Justin Zobel

Writing for Computer Science

Third Edition

Springer

Writing for Computer Science

Justin Zobel

Writing for Computer Science

Third Edition

Springer

Justin Zobel
Department of Computing
and Information Systems
The University of Melbourne
Parkville
Australia

ISBN 978-1-4471-6638-2 ISBN 978-1-4471-6639-9 (eBook)
DOI 10.1007/978-1-4471-6639-9

Library of Congress Control Number: 2014956905

Springer London Heidelberg New York Dordrecht
© Springer-Verlag London 1997, 2004, 2014
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer-Verlag London Ltd. is part of Springer Science+Business Media (www.springer.com)

Preface

Writing for Computer Science is an introduction to doing and describing research. For the most part the book is a discussion of good writing style and effective research strategies, with a focus on the skills required of graduate research students. Some of the material is accepted wisdom, some is controversial, and some are my opinions. The book is intended to be comprehensive; it is broad rather than deep, but, while some readers may be interested in exploring topics further, for most readers this book should be sufficient.

The first edition of this book was almost entirely about writing. The second edition, in response to reader feedback, and in response to issues that arose in my own experiences as an advisor, researcher, and referee, was additionally about research methods. Indeed, the two topics—writing about and doing research—are not clearly separated: it is a small step from asking *how do I write?* to asking *what is it that I write about?*

In this new edition, the third, I've further expanded the material on research methods, as well as refining and extending the guidance on writing. There is a new chapter, on professional communication beyond academia; the chapters on getting started, reading, reviewing, hypotheses, experiments, and statistics have been expanded and reorganized; and there is additional or new material on many topics, including theses, posters, presentations, literature reviews, measurement and variability, evidence, data, and common failings in papers. Every chapter has had some revision, and reader feedback has again been importance in shaping changes. The references have been removed; with so many excellent, up-to-date reading lists available at the click of a search button, a static list seemed anachronistic. The example slides have been dropped too; there are limits to the advice that can be given on dynamic visual presentations in a printed textbook.

As in the earlier editions, the guidance on writing focuses on research, but is intended to be broadly applicable to general technical and professional communication. Likewise, the guidance on the practice of research has wider lessons; for example, a practitioner trying a new algorithm or explaining to colleagues why one solution is preferable to another should be confident that the arguments are built on robust foundations. And, while this edition has a stronger emphasis on the doing of research than did the first two, no topic has been removed; there is additional material on research, but the guidance on writing has not been taken away. I can no longer describe the book as brief, however!

Since the first edition appeared, there have been many changes in the culture of research, and these continue to develop. Physical paper is slowly vanishing as a publication medium, and traditional publishers are being challenged by a range of open-access models. Academic technical reports, already rare a decade ago, have more or less vanished, while online preprint archives have become a key part of the research ecosystem. It now seems to be rare that a spoken presentation is truly unprofessional, whereas in the 1990s many talks were unendurably awful. The growth in the use of good tools for presentations has been a key factor in this change.

Some cultural changes are less positive. A decade ago, I reported that many talks did not have a clear message and were merely a compilation of clever visuals, and that a well-described algorithm had become a welcome, rare exception; both these trends have persisted. Also, while the globalization of English has brought unquestionable benefits to international communication and collaboration, it means that today many papers are written, refereed, and published without passing through the hands of someone who is skilled in the language, so that even experienced researchers occasionally produce work that is far too hard to understand. The Web provides easy access to literature, but perhaps the necessity of using a library imposed a discipline that is now lacking, as past work appears to be increasingly neglected. Some issues concern the integrity of the scientific enterprise itself, such as the growing number of venues that publish work that doesn't meet even the most rudimentary standards.

The perspectives of all scientists are shaped by the research environments in which they work. My research has involved some theoretical studies, but the bulk of my work has been experimental. I appreciate theoretical work for its elegance, yet find it sterile when it is too detached from practical value. While experimental work can be ad hoc, it can also be deeply satisfying, with the rewards of probing the space of possible algorithms and producing technology that can be applied to the things we do in practice. My perspective on research comes from this background, as does the use of experimental work as examples in this book (an approach that is also justified by the fact that such work is generally easier to outline than is a theoretical contribution). But that doesn't mean that my opinions are simply private biases. They are—I hope!—the considered views of a scientist with experience of different kinds of research.

For this new edition, William Webber and Anthony Wirth redrafted some sections, wrote new text, and helped guide the revisions in areas where I am inexpert; I am particularly grateful for their contributions to the chapters on mathematics, algorithms, experiments, and statistics. These sections now represent a consolidation of views, though I have retained the use of *I* and *my* rather than *we* and *our*. Both William and Tony are long-term colleagues; I've appreciated testing my views against theirs, and I think this book is stronger for it. Another new contributor is

Richard Zanibbi, whose suggestions for additional exercises I have gratefully incorporated.

Many other people helped with this book in one way or another. For the first edition, special thanks are due to Alistair Moffat, who contributed to the chapters on figures, algorithms, editing, writing up, and reviewing. Another notable collaborator has been Paul Gruba, my co-author on our two texts on thesis writing skills, *How To Write A Better Thesis* and its prequel, *How To Write A Better Minor Thesis*. With feedback from friends, colleagues, and readers for over 20 years, the list of people who have influenced this book is large; particular thanks are due to Philip Dart, Gill Dobbie, Michael Fuller, Evan Harris, Ian Shelley, Milad Shokouhi, Saiad Tagaghoghi, James Thom, Rodney Topor, and Hugh Williams. I also thank my research students; the hundreds of students who have participated in my research methods lectures; and the many readers who pointed out mistakes or made helpful suggestions.

Many thanks are due to my editor for the second and third editions, Beverley Ford, for her patience during the procrastination, prevarication, and prevalent preponderance of passivity that led to the long delay between editions. Thanks also to the Springer staff who worked on the final editing and production, in particular James Robinson. And, finally, thanks to my partner, Carolyn, for well lots of stuff.

Melbourne, Australia, September 2014 Justin Zobel

Contents

Chapter 1
Introduction

This writing seemeth to me … not much better than the noise or sound which musicians make while they are in tuning their instruments.

Francis Bacon
The Advancement of Learning

No tale is so good that it can't be spoiled in the telling.

Proverb

Writing plays many roles in science. We use it to record events and clarify our thinking. We use it to communicate to our colleagues, as we explain concepts and discuss our work. And we use it to add to scientific knowledge, by contributing to books, journals, and conference proceedings.

Unfortunately, many researchers do not write well. Bacon's quote given above was made four hundred years ago, yet applies to much science writing today. Perhaps we should not always expect researchers to communicate well; surely the skills required for science and writing are different. But are they? The best science is based on straightforward, logical thinking, and it isn't rich, artistic sentences that we expect in a research paper—we expect readability. A scientist who can conceive of and explore interesting ideas in a rigorous way should be able to use much the same skills to solve the problem of how to explain and present those ideas to other people.

However, many researchers undervalue the importance of clarity, and underestimate the effort required to produce a high-quality piece of writing. Some researchers seem content to write badly, and perhaps haven't considered the impact of poor writing on their readers, and thus on their own careers. A research paper can remain relevant for years or even decades and, if published in a major journal or conference, may be read by thousands of students and researchers. Everyone whose work is affected by a poorly written paper will suffer: ambiguity leads to misunderstanding; omissions frustrate; complexity makes readers struggle to reconstruct the author's intention.

Effort used to understand the structure of a paper or the syntax of its sentences is effort not used to understand its content. And, as the proverb tells us, no tale is so good that it can't be spoiled in the telling. Irrespective of the importance and validity of a paper, it cannot be convincing if it is difficult to understand. The more important

© Springer-Verlag London 2014
J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9_1

the results—or the more startling or unlikely they seem—the better the supporting arguments and their presentation should be. Remember that, while you have months or years to prepare your work, reviewers and examiners often have no more than hours and may have rather less. You need to help them to spend their time well.

For writing about science to be respected, a researcher must have something of value to say. A paper or thesis reports on research undertaken according to the norms of the field, to a standard that persuades a skeptical reader that the results are robust and of interest. Thus the written work rests on a program of activity that begins with interesting questions and proceeds through a sound methodology to clear results.

Few researchers are instinctive writers, and few people are instinctive researchers. Yet it is not so difficult to become a good writer. Those who do write well have, largely, learnt through experience. Inexperienced researchers can produce competent papers by doing no more than follow some elementary steps: create a logical organization, use concise sentences, revise against checklists of possible problems, seek feedback. Likewise, the skills of research must be learnt, and early attempts at investigation and experimentation are often marked by mistakes, detours, and fumbling; but, as for writing, competent work can be produced by appreciating that there is a more or less standard template that can be followed, and then using the template to produce a first research outcome.

Most researchers find that their work improves through practice, experience, and willingness to continue to reflect and learn. This observation certainly applies to me. I've continued to develop as a writer, and today produce text much more quickly—and with better results—than when I wrote the second edition a decade ago. I'm also a better scientist, and, looking back just a few years, am aware of poor research outcomes that are due to mistakes I would not make today. In my experience, most scientists develop a great deal as they proceed through their careers.

Kinds of Publication

Scientific results can be presented in a book, a thesis, a journal article, a paper or extended abstract in a conference or workshop proceedings, or a manuscript. Each kind of publication has its own characteristics. Books—the form of publication that undergraduates are the most familiar with—are usually texts that tend not to contain new results or provide evidence for the correctness of the information they present. The main purpose of a textbook is to collect information and present it in an accessible, readable form, and thus textbooks are generally better written than are papers.

The other forms of publication are for describing the outcomes of new research. A thesis is usually a deep—or even definitive—exploration of a single problem. Journals and conference proceedings consist of contributions that range from substantial papers to extended abstracts. A journal paper is typically an end product of the research process, a careful presentation of new ideas that has been revised (sometimes over several iterations) according to referees' and colleagues' suggestions and criticisms.

A paper or extended abstract in conference proceedings can likewise be an end-product, but conferences are also used to report work in progress. Conference papers are usually refereed, but with more limited opportunities for iteration and revision, and may be constrained by strict length limits. There is no universal definition of "extended abstract", but a common meaning is that the detail of the work is omitted. That is, an extended abstract may review the results of a research program, but may not include enough detail to make a solid argument for the claims.

In contrast to books—which can reflect an author's opinions as well as report on established scientific knowledge—the content of a paper must be defended and justified. This is the purpose of reviewing: to attempt to ensure that papers published in a reputable journal or conference are trustworthy, high-quality work. Indeed, in a common usage a *published paper* is distinguished from a mere *paper* by having been refereed.

A typical research paper consists of the arguments, evidence, experiments, proofs, and background required to support and explain a central hypothesis. In contrast, the process of research that leads to a paper can include uninteresting failures, invalid hypotheses, misconceptions, and experimental mistakes. With few exceptions these do not belong in a paper. While a thesis might be more inclusive, for example if the author includes a critical reflection on how the work developed over the course of a Ph.D., such material would usually be limited to mistakes or failures that are genuinely illuminating. A paper or thesis should be an objective addition to scientific knowledge, not a description of the path that was taken to the result. Thus "style" is not just about how to write, but is also about what to say.

Writing, Science, and Skepticism

Science is a system for accumulating reliable knowledge. Broadly speaking, the process of science begins with speculation, observation, and a growing understanding of some idea or phenomenon. This understanding is used to shape research questions, which in turn are used to develop hypotheses that can be tested by proof or experimentation. The results are described in a paper, which is then submitted for independent review before (hopefully) being published; or the results are described in a thesis that is then submitted for examination.

Writing underpins the whole of the research cycle. A key aspect of writing is that the discipline of stating ideas as logical, organized text forces you to formulate and clarify your thoughts. Concepts and ideas are made concrete; the act of writing suggests new concepts to consider; written material can be systematically discussed and debated with colleagues; and the only effective way to develop complex arguments or threads of reasoning, and evaluate whether they are robust, is to write them down. That is, writing is not the end of the research process, but instead shapes it. Only the styling of a paper, the polishing process, truly takes place after the research is complete.

Thus the ability to write well is a key skill of science. Like many aspects of research, writing can only be thoroughly learnt while working with other researchers. Too often, however, the only help a novice receives is an advisor's feedback on drafts of papers. Such interaction can be far from adequate: many researchers have little experience of writing extended documents, and may be confronting the difficulties of writing in English when it is not their first language. It is not surprising that some researchers struggle. Many are intimidated by writing, and avoid it because describing research is less entertaining than actually doing it. For some advisors, the task of helping a student to write well is not one that comes naturally, and can be a distraction from the day-to-day academic work of research and teaching.

Yet writing defines what we consider to be knowledge. Scientific results are only accepted as correct once they are refereed and published; if they aren't published, they aren't confirmed.[1] Each new contribution builds on a foundation of existing concepts that are known and, within limits, trusted. New research may be wrong or misguided, but the process of reviewing eliminates some work of poor quality, while the scientific culture of questioning ideas and requiring convincing demonstrations of their correctness means that, over time, weak or unsupported concepts are forgotten.

A unifying principle for the scientific culture that determines the value of research is that of *skepticism*. Within science, skepticism is an open-minded approach to knowledge: a researcher should accept claims provisionally given reasonable evidence and given agreement (or at least absence of contradiction) with other provisionally accepted claims. A skeptic seeks the most accurate description or solution that fits the known facts, without concern for issues such as the need to seek favour with authorities, while suspending judgement until decisive information is available. Effective research programs are designed to seek the evidence needed to convince a reasonable skeptic. Absolute skepticism is unsustainable, but credulity—the willingness to believe anything—is pointless, as, without some degree of questioning, it is impossible for knowledge to progress.

Skepticism is key to good science. For an idea to survive, other researchers must be persuaded of its relevance and correctness—not with rhetoric, but in the established framework of a scientific publication. New ideas must be explained clearly to give them the best possible chance of being understood, believed, remembered, and used. This begins with the task of explaining our ideas to the person at the next desk, or even to ourselves. It ends with publication, that is, an explanation of results to the research community. Thus good writing is a crucial part of the process of good science.

Using This Book

There are many good general books on writing style and research methods, but the conventions of style vary from discipline to discipline, and broad guidance on science writing can be wrong or irrelevant for a specific area. Some topics—such as algorithms, mathematics, and research methods for computer science—are not discussed in these books at all.

The role of this book is to help computer scientists with their writing and research. For novices, it introduces the elements of a scientific paper and reviews a wide range

[1] Which is why codes of scientific conduct typically require that scientists not publicize their discoveries until after the work has been refereed.

of issues that working researchers need to consider. For experienced researchers, it provides a reference point against which they can assess their own views and abilities, and is an exposure to wider cultures of research. This book is also intended to encourage reflection; some chapters pose questions about research that a responsible researcher should address. Nobody can learn to write or become a researcher just by reading this book, or indeed any book. To become competent it is necessary to practice, that is, to do research and write it up in collaboration with experienced researchers. However, familiarity with the elements of writing and research is essential in scientific training.

Style is in some respects a matter of taste. The advice in this book is not a code of law to be rigidly obeyed; it is a collection of guidelines, not rules, and there are inevitably situations in which the "correct" style will seem wrong. But generally there are good reasons for writing in a certain way. Almost certainly you will disagree with some of the advice in this book, but exposure to another opinion should lead you to justify your own choice of style, rather than by habit continue with what may be poor writing. A good principle is: By all means break a rule, but have a good reason for doing so.

Most computer scientists can benefit from reading a book about writing and research. This book can be used as the principal text for a senior research methods subject, or for a series of lectures on the practice of research. Such a subject would not necessarily follow this book chapter by chapter, but instead use it as a resource. In my own teaching of research methods, lectures on writing style seem to work best as introductions to the key topics of good writing; talking students through the detailed advice given here is less effective than getting them to read the book while they write and undertake research for themselves. That said, for a range of topics—figures, algorithms, presentations, statistics, reading and reviewing, drafting, ethics, and experimentation, for example—the relevant chapter can be used as the basis of one or two lectures.

This book covers the major facets of writing and experimentation for research in computer science:

- Commencing a research program, including getting started on the research and the writing (Chap. 2), reading and reviewing (Chap. 3), and principles of hypotheses, research questions, and evidence (Chap. 4).
- Organization of papers and theses, and the practice of writing (Chap. 5).
- Good writing, including writing style (Chaps. 6–8), mathematical style (Chap. 9), presentation of algorithms (Chap. 10), design of figures and graphs (Chap. 11), expert writing for other professional contexts (Chap. 12), and final editing (Chap. 13).
- Research methodology, including experimentation (Chap. 14) and statistical principles (Chap. 15).
- Presentations, including talks and posters (Chap. 16).
- Ethics (Chap. 17).

There are also exercises to help develop writing and research skills.

