- Er wordt van de muur weggedraaid.
- De robot draait van de muur weg.

# Example 4

- We hebben de manier waarop de gemeten waarde afhangt van de afstand uitgezet in Figuur 3. In deze figuur kunnen we zien dat de gemeten waarde omgekeerd evenredig is met het kwadraat van de afstand.
- In Figuur 3 wordt de gemeten waarde uitgezet in functie van de afstand. Er kan besloten worden dat de gemeten waarde omgekeerd evenredig is met het kwadraat van de afstand.
- Figuur 3 toont dat de gemeten waarde omgekeerd evenredig is met het kwadraat van de afstand.

# Example 5

- Er werd duidelijk goed nagedacht over de softwarearchitectuur.
- De softwarearchitectuur is goed doordacht.

# Example 6

- In this chapter, we will use the experience we have gained from the research described in the previous chapters to formulate a set of guidelines for researchers who want to build performance prediction models for combinatorial optimisation problems.
- In this chapter, we formulate guidelines for building performance prediction models for combinatorial optimisation problems.

# Example

Om de robots te laten samenwerken, gebruiken we een techniek die collaborative diffusion heet. Collaborative diffusion houdt in dat een soort krachtvelden in de omgeving geconstrueerd worden; van Pac-Man en van onverkende gebieden gaat een aantrekkingskracht uit; van elke ghost gaat een afstotende kracht uit. Dit zorgt ervoor dat de ghosts geneigd zijn om uit elkaars buurt te blijven tijdens het zoeken naar Pac-Man en het verkennen van het doolhof. Het zorgt er ook voor dat de ghosts geneigd zijn om Pac-Man van verschillende kanten te benaderen, wat de kans vergroot dat Pac-Man hierdoor ingesloten geraakt. Voor meer informatie over collaborative diffusion verwijzen we naar Repenning (2006).

Om de robots te laten samenwerken, gebruiken we collaborative diffusion (Repenning, 2006). Hierbij wordt een soort krachtveld berekend waarbij Pac-Man en onverkende gebieden de ghosts aantrekken, en de ghosts zelf elkaar afstoten. Hierdoor blijven de ghosts uit elkaars buurt tijdens het exploreren en hebben ze de neiging om Pac-Man van verschillende kanten te benaderen, wat het exploreren en insluiten efficiënter maakt.