If you are new to research, Chaps. 2–5 may be the right place to begin. Note too that much of the book is relevant to writing in computer science in general, in particular Chaps. 6–13. While the examples and so on are derived from research, the lessons are broader, and apply to many of the kinds of writing that professionals have to undertake.

This book has been written with the intention that it be browsed, not memorized or learnt by rote. Read through it once or twice, absorb whatever advice seems of value to you, then consult it for specific problems. There are checklists to be used as a reference for evaluating your work, at the ends of Chaps. 2, 4, 5, and 12–17, and, to some extent, all of the chapters are composed of lists of issues to check.

Some readers of this book will want to pursue topics further. There are areas where the material is reasonably comprehensive, but there are others where it is only introductory, and still others where I've done no more than note that a topic is important. For most of these, it is easy to find good resources on the Web, which is where I recommend that readers look for further information on, for example, statistical methods, human studies and human ethics, and the challenges that are specific to authors whose first language is not English.

Earlier editions of this book included bibliographies. These rapidly dated, and, with many good reading lists online—and new materials appearing all the time—I suggest that readers search for texts and papers on topics of interest, using the online review forums as guides. There are many home pages for research methods subjects, on research in general and in the specific context of computing, where up-to-date readings can be found.

Spelling and Terminology

British spelling is used throughout this book, with just a couple of quirks, such as use of "program" rather than "programme". American readers: There is an "e" in "judgement" and a "u" in "rigour"—within these pages. Australian readers: There is a "z" in "customize". These are choices, not mistakes.

Choice of terminology is less straightforward. An undergraduate is an undergraduate, but the American graduate student is the British or Australian postgraduate. The generic "research student" is used throughout, and, making arbitrary choices, "thesis" rather than "dissertation" and "Ph.D." rather than "doctorate". The academic staff member (faculty in North America) who works with—"supervises"—a research student is, in this book, an "advisor" rather than a "supervisor". Collectively, these people are "researchers" rather than "scientists"; while "computer scientists" are, in a broad sense, not just researchers in the discipline in computer science but people who are computational experts or practitioners. Researchers write articles, papers, reports, theses, extended abstracts, and reviews; in this book, the generic term for these forms of research writing is a "write-up", while "paper" is used for both refereed publications and for work submitted for reviewing, and, sometimes, for theses too.

Some of the examples are based on projects I've been involved in. Most of my research has been collaborative; rather than use circumlocutions such as "my colleagues and I", or "together with my students", the simple shorthand "we" is used to indicate that the work was not mine alone. Many of the examples of language use are drawn from other people's writing; in some cases, the text has been altered to disguise its origin.

Chapter 2
Getting Started

Science is more than a body of knowledge; it is a way of thinking.
Carl Sagan
The Demon-Haunted World

There are as many scientific methods as there are individual scientists.
Percy W. Bridgman
On "Scientific Method"

There are many ways in which a research project can begin. It may be that a conversation with a colleague suggested interesting questions to pursue, or that your general interest in a topic was crystallized into a specific investigation by something learnt in a seminar, or that enrollment in a research degree forced you to identify a problem to work on. Then definite aims are stated; theories are developed or experiments are undertaken; and the outcomes are written up.

The topic of this chapter is about getting started: finding a question, working with an advisor, and planning the research. The perspective taken is a practical one, as a working scientist: What kinds of stages and events does a researcher have to manage in order to produce an interesting, valid piece of work? This chapter, and Chaps. 3, 4, 14, and 15, complement other parts of the book—which are largely on the topic of how research should be described—by considering how the content of paper is arrived at.

Thus this chapter concerns the first of the steps involved in doing a research project, which broadly are:

- Formation of a precise *question*, the answer to which will satisfy the aim of the research.
- Development of a detailed *understanding*, through reading and critical analysis of scientific literature and other resources.
- Gathering of *evidence* that relates to the question, through experiment, analysis, or theory. These are intended to support—or disprove—the hypothesis underlying the question.
- Linking of the question and evidence with an *argument*, that is, a chain of reasoning.
- Description of the work in a publication.

© Springer-Verlag London 2014

J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9_2

Learning to do research involves acquisition of a range of separate skills. It takes experience to see these skills as part of a single integrated "process of research". That is, many people learn to be researchers by working step by step under supervision; only after having been through the research process once or twice does the bigger picture become evident.

Some newcomers try to pursue research as if it were some other kind of activity. For example, in computer science many research students see experimentation as a form of software development, and undertake a research write-up as if they were assembling an essay, a user manual, or software documentation. Part of learning to be a scientist is recognition of how the aims of research differ from those of coursework.

A perspective on research is that it is the process that leads to papers and theses, because these represent our store of accepted scientific knowledge. Another perspective is that it is about having impact; by creating new knowledge, successful research changes the practices and understandings of other scientists. Our work must be adopted in some way by others if it is to be of value. Thus another part of learning to be a scientist is coming to understand that publication is not an end in itself, but is part of an ongoing collaborative enterprise.

Beginnings

The origin of a research investigation is typically a moment of insight. A student attending a lecture wonders why search engines do not provide better spelling correction. A researcher investigating external sorting is at a seminar on file compression, and ponders whether one could be of benefit to the other. An advisor is frustrated by network delays and questions whether the routing algorithm is working effectively. A student asks a professor about the possibility of research on evaluation of code reliability; the professor, who hadn't previously contemplated such work, realises that it could build on recent advances in type theory. Tea-room arguments are a rich source of seed ideas. One person is idly speculating, just to make conversation; another pursues the speculation and a research topic is created. Or someone claims that a researcher's idea is unworkable, and a listener starts to turn over the arguments. What makes it unworkable? How might those issues be addressed?

This first step is a subjective one: to choose to explore ideas that seem likely to succeed, or are intriguing, or have the potential to lead to something new, or that contradict received wisdom. At the beginning, it isn't possible to know whether the work is novel or will lead to valuable results; otherwise there would be no scope for research. The final outcome is an objective scientific report, but curiosity and guesswork are what establish research directions.

It is typically at this stage that a student becomes involved in the research. Some students have a clear idea of what they want to pursue—whether it is feasible, rational, or has research potential is another matter—but the majority are in effect shopping for a topic and advisor. They have a desire to work on research and to be creative, perhaps without any definite idea of what research is. They are drawn by a particular area or problem, or want to work with a particular individual. Students may talk through a range of possible projects with several alternative advisors before making a definite choice and starting to work on a research problem in earnest.

Shaping a Research Project

How a potential research topic is shaped into a defined project depends on context. Experienced scientists aiming to write a paper on a subject of mutual interest tend to be fairly focused: they quickly design a series of experiments or theoretical goals, investigate the relevant literature, and set deadlines.

For students, doing a research project additionally involves training, which affects how the work proceeds. Also, for a larger research program such as a Ph.D., there are both short-term and long-term goals: short-term goals include the current specific explorations, which may be intended to lead to an initial research paper; the long-term goals are the wider investigation that will eventually form the basis of the student's thesis.

At the beginning of a research program, then, you need to establish answers to two key questions. First, what is the broad problem to be investigated? Second, what are the specific initial activities to undertake and outcomes to pursue? Having clear short-term research goals gives shape to a research program. It also gives the student training in the elements of research: planning, reading, programming, testing, analysis, critical thinking, writing, and presentation.

For example, in research in the 1990s into algorithms for information retrieval, we observed that the time to retrieve documents from a repository could be reduced if they were first compressed; the cost of decompression after retrieval was outweighed by savings in transfer times. A broad research problem suggested by this topic is whether compression can be of benefit within a database even if the data is stored uncompressed. Pursuing this problem with a research student led to a specific initial research goal: given a large database table that is compressed as it is read into memory, is it possible to sort it more rapidly than if it were not compressed at all? What kinds of compression algorithm are suitable? Success in these specific explorations leads to questions such as, where else in a database system can compression be used? Failure leads to questions such as, under what conditions might compression be useful?

When developing a topic into a research question, it is helpful to explore what makes the topic interesting. Productive research is often driven by a strong motivating example, which also helps focus the activity towards useful goals. It can be easy to explore problems that are entirely hypothetical, but difficult to evaluate the effectiveness of any solutions. Sometimes it is necessary to make a conscious decision to explore questions where work can be done, rather than where we would like to work; just as medical studies may involve molecular simulations rather than real patients, robotics may involve the artifice of soccer-playing rather than the reality of planetary exploration.

In choosing a topic and advisor, many students focus on the question of "is this the most interesting topic on offer?", often to the exclusion of other questions that are equally important. One such question is "is this advisor right for me?" Students and advisors form close working relationships that, in the case of a Ph.D., must endure for several years. The student is typically responsible for most of the effort, but the intellectual input is shared, and the relationship can grow over time to be a partnership of equals. However, most relationships have moments of tension, unhappiness, or disagreement. Choosing the right person—considering the advisor as an individual, not just as a respected researcher—is as important as choosing the right topic. A charismatic or famous advisor isn't necessarily likeable or easy to work with.

The fact that a topic is in a fashionable area should be at most a minor consideration; the fashion may well have passed before the student has graduated. Some trends are profound shifts that have ongoing effects, such as the opportunities created by the Web for new technologies; others are gone almost before they arrive. While it isn't necessarily obvious which category a new trend belongs in, a topic should not be investigated unless you are confident that it will continue to be relevant.

Another important question is, is this project at the right kind of technical level? Some brilliant students are neither fast programmers nor systems experts, while others do not have strong mathematical ability. It is not wise to select a project for which you do not have the skills or that doesn't make use of your strengths.

A single research area can offer many different kinds of topic. Consider the following examples of strengths and topics in the area of Web search:

Statistical. Identify properties of Web pages that are useful in determining whether they are good answers to queries.

Mathematical. Prove that the efficiency of index construction has reached a lower bound in terms of asymptotic cost.

Analytical. Quantify bottlenecks in query processing, and relate them to properties of computers and networks.

Algorithmic. Develop and demonstrate the benefit of a new index structure.

Representational. Propose and evaluate a formal language for capturing properties of image, video, or audio to be used in search.

Behavioural. Quantify the effect on searchers of varying the interface.

Social. Link changes in search technology to changes in queries and user demographics.

As this list illustrates, many skills and backgrounds can be applied to a single problem domain.

An alternative perspective on the issue of how to choose a topic is this: most projects that are intellectually challenging are interesting to undertake; agonizing over whether a particular option is *the* project may not be productive. However, it is also true that some researchers only enjoy their work if they can identify a broader value: for example, they can see likely practical outcomes. Highly speculative projects leave some people dissatisfied, while others are excited by the possibility of a leap into the future.

When evaluating a problem, a factor to consider is the barrier to entry, that is, the knowledge, infrastructure, or resources required to do work in a particular field. Sometimes it just isn't possible to pursue a certain direction, because of the costs, or because no-one in your institution has the necessary expertise. Another variant of the same issue is the need for a codebase, or experience in a codebase; if investigation of a certain query optimization problem means that you need to understand and modify the source code for a full-strength distributed database system, then possibly the project is beyond your reach.

As research fields mature, there is a tendency for the barrier to entry to rise: the volume of background knowledge a new researcher must master is increased, the scope for interesting questions is narrowed, the straightforward or obvious lines of investigation have been explored, and the standard of the baselines is high. If a field is popular or well-developed, it may make more sense to explore other directions.

Project scale is a related issue. Some students are wildly ambitious, entering research with the hope of achieving something of dramatic significance. However, major breakthroughs are by definition rare—otherwise, they wouldn't be major—while, as most researchers discover, even a minor advance can be profoundly rewarding. Moreover, an ambitious project creates a high potential for failure, especially in a shorter-term project such as a minor thesis. There is a piece of folklore that says that most scientists do their best work in their Ph.D. This is a myth, and is certainly not a good reason for tackling a problem that is too large to resolve.

Most research is to some extent incremental: it improves, repairs, extends, varies, or replaces work done by others. The issue is the magnitude of the increment. A trivial step that does no more than explore an obvious solution to a simple problem—a change, say, to the fields in a network packet to save a couple of bits—is unlikely to be worth investigating. There needs to be challenge and the possibility of unexpected discovery for research to be interesting.

For a novice researcher, it makes sense to identify outcomes that can clearly be achieved; this is research training, after all, not research olympics. A principle is to pursue the smallest question that is interesting. If these outcomes are reached early on, it should be straightforward, in a well-designed project, to move on to more challenging goals.

Some research is concerned with problems that appear to be solved in commercial or production software. Often, however, research on such problems can be justified. In a typical commercial implementation the task is to find a workable solution, while in research the quality of that solution must be measured, and thus work on the same problem that produces similar solutions can nonetheless have different outcomes. Moreover, while it is in a company's interests to claim that a problem is solved by their technology, such claims are not easily verified. In some cases, investigation of a problem for which there is already a commercial solution can be of as much value as investigation of a problem of purely academic interest.

Research Planning

Students commencing their first research project are accustomed to the patterns of undergraduate study: attending lectures, completing assignments, revising for exams. Activity is determined by a succession of deadlines that impose a great deal of structure.

In contrast, a typical research project has just one deadline: completion. Administrative requirements may impose some additional milestones, such as submission of a project outline or a progress report, but many students (and advisors) do not take these milestones seriously. However, having a series of deadlines is critical to the success of a project. The question then is, what should these deadlines be and how should they be determined?

Some people appear to plan their projects directly in terms of the aspects of the problem that attracted them in the first place. For example, they download some code or implement something, then experiment, then write up. A common failing of this approach to research is that each stage can take longer than anticipated, the time for write-up is compressed, and the final report is poor. Yet the write-up is the only part of the work that survives or is assessed. Arguably, an even more significant failing is that the scientific validity of the outcomes can be compromised. It is a mistake, for example, to implement a complete system rather than ask what code is needed to explore the research questions.

A strong approach to the task of defining a project and setting milestones is to explicitly consider what is needed at the end, then reason backwards. The final thing required is the write-up in the form of a thesis, paper, or report; so you need to plan in terms of the steps necessary to produce the write-up. As an example, consider research that is expected to have a substantial experimental component; the write-up is likely to involve a background review, explanations of previous and new algorithms, descriptions of experiments, and analysis of outcomes. Completion of each of these elements is a milestone.

Continuing to reason backwards, the next step is to identify what form the experiments will take. Chapter 14 concerns experiments and how they are reported, but prior to designing experiments the researcher must consider how they are to be used. What will the experiments show, assuming the hypothesis to be true? How will the results be different if the hypothesis is false? That is, the experiments are an evaluation of whether some hypothesised phenomenon is actually observed. Experiments involve data, code, and some kind of platform. Running of experiments requires that all three of these be obtained, and that skeptical questions be asked about them: whether the data is realistic, for example.

Experiments may also involve users. Who will they be? Is ethics clearance required? Computer scientists, accustomed to working with algorithms and proofs, are often surprised by how wide-ranging their university ethics requirements can be.

Many research activities do not have an experimental component, and instead concern principles, or fresh analysis of data, or qualitative interpretation of a case study, or a comparative reflection, or any of a wide range of other kinds of work.

However, milestones can always be identified, because (obviously, I hope!) any substantial project can be meaningfully described as a collection of smaller activities.

Two points here are worth emphasizing. First, while the components of a research project should be identified in advance, they do not necessarily have to be completed in turn. Second, we should plan research with the following attitude: what evidence must we collect to convince a skeptical reader that the results are correct? A successful research outcome rests on finding a good answer to this question.

Having identified specific goals, another purpose of research planning is to estimate the dates at which milestones should be reached. One of the axioms of research, however, is that everything takes longer than planned for.[1] A traditional research strategy is to first read the literature, then design, then analyse or implement, then test or evaluate, then write up. A more effective strategy is to overlap these stages as much as possible. You should begin the implementation, analysis, and write-up as soon as it is reasonable to do so.

For the long-term research activity of a Ph.D., there are other considerations that become significant. A typical concern in the later stages of a Ph.D. is whether enough research has yet been done, or whether additional new work needs to be undertaken. Often the best response to this question is to write the thesis. Once your thesis is more or less complete, it should be easy to assess whether further work is justified. Doing such additional work probably involves filling a well-defined gap, a task that is much better defined than that of fumbling around for further questions to investigate.

Thus, rather than working to a schedule of long-term timelines that may be unrealistic, be flexible. Adjust the work you are doing on a day-to-day basis—pruning your research goals, giving more time to the writing, addressing whatever the current bottleneck happens to be—to ensure that you are reaching overall aims.

Students and Advisors

Advisors are powerful figures in their students' lives, and every student–advisor relationship is different. Some professors at the peak of their careers still have strong views—often outrage or amazement—about their own advisors, despite many years of experience on the other side of the fence. Tales include that of the student who saw his advisor twice, once to choose a topic and once to submit; and that of the advisor who casually advised a student to "have another look at some of those famous open problems". Thankfully these are rare exceptions.

The purpose of a research program—a Ph.D., masters, or minor thesis—is for the university to provide a student with research training, while the student demonstrates the capability to undertake research from conception to write-up, including such skills as working independently and producing novel, critical insights. A side-benefit is that the student, often with the advisor, should produce some publishable research. There

[1] Even after taking this axiom into account.

are a range of approaches to advising that achieve these aims, but they are all based on the strategy of learning while doing.

Some advisors, for example, set their students problems such as verifying a proof in a published paper and seeing whether it can be applied to variants of the theorem, thus, in effect, getting the student to explore the limits at which the theorem no longer applies. Another example is to attempt to confirm someone else's results, by downloading code or by developing a fresh implementation. The difficulties encountered in such efforts are a fertile source of research questions. Other advisors immediately start their students on activities that are expected to lead to a research publication. It is in this last case that the model of advising as apprenticeship is most evident.

Typically, in the early stages the advisor specifies each small step the student should take: running a certain experiment, identifying a suitable source of data, searching the literature to resolve a particular question, or writing one small section of a proposed paper. As students mature into researchers, they become more independent, often by anticipating what their advisors will ask, while advisors gradually leave more space for their students to assert this independence. Over time, the relationship becomes one of guidance rather than management.

The trade-offs implicit in such a relationship are complex. One is the question of authorship of work the student has undertaken, as discussed in Chap. 17. Another is the degree of independence. Advisors often believe that their students are either demanding or overconfident; students, on the other hand, can feel either confined by excessive control or at a loss due to being expected to undertake tasks without assistance. The needs of students who are working more or less alone may be very different to those of students who are part of an extended research group.

An area where the advisor's expertise is critical is in scoping the project. It needs to stand sufficiently alone from other current work, yet be relevant to a group's wider activities. It should be open enough to allow innovation and freedom, yet have a good likelihood of success. It should be close enough to the advisor's core expertise to allow the advisor to verify that the work is sufficiently novel, and to verify that the appropriate literature has been thoroughly explored. The fact that an advisor finds a topic interesting does not by itself justify asking a student to work on it. Likewise, a student who is keen on a topic must consider whether competent supervision is available in that area.

Advisors can be busy people. Prepare for your meetings—bring tables of results or lists of questions, for example. Be honest; if you are trying to convince your advisor that you have completed some particular piece of work, then the work should have been done. Advisors are not fools. Saying that you have been reading for a week sounds like an excuse; and, if it is true, you probably haven't spent your time effectively.

The student–advisor relationship is not only concerned with research training, but is a means for advisors to be involved in research on a particular topic. Thus students and advisors often write papers together. At times, this can be a source of conflict, when, for example, an advisor wants a student to work on a paper while the student wants to make progress on a thesis. On the other hand, the involvement of the advisor—and the incentive for the advisor to take an active role—means that the research is undertaken as teamwork.

Over the years I have noticed that there are several characteristics that are shared by successful research students. First, they show a willingness to read widely, to explore the field broadly beyond their specific topic, to try things out, and to generally take part in the academic community. Second, they have the enthusiasm to develop their interest in some area, and then ask for advice on how that interest can be turned into a thesis project. Third, they have the ability and persistence to undertake a detailed (and even gruelling) investigation of a specific facet of a larger topic. Fourth, they take the initiative in terms of what needs to be done and how to present it, and gradually assume responsibility for all aspects of the research. Fifth, they are systematic and organized, and understand the need for rigour, discipline, stringency, quality, and high standards. Sixth, they actively reflect on habits and working practices, and seek to improve themselves and overcome their limitations and knowledge gaps. Seventh, their work *looks* plausible; it has the form and feel of high-quality published papers. Last, they have the strength to keep working despite some significant failed or unsuccessful activity; in a Ph.D., loss of months of work is not unusual.

Note that neither "brilliance" nor "genius" is in this list. Intellectual capacity is important, but many bright people do not become outstanding Ph.D. students—sometimes, because they underestimate the challenge of extended study. Indeed, I've supervised several students whose previous academic record was uninspiring but who nonetheless produced a strong thesis, in particular because they were persistent and resilient enough to pursue their work despite setbacks and obstacles.

A "Getting Started" Checklist

- Is your proposed topic clearly a research activity? Is it consistent with the aims and purposes of research?
- How is your project different from, say, software development, essay writing, or data analysis?
- In the context of your project, what are the area, topic, and research question? (How are these concepts distinct from each other?)
- Is the project of appropriate scale, with challenges that are a match to your skills and interests? Is the question narrow enough to give you confidence that the project is achievable?
- Is the project distinct from other active projects in your research group? Is it clear that the anticipated outcomes are interesting enough to justify the work?
- Is it clear what skills and contributions you bring to the project, and what will be contributed by your advisor? What skills do you need to develop?
- What resources are required and how will you obtain them?
- What are the likely obstacles to completion, or the greatest difficulties? Do you know how these will be addressed?

- Can you write down a road map, with milestones, that provides a clear path to the anticipated research outcomes?
- Do you and your advisor have an agreed method for working together, with a defined schedule of meetings?

Chapter 3
Reading and Reviewing

You know, we have little bits of understanding, glimpses, a little bit of light here and there, but there's a tremendous amount of darkness.

Noam Chomsky

And diff'ring judgements serve but to declare, That truth lies somewhere, if we knew but where.

William Cowper
Hope

The more that you read, the more things you will know. The more that you learn, the more places you'll go.

Dr. Seuss
I Can Read With My Eyes Shut!

A novice researcher can believe that the doing of research is primarily about investigation—running experiments, developing theory, or doing analysis. With experience, though, researchers discover the importance of *developing an understanding*. It has been argued that many experimental researchers do their best work after they have been in a field for five years or more, because it takes time to acquire a deep, thorough appreciation of the area, and of existing knowledge and its limitations. To acquire this understanding, you need to become an effective reader of research papers.

A successful reader can identify the contributions and value of a paper, while recognizing its flaws; and uses critical scrunity to identify the extent to which the flaws in a paper are serious. This reading then informs new work, directly as a source of knowledge and indirectly as a guide to how to produce work that will be appreciated. A particular application of reading, moreover, is to become a reliable referee or thesis examiner.[1] This chapter, then, is about both reading and reviewing. It is an introduction to the elements of effective reading, and is a guide to reviewing.

[1] I've used "referee" rather than "reviewer" in this book (but "reviewing" rather than the slightly awkward "refereeing"), but in my experience the terms are used to mean much the same thing. Here, though, I am also using "referee" to mean "examiner", because thesis examination involves many of the same skills as paper reviewing, and an understanding of the reviewing process will help you prepare your thesis.

© Springer-Verlag London 2014
J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9_3

Reviewing is a central part of the scientific process. Criticism and analysis of papers written by other scientists is the main mechanism for identifying good research and eliminating bad, and is arguably as important an activity as research itself. Many people are intimidated by the task of reviewing, perhaps because it is a kind of intellectual test: you have been asked to demonstrate a thorough understanding of someone else's work. It is also intimidating because it brings responsibility; you want to neither wrongly criticize solid work nor recommend that flawed research be published. Unhelpfully, the quality of reviewing is highly variable—most researchers have stories to tell of good work rejected with only a few hasty words of explanation, or of referees who haven't read the work at all.

Reviewing can be a chore, but deserves the same effort, care, and ethical standards as any other research activity. And it has rewards, beyond the gratitude of editors and authors. It can lead you to look at your own work from a fresh perspective, and exposes you to different kinds of error or failure in research—the average standard of work submitted for publication is well below that of work that gets published. And, while you may not be asked to referee a paper as part of your research, your own work will be reviewed or examined, and thus this chapter provides a perspective on the standards expected of a submitted paper.

Research Literature

By the time your research is complete, you need to be confident that you have read and understood all of the scientific literature that has a significant connection to your work. Your reading achieves several aims. It establishes that your work is indeed novel or innovative; it helps you to understand current theory, discoveries, and debates; it can identify new lines of questioning or investigation; and it should provide alternative perspectives on your work. This reading will ultimately be summarized in the background sections and the discussions of related work in your write-ups.

The literature on which your work will rely is usually expected to be papers that have been refereed and published in a reputable venue, theses that have been undertaken and examined at a reputable institution, and books that are based on the information presented in refereed theses and books. These are the documents that are accepted by the research community as a source of knowledge; indeed, they can be regarded as being the entirety of our scientific knowledge. The literature does not include primary sources such as lab notebooks, responses to a survey, or outputs from an experiment. What these lack is interpretation of the contents in light of a specific hypothesis. Other literature—news articles, science magazines, Wikipedia pages, or documentation, for example—may alert you to the existence of reputable work, but is rarely worth citing. That is, your learning may be built on a wider literature, but the arguments in your write-up should be based on knowledge that is from a refereed source.

A thorough search of the literature can easily lead to discovery of hundreds of potentially relevant papers. However, papers are not textbooks, and should not be treated as textbooks. A researcher reading a paper is not studying for an exam; there is rarely a need to understand every line. The number of papers that a researcher working on a particular project has to know well is usually small, even though the number the researcher should have read to establish their relevance is large.

Thus it is important to become an effective reader, by giving each paper neither more nor less time that it deserves. The first time you read a paper, skim through it to identify the extent to which it is relevant—only read it thoroughly if there is likely to be value in doing so. Make the effort to properly understand the details, but always beware of details that may be wrong, or garbled.

Expect to have a range of modes of reading: browsing to find papers and get an overview of activity and to understand the main outcomes in a body of work; background reading of texts and popular science magazines; and thorough, focused reading of key or complex papers that stretch your abilities or the limits of your understanding. But don't allow reading to develop into a form of procrastination—it needs to be part of a productive cycle of work, not a dominant use of time.

Finding Research Papers

Each research project builds on a body of prior work. Doing and describing research requires a thorough knowledge of the work of others. However, the number of papers published in major computer science venues each year is at least tens of thousands, a volume that prohibits reading or understanding more than a fraction of the papers appearing in any one field.

A consolation is that, in an active field, other researchers have to a certain extent already explored and digested the older literature. Their work provides a guide to earlier research—as will your work, once it is published—and thus a complete exploration of the archives is rarely necessary. However, this is one more reason to carefully search for current work. And beware: reading about a paper that seems relevant is never a substitute for reading the paper itself. If you need to discuss or cite a paper, read it first.

Comprehensive exploration of relevant literature involves following several intertwined paths:

- Use obvious search terms to explore the Web. You are likely to find, not just papers, but also home pages for projects and teams concerned with the same research area. You are also likely to find documents that suggest further valuable search terms. Be exploratory in your search; sometimes the research in an area is divided across separate communities that have different vocabularies.
- Some of the major search engines have search tools that are specifically for academic papers. These may index by individual, by institution, and by citation. They are today the single most effective method for finding relevant work.
- Visit the websites of research groups and researchers working in the area. These sites should give several kinds of links into the wider literature: the names of researchers whose work you should investigate, the names of their co-authors,

conferences where relevant work appears, and papers with lists of references to explore.

- Follow up the references in promising research papers. These indicate relevant individuals, conferences, and journals.
- Browse the recent issues of the journals and conferences in the area; search other journals and conferences that might carry relevant papers.
- Search the publisher-specific digital libraries. These include publishers such as Wiley and Springer, and professional societies such as the ACM and IEEE. There are also a wide range of online archives, in particular www.arXiv.org.
- Most conferences have websites that list the program, that is, the papers to appear in the conference that year. Within a conference, papers are often grouped by topic—another hint of relevance.
- Consider using the citation indexes. The traditional printed citation indexes have migrated to the Web, but in practice their value for computer science is limited, as only a fraction of publications are included.
- Go to the library. The simple strategy of having similar material shelved together often leads to unexpected discoveries, without the distractions that arise when browsing the Web.
- Discuss your work with as many people as possible. Some of them may well know of relevant work you haven't encountered. Similar problems often arise in disparate research areas, but the difficulties of keeping up with other fields—the phenomenon sometimes characterized as "working in silos"—mean that people investigating similar problems can be unaware of one another.

The process of search and discovery of useful papers can be thought of as a form of learning. Each paper or page that you find should refine your understanding of the terminology, help indicate which papers are significant, suggest new directions to follow up, and further clarify your criteria for whether a paper is "in" or "out".

Occasionally there are several versions of the same paper: a preprint in an online archive, a conference version, and a journal version.[2] You should use the version that the author appears to regard as definitive; this will usually be the polished work that has appeared in a journal.

Take a broad definition of "relevant" when searching for papers. It doesn't just mean those papers that have, say, proposals for competing methods. Does the paper have interesting insights into other research literature? Does it establish a benchmark? Have the authors found a clever way of proving a theorem that you can apply in your own work? Does the paper justify a decision to not pursue some particular line of investigation? Other people's research can have many different kinds of effect on your work.

Finding *all* relevant work is hard; for example, exhaustive professional searches across the medical research literature can take months of full-time work. But finding

[2] As I have noted elsewhere in this book, the practice of revising a conference paper to create a journal submission was once common, but, for refereed conference papers that are available online from a major publisher, is now infrequent—and is increasingly discouraged. However, where multiple versions exist, they need to be cited correctly.

all *significant* work is a critical part of doing research. It is typically the case, too, that the scope of literature that is relevant to you may only be obvious once you have completed the investigative phase of your research and have a good draft of your literature review. Your topic and interests are likely to shift, focus, or broaden during your research, and your perspective will change as your understanding develops—update your searching as you go.

Searching and reading are separate activities, and it is a mistake to try and do both at once. I recommend that you uncritically gather material and then later critically analyze and categorize. Save the papers you find into a directory, and go through it later to understand what you have found. In the context of a single search session, it is also helpful to restrict your attention to one or two specific topics.

Having explored the literature, you may discover that your original idea is not so original after all. If so, be honest—review your work to see what aspects may be novel, but don't fool yourself into working on a problem that is already solved. Occasionally it happens, for example, that the same problem has been investigated by several other teams over a considerable period. At the same time, the fact that other people have worked on the same problem does not mean that it is impossible to make further contributions in the area.

Critical Reading

A key aim of reading is to develop critical thinking skills. Good researchers must demonstrate their ability to objectively analyze the work and claims of others. With experience, you can place each paper in a context of other work that you know, and assess it on a range of characteristics.

In doing so, you will become alert to common mistakes and bogus claims. A challenge of research literature is whether to believe what you read. Work published in a reputable journal or conference is peer-reviewed; work available online could have any history, from being a prepublication version of an accepted journal paper to being plagiarised work poorly translated from a non-English original. A cynical but often accurate rule of thumb is that work that is more than one or two years old and has not been published in a significant venue probably has some serious defect. When you find a version of a paper on the Web, establish whether it has been published somewhere. Use evidence such as the quality of the author's other publications to establish whether it is part of a serious program of research.

Much research—far too much—is just misguided. People investigate problems that are already solved and well understood, or solve problems that technology has made irrelevant, or don't realise that the proposed improvement actually makes the method worse. Mathematics may be pointless; the wrong property may be proved, such as complexity instead of correctness; assumptions may be implausible; evaluation strategies may not make sense. The data set used may be so tiny that the results are meaningless. Some results are just plain wrong. And, while the fact that a paper is refereed is an indicator that it is of value, it is not a guarantee. Many people undertake work that did not deserve to be written; sometimes it gets published.

Indeed, few papers are perfect. They are a presentation of new work rather than a considered explanation of well-known results, and the constraints of writing to a deadline mean that mistakes are undiscovered and some issues unexplored. Some aspects of older papers may be superseded or irrelevant, or may rely on limited or technically outdated assumptions. A paper can be seen as a snapshot of a research program at a moment in time—what the researchers knew when they submitted. For all these reasons, a reader needs to be questioning, balanced, and skeptical. In short, don't accept something as true just because it was published. But that does not justify researchers being dismissive of past work; rather, they should respect it and learn from it, because their own work is likely to have similar strengths and weaknesses. Some inexperienced researchers see other work as either perfect or poor, with nothing in-between. Usually, neither of these extremes is correct.

While many papers may be flawed, they define scientific knowledge. (In contrast, textbooks are usually consolidations of older, established work that is no longer at the frontier.) If many researchers trust a particular paper, it is still reasonable to be skeptical of its results, but this needs to be balanced against the fact that, if skepticism is justified, these other researchers are all mistaken.

Read papers by asking critical questions of them, such as:

- Is there a contribution? Is it significant?
- Is the contribution of interest?
- Are the results correct?
- Is the appropriate literature discussed?
- Does the methodology actually answer the initial question?
- Are the proposals and results critically analyzed?
- Are appropriate conclusions drawn from the results, or are there other possible interpretations?
- Are all the technical details correct? Are they sensible?
- Could the results be verified?
- Are there any serious ambiguities or inconsistencies?

That is, actively attempt to identify the contributions and shortcomings rather than simply reading from one end to the other. This analysis of a paper can be thought of as verifying that each component is robust. If the paper is important to your work, you should analyze it until you have formed a reasoned opinion about each of its components; and if some components are questionable, this should be reflected in your literature review.

Write down your analysis of the paper—you will read hundreds of papers, and in some cases will not formally describe them until months or years later. However, detailed analysis can be difficult before you have undertaken your own work, and in so doing developed a mature perspective. Thus reviewing of literature should not stop when the investigation begins, but continue alongside the research.

Developing a Literature Review

A literature review is a structured analysis of a body of literature, and may cover work from several separate areas of research. This review is not simply a list of these papers. Rather, the papers should be grouped by topic, and critically discussed in a way that allows the reader to understand their contribution to the field, their limitations, and the questions that they leave open.

The task of writing a literature review for a paper can be challenging, and for a thesis can be more demanding than any other single activity. It therefore makes good sense to develop the review progressively.

Begin a rough literature review as soon as you start reading, and, when you read a paper that you think will need to be discussed, add it in. (You should also capture the bibliographic data as you go, and also keep a copy of every paper you read.) Initially, the literature review will be sketchy and unstructured, but as you add papers you can group them by topic and contribution, and add notes on each paper and how they relate to each other. Briefly summarize each paper's contribution and the evidence used to support the claims, and also note any shortcomings or features that are of interest. You might also want to note, for your own reference, how the work might have been done better: for example, are there obvious experiments that should have been tried, or plausible counter-arguments to the claims? Keep in mind that your understanding of other work helps examiners to judge your abilities as a researcher.

There is no need for these drafts of your literature review to be polished—in all likelihood, no-one but you will ever read these early versions—so think of the writing as a letter to your future self. That is, at this stage you should focus on organization and content rather than on presentation. The rewriting, editing, and polishing that produces the final literature review will probably be done in a focused way only once your research is complete.

As your review proceeds, it will become easier to decide whether to include each of the papers you read. Many obvious factors will guide your decisions: how close some other work is to yours, or how influential it has been. Some factors may be more subtle. For example, you may find a survey paper, or a recent paper with a thorough literature review of its own, that means that many of the older papers do not need to be discussed; some papers that initially seemed important might on reflection seem less relevant, and can be set aside or noted in passing; while a paper that at first seemed too theoretical or abstract may on further investigation be revealed as foundational.

I suggest that in early drafts you be as inclusive as possible. When you do remove discussion of a paper, put the discussion in another file (or comment it out) rather than deleting it altogether, as this text is your record of having read the paper.

Authors, Editors, and Referees

When an author completes a paper, it is submitted to the editor of a journal (or the program chair of a conference) for publication.[3] The editor sends the paper to referees, who evaluate the paper and return assessments. The editor then uses these assessments to decide whether the paper should be accepted, or, in the case of a journal paper, whether further reviewing or revision is required.

Authors are expected to be honest, ethical, and careful in their preparation of papers. It is ultimately the responsibility of the author—not of the journal, the editor, or the referees—to ensure that the contents of a paper are correct. It is also the author's responsibility to ensure that the presentation is at an appropriate standard and that it is their own work unless otherwise stated.

Referees should be fair and objective, maintain confidentiality, and avoid conflict of interest. In addition, they should complete reviews promptly, declare their limitations as referees, take proper care in evaluating the paper, and only recommend acceptance when they are confident that the paper is of adequate standard. Although referees can generally assume that authors have behaved ethically, many weak or flawed papers are submitted, and a disproportionate amount of reviewing is spent on such papers—in part, because they are often resubmitted after rejection. Moreover, it would be negligent of a referee to assume that a paper is correct and interesting for superficial reasons such as good writing, impressive mathematics, or author prestige. Referees must also ensure that their reviews are accurate and of an appropriate standard.

The editor's responsibilities are to choose referees appropriately, ensure that the reviewing is completed promptly and to an adequate standard, arbitrate when the referees' evaluations differ or when the authors argue that a referee's evaluation is incorrect, and use the reviews to decide whether the paper should be accepted.

These participants have very different perspectives. At submission, an author may feel that the work is remarkable and perfected, and is likely to be sensitive to criticism. Even researchers who are not working in the environment of a first-rate research centre, or who have never had guidance from a more experienced researcher, may well be convinced that their papers are excellent and that negative comments are misguided. A consequence is that they may seek ways to address issues raised by referees by making only minimal changes, and need to be persuaded through convincing argument that the referees' views are reasonable.

In contrast, the referee and editor are likely to feel unexcited (the majority of submissions are rejected) and disbelieving—in submissions, strong results are usually wrong. And they too may be sensitive, in this case to typical, frustrating faults, and also to undue criticism of their own work in the author's literature review. In a sense, the reviewing process can be seen as a mechanism for reconciling these different points of view.

[3] If you are new to research, you may want to skip the rest of this chapter—but do return to it before you submit work for reviewing or examination.

Contribution

Contribution is the main criterion for judging a paper. In broad terms, a paper is a contribution if it has two properties: *originality* and *validity*.

The originality of a paper is the degree to which the ideas presented are significant, new, and interesting. Most papers are to some degree extensions or variations of previously published work; really groundbreaking ideas are rare. Nonetheless, interesting or important ideas are more valuable than trivial increments to existing work. Deciding whether there is sufficient originality to warrant publication is the main task of the referee. Only a truly excellent presentation, thorough and written well, can save a paper with marginal new ideas, while a revolutionary paper must be appalling in some respect to be rejected.

When evaluating the significance of a contribution, it is helpful to consider its effect, or *impact*: that is, to judge how much change would follow from the paper being published and widely read. If the only likely effect is passing interest from a few specialists in the area, the paper is minor. If, on the other hand, the likely effect is a widespread change of practice or a flow of interesting new results from other researchers, the paper is indeed groundbreaking.

That some ideas appear obvious does not detract from their originality. Many excellent ideas are obvious in retrospect. Moreover, the ideas in a well-presented paper often seem less sophisticated than those in a poorly presented paper, simply because authors of the former have a better knack for explanation. Obviousness is not grounds for rejecting a paper. The real achievement may have been to ask the right question in the first place or to ask it in the right way, that is, to notice that the problem even existed. Organization of existing ideas in a new way or within an alternative framework can also be an original contribution, as can reevaluation of existing ideas or methods.

The validity of a paper is the degree to which the ideas have been shown to be sound. A paper that does no more than claim from intuition that the proposal should hold is not valid. Good science requires a demonstration of correctness, in a form that allows verification by other scientists. As discussed in Chap. 4 such a demonstration is usually by proof or analysis, modelling, simulation, or experiment, or preferably several of these methods together, and is likely to involve some kind of comparison to existing ideas.

In the area of algorithms, proof and analysis are the accepted means of showing that a proposal is worthwhile. The use of theory and mathematical analysis is one of the cornerstones of computer science: computer technology is ephemeral but theoretical results are timeless. Their very durability, however, creates a need for certainty: an untrustworthy analysis is not valuable. Thus a paper reporting experimental work can be a significant contribution. The experiment, to be of sufficient interest, should test behaviour that had not previously been examined empirically, or contradict "known" results.

Demonstrations of validity, whether by theory or experiment, should be rigorous: carefully described, thorough, and verifiable. Experiments for assessment of algorithms should be based on a good implementation; experiments based on statistical tests of subjects should use sufficiently large samples and appropriate controls. Comparison to existing work is an important part of the demonstration of validity. A new algorithm that is inferior to existing alternatives is unlikely to be significant.

Evaluation of Papers

The process of evaluating a paper involves asking critical questions such as those listed under "Critical reading" earlier in this chapter. In addition, there are further questions that should be asked of a paper that is under review, which should not just be correct but should be suitable for the likely audience.

- Is the contribution timely or only of historical interest?
- Is the topic relevant to the venue's typical readership?
- What is missing? What would complete the presentation? Is any of the material unnecessary?
- How broad is the likely readership?
- Can the paper be understood? Is it clearly written? Is the presentation at an adequate standard?
- Does the content justify the length?

Of these, contribution is the single most significant component, and requires a value judgement. The presence of a critical analysis is also important: authors should correctly identify the strengths, weaknesses, and implications of their work, and not ignore problems or shortcomings. It is easier to trust results when they are described in a balanced way.

Most papers have an explicit or implicit hypothesis—some assertion that is claimed to be true—and a proposal or innovation. Try to identify what the hypothesis is: if you can't identify it, there is probably something wrong, and if you can, it will help you to recognize whether all of the paper is pertinent to the hypothesis, and whether important material is missing.

The quality of a paper can be reflected in its bibliography. For example, how many references are there? This is a crude rule-of-thumb, but often effective. For some research problems there are only a few relevant papers, but such cases are the exception. The presence of only a few references may be evidence of bad scholarship. Also, some authors cite a reasonable number of papers without actually citing related literature, thus disguising a core bibliography that is far too short. If many of the references are by the author, it may be that some of them are redundant. If only a couple of the references are recent, the author doesn't appear to be familiar with other research. Similarly, be suspicious of papers with no references to the major journals or conferences in the area. Expect author-provided evidence of novelty and innovation, via the right citations.

Occasionally an author submits a paper that is seriously incomplete. No effort has been made to find relevant literature, or the proofs are only sketched, or it is clear that the paper has never been proofread, or, in an extreme case, the paper does little more than outline the basic idea. Such authors perhaps want to establish that an idea is theirs, without going to the trouble of demonstrating its correctness, or are simply tired of the work and hope referees will supply details they haven't bothered to obtain themselves. Such papers don't deserve a thorough evaluation. However, don't be too quick to judge a paper as being in this category.

Referees should make an effort to search for errors that don't affect the quality of the work but should be corrected before going into print. These include spelling and syntax, written expression, errors in the bibliography, whether all concepts and terms have been defined or explained, errors in any formulas or mathematics, and inconsistency in just about anything from variable names to table layout to formatting of the bibliography. Some of these kinds of errors may be picked up in the typesetting process, if the paper is to appear in a journal, but many of them won't be. In particular, only a referee is likely to find errors in mathematics.

Such errors can become more serious defects that might make the paper unacceptable. A few typographic errors in the mathematics are to be expected, for example, but if the subscripts are mixed up or the notation keeps changing case then it is quite likely that the author has not checked the results with sufficient care; it may well be reasonable to reject the paper and expect the author to review it before submitting again.

Similar arguments apply to the presentation: to a certain extent poorly written papers should be accepted (however reluctantly), but real incompetence in the presentation is grounds for rejection, because a paper is of no value if it cannot be read. But note that the converse does not apply: excellence in presentation does not justify acceptance. Occasionally a referee receives a paper that is well written and shows real care in the development of the results, but which does no more than reproduce existing work. Such papers must, regrettably, be rejected.

A difficult issue for some papers is whether to recommend outright rejection or to recommend resubmission after major changes. The latter means that, with no more than a reasonable amount of additional work, the paper could be of acceptable standard. This recommendation should not be used as a form of "soft reject", to spare the author's feelings or some such, while asking for changes that are in practice impossible; eventual acceptance, perhaps after several more rounds of reviewing, is the usual final result of such a recommendation. If getting the work to an acceptable standard will involve substantial additional research and writing, rejection is appropriate. This verdict can be softened in other ways, such as suggesting that the paper be resubmitted once the problems have been addressed.

As a consequence of the peer review system, active researchers should expect to referee about two to three times as many papers as they submit (or somewhat less if their papers are usually co-authored) and only decline to referee a paper with good reason. For many papers, there may be no potential referee who is truly expert in the area, so be prepared to referee even when you are not confident in your judgement of the paper. Always state your limitations as a referee—that you are unfamiliar with the literature in the area, for example, or were not able to check that certain proofs were correct. That is, you need to admit your ignorance. Ultimately, a referee should not recommend acceptance if the paper is not of adequate standard in some respect—the onus is on the referee to fully evaluate the paper.

Content of Reviews

Reviewing of papers serves two purposes. The explicit purpose is that it is the mechanism used by editors to decide whether papers should be accepted for publication. The implicit purpose—equally important, and often overlooked—is that it is a means of sharing expertise between scientists, via comments for the authors. Reviews usually include other things besides written comments (such as scores on certain criteria, which are used to determine whether the paper should be accepted), but it is the comments that authors find valuable. The review should make some kind of case about the paper: whether it is of an adequate standard and what its flaws are. That is, it is an analysis of the paper, explaining why it is or is not suitable for publication.

There are two main criteria for measuring referees' reviews.

- Is the case for or against the paper convincing?
  When recommending that a paper be accepted, the editor must be persuaded that it is of an adequate standard. Brief, superficial comments with no discussion of the detail of the paper provoke the suspicion that the paper has not been carefully refereed. A positive review should not just be a summary of the paper; it should contain a clear statement of what you believe the contribution to be.
  When recommending that a paper be rejected, a clear explanation of the faults should be provided. It is not reasonable, for example, to simply claim without references and explanation that the work is not original or that it has been done before—why should the author believe such a claim if no evidence is given? Having gone to considerable lengths to conduct and present their work, few authors will be persuaded to discard it by a couple of dismissive comments, and will instead resubmit elsewhere without making changes.
- Is there adequate guidance for the authors?
  When recommending that a paper be accepted, referees should describe any changes required to fix residual faults or to improve the paper in any way—technically, stylistically, whatever. If the referee doesn't suggest such changes, they won't get made.
  When recommending that a paper be rejected, a referee should consider what the authors might do next—how they can proceed from the rejection to good research. There are two cases. One is that the paper has some worthwhile core that, with further work, will be acceptable. A referee should highlight that core and explain at least in general terms how the authors should alter and improve their work. The other case is that nothing of the work is worthwhile, in which event the referee should explain to the author how to come to the same conclusion. Sometimes the referee just cannot tell whether there is worthwhile material because of defects in the presentation. It is helpful to explain to the authors how they might judge the significance of their work for themselves by, for example, sketching questions the authors should consider.

There are many reasons why these criteria should be observed. The scientific community prides itself on its spirit of collaboration, and it is in that spirit that referees should help others to improve their work. Poor reviews, although saving the referee effort, make more work for the research community as a whole: if a paper's shortcomings are not adequately explained, they will still be present if the paper is resubmitted. Most of all, poor reviewing is self-reinforcing and is bad for scientific standards. It creates a culture of lacklustre checking of other people's work and ultimately saps confidence in published research.

In a review recommending acceptance, there is no further chance to correct mistakes—the referee is the last expert who will carefully examine the paper prior to its going into print. As noted earlier, only obvious errors such as spelling and punctuation may be caught later, and the referee should check that the paper is substantially correct: no obvious mathematical errors, no logical errors in proofs, no improbable experimental results, no problems in the bibliography, no bogus or inflated claims, and no serious omission of vital information or inclusion of irrelevant text.

In reviews that recommend rejection or substantial revision, such fine-grain checking is not as important, since (presumably) the paper contains gross errors of some kind. Nonetheless some level of care is essential, if only to prevent a cycle of correction and resubmission with only a few points addressed each time. Specific, clear guidance on improving the paper is always welcome. But don't slip into doing the research for the author. If the work is inept, step back; if the work is strong, your contribution isn't needed; if simple changes will make a real difference, suggest them, but it is the author's job to take them to completion.

Drafting a Review

First impressions of papers can be misleading. My reviewing process is to read the paper and make marginal notes, then decide whether the paper should be accepted, then write the comments to the authors. But often, even in that last stage, my opinion of a paper changes, sometimes dramatically. Perhaps what seemed a minor problem is revealed as a major defect, or perhaps the depth of the paper becomes more evident, so that it has greater significance than had seemed to be the case. The lesson is that referees should always be prepared to change their minds, and not commit too soon to a particular point of view.

Another lesson is that positives are as important as negatives: reviews should be constructive. For example, in the reviewing process it is sometimes possible to strengthen the paper anonymously on behalf of the author. The reviewing process can all too easily consist of fault-finding, but it is valuable for authors to learn which aspects of their papers are good as well as which aspects are bad. The good aspects will form the basis of any reworking of the material and should thus be highlighted in a review.

Some referees construct flaws in papers where none exist. For example, an assessment may include generic statements that could be made almost regardless of relevance to the paper's topic, such as "the authors have not considered parallel architectures" on a paper about document processing. Other examples are vague complaints such as “the problem could have been investigated more deeply” or “aspects of the problem were not considered”. Comments of this kind suggest that the referee is not concerned with making a fair evaluation. If there is a genuine problem, then describe it, preferably with examples; otherwise say nothing.

Referees should offer obvious or essential references that have been overlooked, but should not send authors hunting for papers unnecessarily, especially if they are hard to find. A referee who recommends acceptance requires at least a passing familiarity with the literature—enough to have reasonable confidence that the work is new and to recommend references as required.

Referees need to be polite. It can be tempting to break this rule (particularly when evaluating an especially frustrating or ill-considered paper) and be patronizing, sarcastic, or downright insulting, but such comments are not acceptable.

Some review processes allow for confidential remarks that are not seen by the author. You can use these remarks to emphasize particular aspects of your review or, if the editor requested a score rather than a recommendation to accept or reject, to state explicitly whether the paper should be accepted. You can also use this space to tell the editor about your own limitations. However, since authors have no opportunity to defend themselves against comments they cannot see, it is not appropriate to make criticisms in addition to those visible to the author.

Checking Your Review

When you recommend that a paper be accepted, you should:

- Convince yourself that it has no serious defects.
- Convince the editor that it is of an acceptable standard, by explaining why it is original, valid, and clear.
- List the changes, major and minor, that should be made before it appears in print, and where possible help the author by indicating not just what to change but what to change it to; but if there are excessive numbers of errors of some kind, you may instead want to give a few examples and recommend that the paper be proofread.
- Take reasonable care in checking details such as mathematics, formulas, and the bibliography.

When you recommend that a paper be rejected, or recommend that it be resubmitted after major changes, you should:

- Give a clear explanation of the faults and, where possible, discuss how they could be rectified.
- Indicate which parts of the work are of value and which should be discarded, that is, discuss what you believe the contribution to be.
- Check the paper to a reasonable level of detail, unless it is unusually sloppy or ill-thought.

In either case you should:

- Provide good references with which the authors should be familiar.
- Ask yourself whether your comments are fair, specific, and polite.
- Be honest about your limitations as a referee of that paper.
- Check your review as carefully as you would check one of your own papers prior to submission.

Present your arguments in reasonable detail; your writing and presentation may not be at the same standard as in a paper, but the rigour of argument should be similar. Remember that the editor will tend to trust your judgement and views ahead of that of the author. Do not abuse that trust.

Chapter 4
Hypotheses, Questions, and Evidence

The intensity of the conviction that a hypothesis is true has no bearing on whether it is true or not.

P.B. Medawar
Advice to a Young Scientist

The great tragedy of Science, the slaying of a beautiful hypothesis by an ugly fact.

T.H. Huxley
Biogenesis and Abiogenesis

An argument is a connected series of statements intended to establish a proposition … Argument is an intellectual process. Contradiction is just the automatic gainsaying of anything the other person says.

Monty Python
The Argument Sketch

The first stages of a research program involve choice of interesting topics or problems, and then identification of particular issues to investigate. The research is given direction by development of specific questions that the program aims to answer. These questions are based on an understanding—an informal model, perhaps—of how something works, or interacts, or behaves. They establish a framework for making observations about the object being studied. This framework can be characterised as a statement of belief about how the object behaves—in other words, a *hypothesis*.

Many hypotheses concern some aspect of the physical world: whether something is occurring, whether it is possible to alter something in a predictable way, or whether a model is able to accurately predict new events. Astronomers use nuclear physics to predict the brightness of stars from their mass and chemical composition, for example, while a geneticist may seek to know whether substituting one gene for another can improve the health of a cell.

In computer science, some hypotheses are of this kind. We examine the limits of speech recognition, ask whether Web search can be used effectively by children, or predict how well a service will respond to increasing load. Other hypotheses are constructive. For example, we propose new technologies and explore their limitations and feasibility, or propose theorems that imply that there may be new solutions to long-standing algorithmic problems. Regardless of field, if you wish to achieve

© Springer-Verlag London 2014

J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9_4

robust research outcomes it is essential to have a hypothesis. This chapter concerns hypotheses and research questions, and how we use evidence to confirm or disprove them.

Hypotheses

In outline, an example research program might proceed as follows.

- A researcher investigating algorithms might speculate as to whether it is possible to make better use of the cache on a CPU to reduce computational costs.
- Preliminary investigation might lead to the *hypothesis* that a tree-based structure with poor memory locality will be slower in practice than an array-based structure with high locality, despite the additional computational cost.
- The hypothesis suggests the *research question* of whether a particular sorting algorithm can be improved by replacing the tree structure with the array structure.
- The *phenomenon* that should be observed if the hypothesis is correct is a trend: for example, as the number of items to be sorted is increased, the tree-based method should increasingly show a high rate of cache misses compared to the array-based method.
- The *evidence* is the number of cache misses for several sets of items to be sorted. Alternatively, external evidence might be used, such as changes in execution time as the volume of data changes.

As this example illustrates, the structure of the research program flows from having a definite research question and hypothesis.

A hypothesis or research question should be specific and precise, and should be unambiguous; the more loosely a concept is defined, the more easily it will satisfy many needs simultaneously, even when these needs are contradictory. And it is important to state what is *not* being proposed—what the limits on the conclusions will be. Consider an example. Suppose P-lists are a well-known data structure used for a range of applications, in particular as an in-memory search structure that is fast and compact. A scientist has developed a new data structure called the Q-list. Formal analysis has shown the two structures to have the same asymptotic complexity in both space and time, but the scientist intuitively believes the Q-list to be superior in practice and has decided to demonstrate this by experiment.

This motivation by belief, or instinct, is a crucial element of the process of science: since ideas cannot be known to be correct when they are first conceived, it is intuition or plausibility that suggests them as worthy of consideration. That is, the investigation may well have been undertaken for subjective reasons; but the final report on the research—that is, the published paper—must be objective.

Continuing the example above, the hypothesis might be encapsulated as

✗ Q-lists are superior to P-lists.

But this statement is not sufficient as the basis of an experiment: success would have to apply in all applications, in all conditions, for all time. Formal analysis might be able to justify such a result, but no experiment will be so far-reaching. In any case, it is rare for a data structure to be completely superseded—consider the durability of arrays and linked lists—so in all probability this hypothesis is incorrect. A testable hypothesis might be

✓ As an in-memory search structure for large data sets, Q-lists are faster and more compact than P-lists.

Further qualification may well be necessary.

✓ We assume there is a skew access pattern, that is, that the majority of accesses will be to a small proportion of the data.

The qualifying statement imposes a scope on the claims made on behalf of Q-lists. A reader of the hypothesis has enough information to reasonably conclude that Q-lists do not suit a certain application; this limitation does not invalidate the result, but instead strengthens it, by making it more precise. Another scientist would be free to explore the behaviour of Q-lists under another set of conditions, in which they might be inferior to P-lists, but again the original hypothesis remains valid.

As the example illustrates, a hypothesis must be testable. One aspect of testability is that the scope be limited to a domain that can feasibly be explored. Another, crucial aspect is that the hypothesis should be capable of falsification. Vague claims are unlikely to meet this criterion.

✗ Q-list performance is comparable to P-list performance.

✗ Our proposed query language is relatively easy to learn.

The exercise of refining and clarifying a hypothesis may expose that it is not worth pursuing. For example, if complex restrictions must be imposed to make the hypothesis work, or if it is necessary to assume that problems that are currently insoluble must be addressed before the work can be used, how interesting is the research?

A form of research where poor hypotheses seem particularly common is "black box" work, where the black box is an algorithm whose properties are poorly understood. For example, some research consists of applying a black-box learning algorithm to new data, with the outcome that the results are an improvement on a baseline method. (Often, the claim is to the effect that "our black box is significantly better than random".) The apparent ability of these black boxes to solve problems without creative input from a scientist attracts research of low value. A weakness of such research is that it provides no insights into the data or the black box, and has no implications for other investigations. In particular, such results rarely tell us whether the same behaviour would occur if the same approach were applied to a different situation, or even to a new but similar data set.

That is, the results are not *predictive*. There may be cases in which it is interesting to observe the behaviour of an algorithm on some data, but in general the point of experimentation is to confirm models or theories, which can then be used to predict future behaviour. That is, we use experiments to learn about more general properties, a characteristic that is missing from black-box research.

A related problem is the re-naming fallacy, often observed in the work of scientists who are attempting to reposition their research within a fashionable area. Calling a network cache a “local storage agent” doesn’t change its behaviour, and if the term “agent” can legitimately be applied to any executable process then the term’s explanatory power is slim—a particular piece of research is not made innovative merely by changing the terminology. Likewise, a paper on natural language processing for “Web documents” should presumably concern some issues specific to the Web, not just any text; a debatable applicability to the Web does not add to the contribution. And it seems unlikely that a text indexing algorithm is made “intelligent” by improvements to the parsing. Renaming existing research to place it in another field is bad science.

It may be necessary to refine a hypothesis after initial testing; indeed, much of scientific progress can be viewed as refinement and development of hypotheses to fit new observations. Occasionally there is no room for refinement, a classic example being Einstein’s prediction of the deflection of light by massive bodies—a hypothesis much exposed to disproof, since it was believed that significant deviation from the predicted value would invalidate the theory of general relativity. But more typically a hypothesis evolves in tandem with refinements in the experiments.

However, the hypothesis should not follow the experiments. A hypothesis will often be based on observations, but can only be regarded as confirmed if it is able to make successful predictions. There is a vast difference between an observation such as “the algorithm worked on our data” and a tested hypothesis such as “the algorithm was predicted to work on any data of this class, and this prediction has been confirmed on our data”. Another perspective on this issue is that, as far as possible, tests should be blind. If an experiment and hypothesis have been fine-tuned on the data, it cannot be said that the experiment provides confirmation. At best the experiment has provided observations on which the hypothesis is based. In other words: first hypothesize, then test.

Where two hypotheses fit the observations equally well and one is clearly simpler than the other, the simpler should be chosen. This principle, known as Occam’s razor, is purely a convenience; but it is well-established and there is no reason to choose a complex explanation when another is available.

Defending Hypotheses

One component of a strong paper is a precise, interesting hypothesis. Another component is the testing of the hypothesis and the presentation of the supporting evidence. As part of the research process you need to test your hypothesis and if it is correct—or, at least, not falsified—assemble supporting evidence. In presenting the hypothesis, you need to construct an argument relating your hypothesis to the evidence.

For example, the hypothesis “the new range searching method is faster than previous methods” might be supported by the evidence “range search amongst n elements requires $2 \log_2 \log_2 n + c$ comparisons”. This may or may not be good evidence, but it is not convincing because there is no argument connecting the evidence to the hypothesis. What is missing is information such as "results for previous methods indicated an asymptotic cost of $\Theta(\log n)$". It is the role of the connecting argument to show that the evidence does indeed support the hypothesis, and to show that conclusions have been drawn correctly.

In constructing an argument, it can be helpful to imagine yourself defending your hypothesis to a colleague, so that you play the role of inquisitor. That is, raising objections and defending yourself against them is a way of gathering the material that is needed to convince the reader that your argument is correct. Starting from the hypothesis that "the new string hashing algorithm is fast because it doesn't use multiplication or division" you might debate as follows:

- I don't see why multiplication and division are a problem.

  On most machines they use several cycles, or may not be implemented in hardware at all. The new algorithm instead uses two exclusive-or operations per character and a modulo in the final step. I agree that for pipelined machines with floating-point accelerators the difference may not be great.
- Modulo isn't always in hardware either.

  True, but it is only required once.
- So there is also an array lookup? That can be slow.

  Not if the array is cache-resident.
- What happens if the hash table size is not 2^8?

  Good point. This function is most effective for tables of size 2^8, 2^{16}, and so on.

In an argument you need to rebut likely objections while conceding points that can't be rebutted, while also admitting when you are uncertain. If, in the process of developing your hypothesis, you raised an objection but reasoned it away, it can be valuable to include the reasoning in the paper. Doing so allows the reader to follow your train of thought, and greatly helps the reader who independently raises the same objection. That is, you need to anticipate concerns the reader may have about your hypothesis. Likewise, you should actively search for counter-examples.

If you think of an objection that you cannot refute, don't just put it aside. At the very least you should raise it yourself in the paper, but it may well mean that you must reconsider your results.

A hypothesis can be tested in a preliminary way by considering its effect, that is, by examining whether there is a simple argument for keeping or discarding it. For example, are there any improbable consequences if the hypothesis is true? If so, there is a good chance that the hypothesis is wrong. For a hypothesis that displaces or contradicts some currently held belief, is the contradiction such that the belief can only have been held out of stupidity? Again, the hypothesis is probably wrong. Does the hypothesis cover all of the observations explained by the current belief? If not, the hypothesis is probably uninteresting.

Always consider the possibility that your hypothesis is wrong. It is often the case that a correct hypothesis at times seems dubious—perhaps in the early stages, before it is fully developed, or when it appears to be contradicted by initial experimental evidence—but the hypothesis survives and may even be strengthened by test and refinement in the face of doubt. But equally often a hypothesis is false, in which case clinging to it is a waste of time. Persist for long enough to establish whether or not it is likely to be true, but to persist longer is foolish.

A corollary is that the stronger your intuitive liking for a hypothesis, the more rigorously you should test it—that is, attempt to confirm it or disprove it—rather than twist results, and yourself, defending it.

Be persuasive. Using research into the properties of an algorithm as an example, issues such as the following need to be addressed.

- Will the reader believe that the algorithm is new?

  Only if the researcher does a careful literature review, and fully explores and explains previous relevant work. Doing so includes giving credit to significant advances, and not overrating work where the contribution is small.
- Will the reader believe that the algorithm is sensible?

  It had better be explained carefully. Potential problems should be identified, and either conceded—with an explanation, for example, of why the algorithm is not universally applicable—or dismissed through some cogent argument.
- Are the experiments convincing?

  If the code isn't good enough to be made publicly available, is it because there is something wrong with it? Has the right data been used? Has enough data been used?

Every research program suggests its own skeptical questions. Such questioning is also appropriate later in a research program, where it gives the author an opportunity to make a critical assessment of the work.

Forms of Evidence

A paper can be viewed as an assembly of evidence and supporting explanations; that is, as an attempt to persuade others to share your conclusions. Good science uses objective evidence to achieve aims such as to persuade readers to make more informed decisions and to deepen their understanding of problems and solutions. In a write-up you pose a question or hypothesis, then present evidence to support your case. The evidence needs to be convincing because the processes of science rely on readers being critical and skeptical; there is no reason for a reader to be interested in work that is inconclusive.

There are, broadly speaking, four kinds of evidence that can be used to support a hypothesis: proof, modelling, simulation, and experiment.

Proof. An proof is a formal argument that a hypothesis is correct (or wrong). It is a mistake to suppose that the correctness of a proof is absolute—confidence in a proof may be high, but that does not guarantee that it is free from error; it is common for a researcher to feel certain that a theorem is correct but have doubts about the mechanics of the proof.[1]

Some hypotheses are not amenable to formal analysis, particularly hypotheses that involve the real world in some way. For example, human behaviour is intrinsic to questions about interface design, and system properties can be intractably complex. Consider an exploration to determine whether a new method is better than a previous one at lossless compression of images—is it likely that material that is as diverse as images can be modelled well enough to predict the performance of a compression algorithm? It is also a mistake to suppose that an asymptotic analysis is always sufficient. Nonetheless, the possibility of formal proof should never be overlooked.

Model. A model is a mathematical description of the hypothesis (or some component of the hypothesis, such as an algorithm whose properties are being considered) and there will usually be a demonstration that the hypothesis and model do indeed correspond.

In choosing to use a model, consider how realistic it will be, or conversely how many simplifying assumptions need to be made for analysis to be feasible. Take the example of modelling the cost of a Boolean query on a text collection, in which the task is to find the documents that contain each of a set of words. We need to estimate the frequency of each word (because words that are frequent in queries may be rare in documents); the likelihood of query terms occurring in the same document (in practice, query terms are thematically related, and do not model well as random co-occurrences); the fact that longer documents contain more words, but are more expensive to fetch; and, in a practical system, the probability that the same query had been issued recently and the answers are cached in memory. It is possible to define a model based on these factors, but, with so many estimates to make and parameters to tune, it is unlikely that the model would be realistic.

Simulation. A simulation is usually an implementation or partial implementation of a simplified form of the hypothesis, in which the difficulties of a full implementation are sidestepped by omission or approximation. At one extreme a simulation might be little more than an outline; for example, a parallel algorithm could be tested on a sequential machine by use of an interpreter that counts machine cycles and communication costs between simulated processors; at the other extreme a simulation could be an implementation of the hypothesis, but tested on artificial data. A simulation is a "white coats" test: artificial, isolated, and conducted in a tightly controlled environment.

A great advantage of a simulation is that it provides parameters that can be smoothly adjusted, allowing the researcher to observe behaviour across a wide spectrum of inputs or characteristics. For example, if you are comparing algorithms for removal of errors in genetic data, use of simulated data might allow you to control the error rate, and observe when the different algorithms begin to fail. Real data may have unknown numbers of errors, or only a couple of different error rates, so in some sense can be less informative. However, with a simulation there is always the risk

[1] Which can, of course, lead to the discovery that the theorem is wrong after all.

that it is unrealistic or simplistic, with properties that mean that the observed results would not occur in practice. Thus simulations are powerful tools, but, ultimately, need to be verified against reality.

Experiment. An experiment is a full test of the hypothesis, based on an implementation of the proposal and on real—or highly realistic—data. In an experiment there is a sense of *really doing it*, while in a simulation there is a sense of *only pretending*. For example, artificial data provides a mechanism for exploring behaviour, but corresponding behaviour needs to be observed on real data if the outcomes are to be persuasive.

In some cases, though, the distinction between simulation and experiment can be blurry, and, in principle, an experiment only demonstrates that the hypothesis holds for the particular data that was used; modelling and simulation can generalize the conclusion (however imperfectly) to other contexts.

Ideally an experiment should be conducted in the light of predictions made by a model, so that it confirms some expected behaviour. An experiment should be severe; seek out tests that seem likely to fail if the hypothesis is false, and explore extremes. The traditional sciences, and physics in particular, proceed in this way. Theoreticians develop models of phenomena that fit known observations; experimentalists seek confirmation through fresh experiments.

Use of Evidence

Different forms of evidence can be used to confirm one another, with say a simulation used to provide further evidence that a proof is correct. But the different forms should not be confused with one another. For example, suppose that for some algorithm there is a mathematical model of expected performance. Encoding this model in a program and computing predicted performance for certain values of the model parameters is not an experimental test of the algorithm and should never be called an experiment; it does not even confirm that the model is a description of the algorithm. At best it confirms claimed properties of the model.

When choosing whether to use a proof, model, simulation, or experiment as evidence, consider how convincing each is likely to be to the reader. If your evidence is questionable—say a simplistic and assumption-laden model, an involved algebraic analysis and application of advanced statistics, or an experiment on limited data—the reader may well be skeptical of the result. Select a form of evidence, not so as to keep your own effort to a minimum, but to be as persuasive as possible.

Having identified the elements a research plan should cover, end-to-start reasoning suggests how these elements should be prioritized. The write-up is the most important thing; so perhaps it should be started first. Completing the report is certainly more important than hastily running some last-minute experiments, or quickly browsing the literature to make it appear as if past work has been fully evaluated.

Some novice researchers feel that the standards expected of evidence are too high, but readers—including referees and examiners—tend to trust work that is already published in preference to a new, unrefereed paper, and have no reason to trust work where the evidence is thin. Moreover, experienced researchers are well aware that skepticism is justified. It has been said, with considerable truth, that most published research findings are false; and unpublished findings are worse.

This means that a paper must be persuasive. Your written work is the one chance to persuade readers to accept the ideas, and they will only do so if the evidence and arguments are complete and convincing.

Approaches to Measurement

A perspective on the history of science is that it is also a history of development of tools of measurement. Our understanding of the laws of physics followed from development of telescopes, voltmeters, thermometers, and so on. Each improvement in the measurement technology has refined our understanding of the underlying properties of the universe.

From this perspective, the purpose of experimentation is to take measurements that can be used as evidence. A good choice of measure is essential to to practical system improvement and to persuasive and insightful writing. The measurements are intended to be a consequence of some underlying phenomenon that is described by a theory or hypothesis. In this approach to research, phenomena—the eternal truths studied by science—cannot change, but the measurements can, because they depend on the context of the specific experiment. Measurements can be quantitative, such as number or duration or volume—the speed of a system, say, or an algorithm's efficiency relative to a baseline. They can also be qualitative, such as an occurrence or difference—whether an outcome was achieved, or whether particular features were observed.

As you develop your research questions, then, you should ask *what is to be measured?* and *what measures will be used?* For example, when examining an algorithm, will it be measured by execution time? And if so, what mechanism will be used to measure it? This question can be tricky to answer for a single-threaded process running on a single machine. For a distributed process using diverse resources across a network, there probably is no perfect answer, only a range of choices with a variety of flaws and shortcomings, each of which needs to be understood by you and by your readers.

There is then a critical, but more subtle, question: you need to be satisfied that the properties being measured are logically connected to the aims of the research. Typically, research aims are *qualitative*. We seek to improve an interface, accelerate an algorithm, extract information from an image, generate better timetables for lectures, and so on. Measurement is *quantitative*; we find a property that can be represented as a quantity or value. For example, the effectiveness of machine translation systems is sometimes assessed by counting the textual overlap (words or substrings) of a computer translation with that made by a human. However, such a measure is obviously imperfect: not only are there many possible human translations, but a highly overlapping text can still be incoherent, that is, not a good translation.

As another example, we might say that the evidence for the claim that a network is qualitively improved is that average times to transmit a packet are reduced—a quantity that can be measured. But if the aim of network improvement is simplified to the goal of reducing wait times, then other aspects of the qualitative aim (smoothness of transmission of video, say, or effectiveness of service for remote locations) may be neglected.

In other words, once a qualitative aim is replaced by a single quantitative measure, the goal of research in the field can shift away from achievement of a practical outcome, and instead consist entirely of optimization to the measure, regardless of how representative the measure is of the broader problem. A strong research program will rest, in part, on recognition of the distinction between qualitative goals and different quantitative approximations to that goal.

The problem of optimization-to-a-measure is particularly acute for fields that make use of shared reference data sets, where this data is used for evaluation of new methods. It is all too easy for researchers to begin to regard the standard data as being representative of the problem as a whole, and to tune their methods to perform well on just these data sets. Any field in which the measures and the data are static is at risk of becoming stagnant.

Good and Bad Science

Questions about the quality of evidence can be used to evaluate other people's research, and provide an opportunity to reflect on whether the outcomes of your work are worthwhile. There isn't a simple division of research into "good" and "bad", but it is not difficult to distinguish valuable research from work that is weak or pointless.

The merits of formal studies are easy to appreciate. They provide the kind of mathematical link between the possible and the practical that physics provides between the universe and engineering.

The merits of well-designed experimental work are also clear. Work that experimentally confirms or contradicts the correctness of formal studies has historically been undervalued in computer science: perhaps because standards for experimentation have not been high; perhaps because the great diversity of computer systems, languages, and data has made truly general experiments difficult to devise; or perhaps because theoretical work with advanced mathematics is more intellectually imposing than work that some people regard as mere code-cutting. However, many questions cannot be readily answered through analysis, and a theory without practical confirmation is of no more interest in computing than in the rest of science.

Research that consists of proposals and speculation, entirely without a serious attempt at evaluation, can be more difficult to respect. Why should a reader regard such work as valid? If the author cannot offer anything to measure, arguably it isn't science. And research isn't "theoretical" just because it isn't experimental. Theoretical work describes testable theories.

The quality of work can be unclear if the terminology used to describe it is over-inflated. Sometimes such terminology is used to avoid having to define terms properly. A *hypernet*, for example, sounds much more powerful than a network; but who knows if there is really a difference. Researchers use such terminology to make cloudy, big-picture claims that are rarely justified by their actual outcomes.

Terms that in common usage describe aspects of cognition or consciousness, such as "intelligent" or "belief", or even "aware", are particularly slippery. They sound like ordinary concepts we are all familiar with. But in their common usage they are not well defined; and when terms are borrowed from common usage their meaning changes. These terms anthropomorphize the computational behaviour to create a sense of specialness or drama, when in fact what is being described may well be highly mechanical and deterministic, and possibly isn't very interesting. This is a form of the renaming fallacy noted earlier. Thus, while we might have an impression of what the author means when they claim that a system is intelligent, that impression is vague. Successful science is not built on vagueness.

A particular example is the widely misused term "semantic", which, strictly speaking, concerns the meaning of a concept, as distinct from its syntax or representation. But computers are machines for processing representations: enriching the representation by, say, addition of further descriptors does not "bridge the semantic gap". At best, it shifts the problem, from one of computing in the absence of descriptors to one of creating and then making use of descriptors. For example, a text indexing technique that "maps terms to concepts, allowing semantic retrieval" might be no more than a trivial function in which an ontology is used to map words, correctly or otherwise, to labels; retrieval then proceeds as usual, but with labels instead of terms as queries. An additional resource (a dictionary) has been introduced into the process, but the method isn't semantic, and certainly isn't particularly intelligent.

Some science is not simply weak, but can be described as pseudoscience. Inevitably, some claimed achievements are delusional or bogus. Pseudoscience is a broad label covering a range of scientific sins, from self-deception and confusion to outright fraud. A definition is that pseudoscience is work that uses the language and respectability of science to gain credibility for statements that are not based on evidence that meets scientific standards. Much pseudoscience shares a range of characteristics: the results and ideas don't seem to develop over time, systems are never quite ready for demonstration, the work proceeds in a vacuum and is unaffected by other advances, protagonists argue rather than seek evidence, and the results are inconsistent with accepted facts. Often such work is strenuously promoted by one individual or a small number of devotees while the rest of the scientific community ignores it.[2]

[2] An example of pseudoscience in computing are schemes for high-performance video compression that promised delivery of TV-quality data over low-bandwidth modems. In the 1990s, the commercial implications of such systems were enormous, and this incentive created ample opportunities for fraud. In one case, for example, millions of dollars were scammed from investors with tricks such as hiding a video player inside a PC tower and hiding a network cable inside a power cable. Yet, skeptically considered, such schemes are implausible. For example, with current technology, even a corner of a single TV-resolution image—let alone 25 frames—cannot be compressed into

An example is what might be described as "universal" indexing methods. In such methods, the object to be indexed—whether an image, movie, audio file, or text document—is manipulated in some way, for example by a particular kind of hash function. After this manipulation, objects of different type can be compared: thus, somehow, documents about swimming pools and images of swimming pools would have the same representation. Such matching is clearly an extremely difficult problem, if not entirely insoluble; for instance, how does the method know to focus on the swimming pool rather than some other element of the image, such as children, sunshine, or its role as a metaphor for middle-class aspirations?[3]

In some work, the evidence or methods are internally inconsistent. For example, in a paper on how to find documents on a particular topic, the authors reported that the method correctly identified 20,000 matches in a large document collection. But this is a deeply improbable outcome. The figure of 20,000 hints at imprecision—it is too round a number. More significantly, verifying that all 20,000 were matches would require many months of effort. No mention was made of the documents that weren't matches, implying that the method was 100 % accurate; but even the best document-matching methods have high error rates. A later paper by the same authors gave entirely different results for the same method, while claiming similar good results for a new method, thus throwing doubt on the whole research program. And it is a failure of logic to suppose that the fact that two documents match according to some arbitrary algorithm implies that the match is useful to a user.

The logic underlying some papers is outright mystifying. To an author, it may seem a major step to identify and solve a new problem, but such steps can go too far. A paper on retrieval for a specific form of graph used a new query language and matching technique, a new way of evaluating similarity, and data based on a new technique for deriving the graphs from text and semantically (that word again!) labelling the edges. Every element of this paper was a separate contribution whose merit could be disputed. Presented in a brief paper, the work seemed worthless. Inventing a problem, a solution to the problem, and a measure of the solution—all without external justification—is a widespread form of bad science.[4]

(Footnote 2 continued)
the 7 kilobytes that such a modem could transmit per second. Uncompressed, the bandwidth of a modem was only sufficient for one byte per row per image, or, per image, about the space needed to transmit a desktop icon. A further skeptical consideration in this case was that an audio signal was also transmitted. Had the system been legitimate, the inventor must have developed new solutions to the independent problems of image compression, motion encoding, and audio compression.

[3] In another variant of this theme, objects of the same type were clustered together using some kind of similarity metric. Then the patterns of clustering were analyzed, and objects that clustered in similar ways were supposed to have similar subject matter. Although it is disguised by the use of clustering, to be successful such an approach assumes an underlying universal matching method.

[4] An interesting question is how to regard "Zipf's law". This observation—"law" seems a poor choice of terminology in this context—is if nothing else a curious case study. Zipf's books may be widely cited but they are not, I suspect, widely read. In *Human Behaviour and the Principle of Least Effort* (Addison-Wesley, 1949), Zipf used languages and word frequencies as one of several examples to illustrate his observation, but his motivation for the work is not quite what might be expected. He states, for example, that his research "define[s] objectively what we mean by the term

We need to be wary of claimed results, not only because we might disagree for technical reasons but because the behaviour of other researchers may not be objective or reasonable. Another lesson is that acceptance of (or silence about) poor science erodes the perceived need for responsible research, and that it is always reasonable to ask skeptical questions. Yet another lesson is that we need to take care to ensure that our own research is well founded.

Reflections on Research

Philosophers and historians of science have reflected at length on the meaning, elements, and methods of research, from both practical and abstract points of view. While philosophy can seem remote from the practical challenges of research, these reflections can be of great benefit to working scientists, who can learn from an overall perspective on their work. Being able to describe what we do helps us to understand whether we are doing it well.

Such philosophies and definitions of science help to establish guidelines for the practical work that scientists do, and set boundaries on what we can know. However, there are limits to how precise (or interesting) such definitions can be. For example, the question "is computer science a science?" has a low information content.[5] Questions of this kind are sometimes in terms of definitions of science such as "a process for discovering laws that model observed natural phenomena". Such definitions not only exclude disciplines such as computing, but also exclude much of the research now undertaken in disciplines such as biology and medicine. In considering definitions of science, a certain degree of skepticism is valuable; these definitions are made by scientists working within particular disciplines and within the viewpoints that those disciplines impose.[6]

(Footnote 4 continued)
personality" (p. 18), explains the "drives of the Freudian death wish" (p. 17), and "will provide an objective language in terms of which persons can discuss social problems impersonally" (p. 543). It "will help to protect mankind from the virtual criminal action of persons in strategic political, commercial, social, intellectual and academic positions" (p. 544) and "as the authority of revealed religion and its attendant ethics declines, something must take its place … I feel that this type of research may yield results that will fulfill those needs" (p. 544). Perhaps these extraordinary claims are quirks, and in any case opinions do not invalidate scientific results. But it has been argued that the behaviour captured by Zipf's conjecture is a simple consequence of randomness, and, for the example for which the conjecture is often cited (distribution of words in text), the fit between hypothesis and observation is not always strong.

[5] Two philosophers are arguing in a bar. The barman goes over to them and asks, "What are you arguing about?"

"We're debating whether computer science is a science", answers one of them.
"And what do you conclude?" asks the barman.
"We're not sure yet," says the other. "We can't agree on what 'is' means".

[6] But, in fairness, the views here have the same limitations, as they are those of a computer scientist who believes that the discipline stands alongside the traditional sciences.

It is true that, considered as a science, computing is difficult to categorize. The underlying theories—in particular, information theory and computability—appear to describe properties as eternal as those of physics. Yet much research in computer science is many steps removed from foundational theory and more closely resembles engineering or psychology.

A widely agreed description of science is that it is a method for accumulating reliable knowledge. In this viewpoint, scientists adopt the belief that rationality and skepticism are how we learn about the universe and shape new principles, while recognizing that this belief limits the application of science to those ideas that can be examined in a logical way. If the arguments and experiments are sound, if the theory can withstand skeptical scrutiny, if the work was undertaken within a framework of past research and provides a basis for further discovery, then it is science. Much computer science has this form.

Many writers and philosophers have debated the nature of science, and aspects of science such as the validity of different approaches to reasoning. The direct impact of this debate on the day-to-day activity of scientists is small, but it has helped to shape how scientists approach their work. It also provides elements of the ethical framework within which scientists work.

One of the core concepts is *falsification*: experimental evidence, no matter how substantial or voluminous, cannot prove a theory true, while a single counter-example can prove a theory false. A practical consequence of the principle of falsification is that a reasonable scientific method is to search for counter-examples to hypotheses. In this line of reasoning, to search for supporting evidence is pointless, as such evidence cannot tell us that the theory is true. A drawback of this line of reasoning is that, using falsification alone, we cannot learn any new theories; we can only learn that some theories are wrong. Another issue is that, in practice, experiments are often unsuccessful, but the explanation is not that the hypothesis is wrong, but rather that some other assumption was wrong—the response of a scientist to a failed experiment may well be to redesign it. For example, in the decades-long search for gravity waves, there have been many unsuccessful experiments, but a general interpretation of these experiments has been that they show that the equipment is insufficiently sensitive.

Thus falsification can be a valuable guide to the conduct of research, but other guides are also required if the research is to be productive. One such guide is the concept of *confirmation*. In science, confirmation has a weaker meaning than in general usage; when a theory is confirmed, the intended meaning is not that the theory is proved, but that the weight of belief in the theory has been strengthened. Seeking of experiments that confirm theories is an alternative reasonable view of scientific method.

A consequence is that a hypothesis should allow some possibility of being disproved—there should be some experiment whose outcomes could show that they hypothesis is wrong. If not, the hypothesis is simply uninteresting. Consider, for example, the hypothesis "a search engine can find interesting Web pages in response to queries". It is difficult to see how this supposition might be contradicted.

In the light of these descriptions, science can be characterized as an iterative process in which theory and hypothesis dictate a search for evidence—or "facts"—

while we learn from facts and use them to develop theories. But we need initial theories to help us search for facts.

Thus confirmation, falsification, and other descriptions of method help to shape research questions as well as research processes, and contribute to the practice of science. We need to be willing to abandon theories in the face of contradictions, but flexible in response to failure; contradictions may be due to an incorrect hypothesis, faulty experimental apparatus, or poor measurement of the experimental outcomes. We need to be ready to seek plausible alternative explanations of facts or observations, and to find experiments that yield observations that provide insight into theories. That is, theories and evidence are deeply intertwined. A scientific method that gives one primacy over the other is unlikely to be productive, and, to have high impact, our research programs should be designed so that theory and evidence reinforce each other.

A "Hypotheses, Questions, and Evidence" Checklist

Regarding *hypotheses and questions*,

- What phenomena or properties are being investigated? Why are they of interest?
- Has the aim of the research been articulated? What are the specific hypotheses and research questions? Are these elements convincingly connected to each other?
- To what extent is the work innovative? Is this reflected in the claims?
- What would disprove the hypothesis? Does it have any improbable consequences?
- What are the underlying assumptions? Are they sensible?
- Has the work been critically questioned? Have you satisfied yourself that it is sound science?

Regarding *evidence and measurement*,

- What forms of evidence are to be used? If it is a model or a simulation, what demonstrates that the results have practical validity?
- How is the evidence to be measured? Are the chosen methods of measurement objective, appropriate, and reasonable?
- What are the qualitative aims, and what makes the quantitative measures you have chosen appropriate to those aims?
- What compromises or simplifications are inherent in your choice of measure?
- Will the outcomes be predictive?
- What is the argument that will link the evidence to the hypothesis?
- To what extent will positive results persuasively confirm the hypothesis? Will negative results disprove it?
- What are the likely weaknesses of or limitations to your approach?

Chapter 5
Writing a Paper

I used to think about my sentences before writing them down; but ... I have found that it saves time to scribble in a vile hand whole pages as quickly as I possibly can ... Sentences thus scribbled down are often better ones than I could have written deliberately.

Charles Darwin
Autobiography

In every research project, a stage is reached at which it makes sense to begin to write up. A good principle is to begin early: if it is possible to start writing, then writing should start. Shaping the research and its outcomes into a write-up is an effective way of giving structure to a project, even if the work itself has not yet begun.

The three main phases of a write-up are organizing materials so that the work tells a story, giving this story the structure of a thesis or of an academic paper, and actually writing. A paper or thesis, then, is an outcome of a cycle of activity, from speculation through definition and experimentation to write-up, with a range of obstacles and issues that can arise on the way.

But it is much more than a record of the work. Although a research paper or thesis is the result of a process of research that may have been proceeding for months or years, with just a few pages to represent months or more of activity by several people a paper may be only a small window into the work that was done. A thesis is more substantial, and may in some respects be a complete piece of research, but even so represents a digesting of the learning and outcomes into a relatively compact document. How to proceed to a complete document is the topic of this chapter.

The Scope of a Paper

To begin a paper, the first task is to describe your aims. An effective exercise is to write down everything that motivated you to start the research. What did you want to achieve? What problems did you expect to address? What makes the problems interesting? Next, define the scope of the work that you plan to write up. To do so, it is necessary to make choices about what to include, and thus it is necessary to identify

© Springer-Verlag London 2014

J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9_5

what *might* be included. Typically, by this stage your research has become focused on investigation of a small number of specific questions, and you have preliminary experimental or theoretical results that suggest what the core contribution of the work is going to be.

You might start, for example, by asking questions such as:

- Which results are the most surprising?
- What is the one result that other researchers might adopt in their work?
- Are the other outcomes independent enough to be published separately later on? Are they interesting enough to justify their being included?
- Does it make sense to explain the new algorithms first, followed by description of the previous algorithms in terms of how they differ from the new work? Or is the contribution of the new work more obvious if the old approaches are described first, to set the context?
- What assumptions or definitions need to be formalized before the main theorem can be presented?
- What is the key background work that has to be discussed?
- Who is the readership? For example, are you writing for specialists in your area, your examiners, or a general computer science audience?

Other questions are given in the checklist at the end of this chapter.

A valuable exercise at this stage is to speculate on the format and scope of the results. Early in the investigation, decisions will have been made about how the results are to be evaluated—that is, about which measures are to be used to determine whether the research has succeeded or failed. For example, it may be that network congestion is the main respect in which the research is expected to have yielded improvements in performance. But how is network congestion to be measured? As a function of data volume, number of machines, network bandwidth, or something else? Answering this question suggests a form of presentation into which the experimental results can be inserted: a graph, perhaps. The form of this graph can be sketched even before any coding has begun, and doing so identifies the kind of output that the code is required to produce.

Consider a detailed example: an investigation of external sorting in database systems. In this task, a large database table—tens of millions of records, say, constituting many gigabytes—must be sorted on a field that is specified in a query. An effective sorting method is to sort the table one block at a time, storing the sorted blocks in a temporary file then merging them to give the final result. Costs include processing time for sorting and merging, transfer time to and from disk, and temporary space requirements. The balance between these costs is governed by available in-memory buffer space, as large blocks are expensive to sort but cheap to merge. The specific research question being investigated is whether disk costs can be reduced by compressing the data while it is sorted.

Speculation about how compression might affect costs suggest how the work should be measured. For small tables, compression seems unlikely to be of help—compressing and then decompressing adds processing costs but does not provide savings if all the data fits in memory. For large tables, on the other hand, the savings due to reduced disk traffic, increases in the numbers of records per block, and use of less temporary space may be significant. Thus it seems likely that the savings due to compression would increase with the size of table to be sorted, suggesting that results be presented in a graph of data volume against sorting time for fixed block size. Note too that the question of what to measure identifies an implicit assumption: that the data was uncompressed to begin with and is returned uncompressed. All of these decisions and steps help to determine the paper's content.

You may be reporting a particular piece of work, but the way it is reported is determined by the characteristics of the audience. For example, a paper on machine learning for computer vision may have entirely different implications for the two fields, and thus different aspects of the results might be emphasized. Also, an expert on vision cannot be assumed to have any expertise in machine learning, so the way in which the material is explained to the two readerships must be based on your judgement, in each case, of what is common knowledge and what is unfamiliar. The nature of the audience may even determine what can be reported.

Making choices about the content of a paper places limits on its scope; these choices identify material to be excluded. Broadly speaking, many research programs are a cycle of innovation and evaluation, with the answers or resolution of one investigation creating the questions that lead to the next. An advance in, say, string sorting might well have implications for integer sorting, and further work could pursue these implications. But at some point it is necessary to stop undertaking new work and write up what has been achieved so far. The new ideas may well be exciting—and less stale than the work that has been preoccupying you for months—but they are likely to be less well understood, and completing the old work is more important than trying to include too many results. If the newer work can be published independently, then write it up separately. A long, complex paper, however big a breakthrough it represents, is hard to referee. From an editor's perspective, accepting such a paper may be difficult to justify if it squeezes out several other contributions.

Another element in the process of developing a paper is deciding where the work might be published. There are many factors that should be considered when making this decision, such as relevance to your topic and how your work measures against the standard for that forum. In particular, the venue partly determines the scope of a paper. For example, is there a page limit? Are there specific conventions to be observed? Are the other papers in that venue primarily theoretical or experimental? What prior knowledge or background is a reader likely to have? Do the editors require that your code be available online? If you select a particular forum but haven't cited any papers that have appeared there, you may have made the wrong choice.

Once the material for a paper has been collected it has to be organized into a coherent self-contained narrative, which ultimately will form the body of the write-up. Turning this narrative into a write-up involves putting it in the form of an academic paper: including an introduction, a bibliography, and so on. These issues are discussed later in this chapter.

Telling a Story

A cornerstone of good writing is identifying what the reader needs to learn. A strong thesis or paper has a story-like flow, with a sequence of concepts building from a foundation of knowledge assumed to be common to all readers up to new ideas and results. Thus an effective paper educates its readers. It leads readers from what they already know to new knowledge you want them to learn. For this reason, the body of a good paper—everything between the introduction and the conclusions—should have a logical flow that has the feel of a narrative.

The story told by a paper is a walk through the ideas and outcomes that explains the material in a structured way. The first parts of the paper teach the readers the things they need to understand for the later parts, while information that isn't a natural part of this narrative should probably be left out. A way to think about the starting point is to consider the "you" as you were the day that you began your research. Think about what you knew (and didn't know) at that time, and what you have learnt since; your paper or thesis is a chance to teach the past "you" all the knowledge that is needed to become the current "you".

Thus a paper isn't a commentary on the research program or the day-to-day activities of the participants, and isn't an unstructured collection of information and results; nor is it meant to be mysterious. Instead, it is like a guided tour through a gallery, in which each room contains something new for the readers to comprehend. There is also an expectation of logical closure. The early parts of the paper's body typically explain hypotheses or claims; the reader expects to discover by the end whether these are justified.

There are several common ways for structuring the body of a paper, including as a chain, by specificity, by example, and by complexity. Perhaps the most common structure is the first of these alternatives, a *chain* in which the results and the background on which they build dictate a logical order for presentation of the material. First might come, say, a problem statement, then a review of previous solutions and their drawbacks, then the new solution, and finally a demonstration that the solution improves on its predecessors.

The "compression for fast external sorting" project suggests a structure of this kind. The problem statement consists of an explanation of external sorting and an argument that disk access costs are a crucial bottleneck. The review explains standard compression methods and why they cannot be integrated into external sorting. The new solution is the compression method developed in the research. The demonstration is a series of graphs and tables based on experiments that compare the costs of sorting with and without compression.

For some kinds of results, other structures may be preferable. One option is to structure by *specificity*, an approach that is particularly appropriate for results that can be divided into several stages. The material is first outlined in general terms, then the details are progressively filled in. Most technical papers have this organization at the high level, but it can also be used within sections.

Material that might have such a structure is an explanation of a retrieval system. Such systems generally have several components. For example, in text retrieval a parser is required to extract words from the text that is being indexed; this information must be passed to a procedure for building an index; queries must likewise be parsed into a format that is consistent with that of the stored text; and a query evaluator uses the index to identify the records that match a given query. The explanation might begin with a review of this overall structure, then proceed to the detail of the elements.

Another structure is by *example*, in which the idea or result is initially explained by, say, applying it to some typical problem. Then the idea can be explained more formally, in a framework the example has made concrete and familiar. The "compression for fast external sorting" could also be approached in this way. The explanation could begin by considering, hypothetically, the likely impact of compression on sorting. To make the discussion more concrete, a couple of specific instances—a small table and a large table, say—could be used to illustrate the expected behaviour in different circumstances. Given a clear explanation of the hypothetical scenario, you can then proceed to fill in details of the method that was tested in the research.

A final alternative is to structure the body by *complexity*. For example, a simple case can be given first, then a more complex case can be explained as an extension, thus avoiding the difficulty of explaining foundational concepts in a complex framework. This approach is a kind of tutorial: the reader is brought by small steps to the full result. For example, a mathematical result for an object-oriented programming language might initially be applied to some simple case, such as programs in which all objects are of the same class. Then the result could be extended by considering programs with inheritance.[1]

Some other structures are inappropriate for a write-up. For example, the paper should not be a chronological list of experiments and results. The aim is to present the evidence needed to explain an argument, not to list the work undertaken.

The traditional structure for organizing research papers can encourage you to list all proofs or results, then analyze them later; with this structure, however, the narrative flow can be poor. It usually makes more sense to analyze proofs or experimental results as they are presented, particularly since experiments or theorems often follow a logical sequence in which the outcome of one dictates the parameters of the next.

When describing specific results, it is helpful, although not always possible, to begin with a brief overview of whatever has been observed. The rest of the discussion can then be used for amplification rather than further observations. Newspaper articles are often written in this "pyramid" style. The first sentence summarizes the story; the next few sentences review the story again, giving some context; then the remainder of the article presents the whole story in detail. Sections of research papers can sometimes be organized in this way.

---

[1] Structuring by complexity is good for a paper but, often, not so good for ongoing research. For example, the authors may have solved an easy case of a problem, say optimizations for iteration-free programs, motivated by hopeful claims such as "we expect these results to throw light on optimization of programs with loops and recursion". All too often the follow-up paper never appears.

Organization

Scientific papers follow a standard structure that allows readers to quickly discover the main results, and then, if interested, to examine the supporting evidence. Many readers accept or reject conclusions based on a quick scan, not having time to read all the papers they see. A well-structured write-up accommodates this behaviour by having important statements as near the beginning as possible. You need to:

- Describe the work in the context of accepted scientific knowledge.
- State the idea that is being investigated, often as a theory or hypothesis.
- Explain what is new about the idea, what is being evaluated, or what contribution the paper is making.
- Justify the theory, by methods such as proof or experiment.

Theses, journal articles, and conference papers have much the same organization when viewed in outline. There are distinctions in emphasis rather than in specific detail. For a thesis, for example, the literature review may be expected to include a historical discussion outlining the development of the key ideas. There is also an expectation that a thesis is a completed, rounded piece of work—a consolidation of the achievements of a research program as well as a report on specific scientific results. Nonetheless, these forms of write-up have similar structure.

A typical write-up has most of the following components:

Title and Author

Papers begin with their title and information about authors including name, affiliation, and address. The convention in computer science is to not give your position, title, or qualifications; but whether you give your name as A. B. Cee, Ae Cee, Ae B. Cee, or whatever, is a personal decision. Use the same style for your name on all your papers, so that they are indexed together. Include a durable email address or Web address.

Also include a date. Take the trouble to type in the date rather than using “today” facilities that print the date on which the document was last processed, or later you may not be able to tell when the document was completed.

The front matter of a paper may also include other elements. One is acknowledgements, as discussed in Chap. 6, which alternatively may follow the conclusions. Another element is a collection of search terms, keywords, or key phrases—additional terminology that can be used to describe the topic of the paper. Sometimes these keywords must be selected from a specific list. In other cases, the conventions for choosing such terms are not always clear, but in general it is unhelpful to use words that, for example, are a description of the experimental methodology: don’t write “timing experiments”, for example. Use words that concern the paper’s principal themes.

Abstract

An abstract is typically a single paragraph of about 50–200 words. The function of an abstract is to allow readers to judge whether or not the paper is of relevance to them. It should therefore be a concise summary of the paper's aims, scope, and conclusions. There is no space for unnecessary text; an abstract should be kept to as few words as possible while remaining clear and informative. Irrelevancies, such as minor details or a description of the structure of the paper, are usually inappropriate, as are acronyms, mathematics, abbreviations, or citations. Only in rare circumstances should an abstract cite another paper (for example, when one paper consists entirely of analysis of results in another), in which case the reference should be given in full, not as a citation to the bibliography. Sentences such as "We review relevant literature" should be omitted.

Many abstracts follow a five-element organization:

1. A general statement introducing the broad research area of the particular topic being investigated.
2. An explanation of the specific problem (difficulty, obstacle, challenge) to be solved.
3. A review of existing or standard solutions to this problem and their limitations.
4. An outline of the proposed new solution.
5. A summary of how the solution was evaluated and what the outcomes of the evaluation were.

Thus a draft of an abstract can consist of five sentences, one for each of the points above. Introductions should be structured in much the same way, but with a paragraph or two, not a sentence, for each component. A valuable exercise is to read other papers, analyze their abstracts and introductions to see if they have this form, and then decide whether they are effective.

The more specific an abstract is, the more interesting it is likely to be. Instead of writing "space requirements can be significantly reduced", for example, write "space requirements can be reduced by 60 %". Instead of writing "we have a new inversion algorithm", write "we have a new inversion algorithm, based on move-to-front lists".

Many scientists browse research papers outside their area of expertise. You should not assume that all likely readers will be specialists in the topic of the paper—abstracts should be self-contained and written for as broad a readership as possible.

Introduction

An introduction can be regarded as an expanded version of the abstract. It should describe the paper's topic, the problem being studied, references to key papers, the approach to the solution, the scope and limitations of the solution, and the outcomes. There needs to be enough detail to allow readers to decide whether or not they need to read further. It should include motivation: the introduction should explain why the problem is interesting, what the relevant scientific issues are, why the approach taken is a good one, and why the outcomes are significant.

That is, the introduction should show that the paper is worth reading and it should allow the reader to understand your perspective, so that the reader and you can proceed on a basis of common understanding.

The introduction can discuss the importance or ramifications of the conclusions but should include only a brief summary of the supporting evidence, which the interested reader can find in the body of the paper. Relevant literature can be cited in the introduction, but specialized terminology, complex mathematics, and in-depth discussion of the literature belong elsewhere.

A paper isn't a story in which results are kept secret until a surprise ending. The introduction should clearly tell the reader what in the paper is new and what the outcomes are. There may still be a little suspense: revealing what the results are does not necessarily reveal how they were achieved. If, however, the existence of results is concealed until later on, the reader might assume there are no results and discard the paper as worthless.[2]

By the end of the introduction, the reader should understand the scope of the work, and of the problem. For example, if the topic is "mechanisms for collaborative authoring", then you need to explain who is doing the authoring; what abilities and experience are assumed to have; what kinds of tasks they are trying to complete; and how sophisticated the mechanisms need to be.

The reader should also understand the contribution, that is, what the *discovery* of the work is—the core idea that the referees or examiners need to appreciate as novel and important. This understanding requires that the reader appreciates what the properties of this contribution are, what makes it interesting and plausible, what method was used to investigate it, and why the method is appropriate.

Body

The body of a paper should present the results. This presentation should provide necessary background and terminology, explain the chain of reasoning that leads to the conclusions, provide the details of central proofs, summarize any experimental outcomes, and state in detail the conclusions outlined in the introduction. Descriptions of experiments should permit reproduction and verification, as discussed in Chap. 14. There should be careful definitions of the hypothesis and major concepts, even those that were described informally in the introduction. The structure should be evident in the section headings. Since the body can be long, narrative flow and a clear logical structure are essential.

[2] There is an irritating kind of paper in which the reader's interest is baited with comments such as "the analysis, as we show, led to surprising insights" or "as discussed later, this decision had unanticipated benefits", with no hint as to what the surprises or benefits were. This is not an endearing style of presentation.

The body should be reasonably independent of other papers. If, to understand your paper, the reader must find specialized literature such as your earlier papers or an obscure paper by your advisor, then its audience will be limited.

In some disciplines, research papers have highly standardized structures. Editors may require, for example, that you use only the four headings Introduction–Methods–Results–Discussion. This convention has not taken hold in computer science, and in some cases such a structure impedes a clear explanation of the work. For example, use of fixed headings may prohibit development of a complex explanation in stages. In work combining two query resolution techniques, we had to determine how they would interact, based on a fresh evaluation of how they behaved independently. The final structure was, in effect, Introduction–Background–Methods–Results–Discussion–Methods–Results–Discussion.

Even if the standardized section names are not used, the body needs these elements, if not necessarily under their standard headings. Components of the body might include, among other things, background, previous work, proposals, experimental design, analysis, results, and discussion. Specific research projects suggest specific headings. For the "compression for fast external sorting" project sketched earlier, the complete set of section headings might be:

1. Introduction
2. External sorting
3. Compression techniques for database systems
4. Sorting with compression
5. Experimental setup
6. Results and discussion
7. Conclusions

The wording of these headings does not follow the standard form, but the intent of the wording is the same. Sections 2 and 3 are the background; Section 4 contains novel algorithms, and Sections 4 and 5 together are the methods.

The background material can be entirely separate from the discussion of previous work on the same problem. The former is the knowledge the reader needs to understand your contribution. The latter is, often, alternative approaches that are superseded by your work. Together, the discussion of background and previous work also introduce the state of the art and its failings, the importance and circumstances of the research question, and benchmarks or baselines that the new work should be compared to.

A body that consists of descriptions of algorithms followed by a dump of uninterpreted experimental results is not sound science. In such a paper, the context of prior work is not explained, as readers are left to draw their own inferences about what the results mean.

The results section is an assembly of evidence on which the key arguments are based. This typically includes presentations of experimental outcomes, theorems, proofs, analyses of data, and tabulations of investigative outcomes and discoveries. The arguments then convey what the results mean—that is, they need to be explained, analyzed, and interpreted. For the data, readers need to know how the data was gathered, how they might obtain or create the data for themselves, and background on issues such as limitations of the data.

Most experiments yield far more data than can be presented in a paper of reasonable length. Important results can be summarized in a graph or a table, and other outcomes reported in a line or two. It is acceptable to state that experiments have yielded a certain outcome without providing details, so long as those experiments do not affect the main conclusions of the paper (and have actually been performed). Similarly, there may be no need to include the details of proofs of lemmas or minor theorems. This does not excuse you from conducting the experiments or convincing yourself that the results are correct, but such information can be kept in logs of the research rather than included in the paper.

In a thesis, each chapter has structure, including an introduction and a summary or conclusions. This structure varies with the chapter's purpose. A background chapter may gather a variety of topics necessary to understanding of the contribution of the thesis, for example, whereas a chapter on a new algorithm may have a simple linear organization in which the parts of the algorithm are presented in turn. However, the introduction and summary should help to link the thesis together—and thus show how each chapter builds on previous chapters and how subsequent chapters make use of it.

Literature Review

Few results or experiments are entirely new. Usually they are extensions of or corrections to previous research—that is, most results are an incremental addition to existing knowledge. As discussed in Chap. 3, a literature review, or survey, is used to compare the new results to similar previously published results, to describe existing knowledge, and to explain how it is extended by the new results. A survey can also help a reader who is not expert in the field to understand the paper and may point to standard references such as texts or survey articles.

In an ideal paper, the literature review is as interesting and thorough as the description of the paper's contribution. There is great value for the reader in a precise analysis of previous work that explains, for example, how existing methods differ from one another and what their respective strengths and weaknesses are. Such a review also creates a specific expectation of what the contribution of the paper should be—it shapes what the readers expect of your work, and thus shapes how they will respond to your ideas. It is where the reader learns why the problem is a challenge and also learns about the limitations of simple or previous solutions.

The literature review can be early in a paper, to describe the context of the work, and might in that case be part of the introduction; or, less commonly, the literature review can follow or be part of the main body, at which point a detailed comparison between the old and the new can be made. If the literature review is late in a paper, it is easier to present the surveyed results in a consistent terminology, even when the cited papers have differing nomenclature and notation.

In some papers the literature review material is not gathered into a single section, but is discussed where it is used—background material in the introduction, analysis of other researchers' work as new results are introduced, and so on. This approach can help you to write the paper as a flowing narrative, but makes it difficult for a reader to assess your depth of understanding of the field, and I recommend against it.

An issue that is problematic in some research is the relationship between new scientific results and proprietary commercial technology. It often is the case that scientists investigate problems that appear to be solved or addressed in commercial products. For example, there is ongoing academic research into methods for information retrieval despite the success of the search engines deployed on the Web. From the perspective of research principles, the existence of a commercial product is irrelevant: the ideas are not in the public domain, it is not known how the problems were solved in the product, and the researcher's contribution is valid. However, it may well be reckless to ignore the product; it should be cited and discussed, while noting, for example, that the methods and effectiveness of the commercial solution are unknown.

Conclusions

The conclusions section, or summary, is used to draw together the topics discussed in the paper. This section should include a concise statement of the paper's important results and an explanation of their significance. This is an appropriate place to state (or restate) any limitations of the work: shortcomings in the experiments, problems that the theory does not address, and so on.

The conclusions are an appropriate place to look beyond the current context to other problems that were not addressed, to questions that were not answered, to variations that could be explored. They may include speculation, such as discussion of possible consequences of the results.

A *conclusion* is that which concludes, or the end. *Conclusions* are the inferences drawn from a collection of information. Write "Conclusions", not "Conclusion". If you have no conclusions to draw, write "Summary", which is often the appropriate way to end a thesis chapter.

Bibliography

A paper's bibliography, or its set of references, is a complete list of theses, papers, books, and reports cited in the text. No other items should be included.

Appendices

Some papers have appendices giving detail of proofs or experimental results, and, where appropriate, material such as listings of computer programs. The purpose of an appendix is to hold bulky material that would otherwise interfere with the narrative flow of the paper, or material that even interested readers do not need to refer to.

Appendices are only occasionally necessary for a paper, in cases where there is material such as a proof whose length would interrupt the flow. But they often have a useful role in a thesis, where they can be used for supporting material such as ethics approvals, extended tables of data, and transcripts of interviews.

The First Draft

For the first draft of a write-up you may find it helpful to write freely—without particular regard to style, layout, or even punctuation—so that you can concentrate on presenting a smooth flow of ideas in a logical structure. Worrying about how to phrase each sentence tends to result in text that is clear but doesn't form a continuous whole, and authors who are too critical on the first draft are often unable to write anything at all. If you tend to get stuck, just write anything, no matter how awful; but be sure to delete any ravings later.

Some people, when told to just say anything, find they can write freely—if anything is acceptable, then nothing is wrong. For others, finding words is a struggle. A last resort is to write in brief sentences making the simplest possible statements.

✓ In-memory sorting algorithms require random access to records. For large files stored on disk, random access is impractically slow. These files must be sorted in blocks. Each block is loaded into memory and sorted in turn. Sorted blocks are written to temporary files. These temporary files are then merged. There may be many files but in practice the merge can be completed in one pass. Thus each record is read twice and written twice. Temporary space is required for a complete copy of the original file.

This text certainly isn't elegant—it is annoying to read and should be thoroughly edited before the paper is submitted. But it is capturing the ideas, and the writing is proceeding.

A consequence of having a sloppy first draft is that you must edit and revise carefully; initial drafts are often awkwardly written and full of mistakes. But few authors write well on the first draft anyway. The best writing is the result of frequent, thorough revision.

Mathematical content, definitions, and the problem statement should be made precise as early as possible. The hypothesis and the results flow from a clear statement of the problem being tackled. Describing the problem forces you to consider in depth the scope and nature of the research. If you find that you cannot describe the problem precisely, then perhaps your understanding is lacking or the ideas are insufficiently developed.

It was said earlier, but is worth repeating: the writing should begin as soon as the research is started. Right from the start, expect to accumulate useful fragments of text that will later be drawn into the finished write-up. The later the writing is begun, the harder it will be. Delay increases the time between having ideas and having to write about them, increases the pressure to read papers to be discussed, and reduces the number of experiments that can be thoroughly described. Completing your reading, for example, is a poor reason to defer writing, because reading is never complete; and in any case, the best way to develop your understanding of other papers is to write about them.

The writing defines the research, and the one cannot proceed without the other. Writing is a stimulus to research, suggesting fresh ideas and clarifying vague concepts and misunderstandings; and development of the presentation of the results oftens suggest the form the proofs or experiments should take. Gaps in the research may not be apparent until it has been at least preliminarily described. Research is also a stimulus to writing—fine points are quickly forgotten once the work is complete. Don't expect the writing to progress steadily, but do expect progress overall. If the writing seems to have stalled, it is time to put other tasks aside for a while.

A thesis is typically completed over a much longer time than is a paper, but the guidance on writing is the same: the real start of the work is when the writing has begun. Being disciplined about writing is even more important than for a paper, because over a period of years early work will be forgotten if it isn't captured in an organized way.

However, the task of writing a thesis can be broken into manageable stages. In a Ph.D., each chapter can be as rich as a paper, and each is likely to be written separately. Drafting of the technical chapters that contain the contribution tends to be relatively easy (even though the research that underlies these chapters may have consumed years of work), in part because you have been immersed in this material and will find that you have plenty to say about it. The most difficult chapter is usually the background and literature review. The volume of careful reading can be an obstacle, as is the need to write succinctly and fairly about other people's work. If you finish the background first, it will seem as if the main task of writing the thesis is complete.

The introduction can be surprisingly challenging: achievement of a conversational, natural writing style can take many revisions. But this is where the examiner meets you for the first time and, as for any initial meeting, it is important to make a good impression.

From Draft to Submission

There are many approaches to the process of assembling a technical paper. The technique I use for composing is to brainstorm, writing down in point form what has been learnt, what has been achieved, and what the results are. The next step is to prepare a skeleton, choosing results to emphasize and discarding material that on reflection seems irrelevant, and then work out a logical sequence of sections that leads the reader naturally to the results.

A useful discipline is to choose the section titles before writing any text, because if material to be included doesn't seem to belong in any section then the paper's structure is probably faulty. The introduction is completed first and includes an overview of the paper's intended structure, that is, an outline of the order and content of the sections. When the structure is complete, each section can be sketched in perhaps 20–200 words. This approach has the advantage of making the writing task less daunting—it is broken into parts of manageable size—while also creating the impression that the writing is well under way.

When the body and the closing summary are complete, the introduction usually needs substantial revision because the arguments presented in the paper are likely to mature and evolve as the writing proceeds. The final version of the abstract is the last part to be written.

With a reasonably thorough draft completed, it is time to review the write-up's content and contribution. Anticipate likely concerns or objections that the referees may have, and address them; if they can't be addressed, acknowledge them. Consider whether extra work is needed to fill a hole. Ask the probing, critical questions that you would ask of other people's work. The burden of proof is on you, not the reader, so be conservative in your claims and thorough with your evidence.

Completion of a paper tends to focus on writing of the whole document, while a thesis is typically completed chapter by chapter. When planning a schedule for completion of a thesis, you need to allow time for multiple revisions of each chapter, and, crucially, time for your advisor to read each chapter. If two weeks is a typical time for your advisor to return a chapter, and for each of the eight chapters there will be two versions, then your schedule will need to include around eight months to allow for this reading and review time—and also needs to include activities that you can complete while you are waiting for your advisor to respond. This is another reason why it is so important to write early.

During drafting and revision, ensure that the topic of the paper does not drift. At the start of the writing process, you wrote down your aims, motivation, and scope. Use these as a reference. If you feel that you need to write something that is not obviously relevant to your original aims, then either establish the connection clearly or alter the aims. Changing the aims can affect the work in many ways, however, so only do so with great care.

For a novice writer who doesn't know where to begin, a good starting point is imitation. Choose a paper or thesis whose results are of a similar flavour to your own, analyze its organization, and sketch an organization for your results based on the same pattern. The habit of using similar patterns for papers—their standardization—helps to make them easier to read.

Students should keep a comprehensive file of notes as they proceed. This can include records of:

- Meetings.
- Decisions.
- Ideas.
- Expectations of outcomes.
- Papers you have read.
- Sketches of algorithms.

- Code versions.
- Theorems.
- Sources of data.
- Experiments and outcomes
- Sketches of proofs.

Expect this log of activity to be a mixture of written material and data. In its raw state, the content of a file of notes is not suitable for inclusion in a paper, but the themes and issues of the paper can be drawn from the file, and it serves as a memory of issues to discuss and material to include.

Co-authoring

In computer science, most papers are co-authored. The inclusion of several people as authors means that, in principle, all these people contributed in some significant way to the intellectual content of the paper. In many cases, it also means that the task of writing was shared. There are a range of strategies for co-authoring, which vary from colleague to colleague and paper to paper. It is not unusual, for example, for an advisor to use a student's thesis as the basis of a paper, in which case both advisor and student are listed as authors. In this process, the advisor may well dramatically revise the student's work, if only because a typical paper is much shorter than a typical thesis.

In cases where researchers are working more or less as equals, one strategy is to brainstorm the contents of the paper, then for each author to write a designated section. Another strategy—my preferred model for collaboration—is to take turns. One person writes a draft, the next revises and extends, and so on, with each person holding an exclusive lock on the paper while amending it. With this approach, the final paper is likely to be a fairly seamless integration of the styles and contributions of each of the authors (especially if each author contributes to revision of the other authors' work). In contrast, the strategy of writing sections separately tends to lead to papers in which the authorial voice makes dramatic shifts, the tables and figures are inconsistent, and there is a great deal of repetition and omission.

Taking turns is effective, but it does have pitfalls, and agreed ground rules are needed to make it work. For example, I rarely delete anything a co-author has written, but may comment it out; thus no-one feels that their work has been thrown away. Another element of successful co-authoring is respect; accept your colleagues' views unless you have a good reason not to.

Co-authoring is a form of research training. It is an opportunity for advisors to learn in detail where their students are weak, while a paper that has been revised by an advisor is an opportunity for students to contrast their attempts at research writing with that of people with more experience. An advisor's revision of a student's draft of paper can involve a great deal of work, and may be the most thorough feedback on writing that the student receives during the course of a research program.

Theses

A thesis (or dissertation) is how research students present their work for examination. A thesis may have longer-term importance as a description of significant research results, but your primary goal should be to produce a piece of work that the examiners will pass.

The questions that examiners respond to are much the same as those a referee would ask of a paper. That is, the examiners seek evidence of an original, valid contribution developed to an appropriate standard. However, it is a mistake to view a thesis as no more than an extended paper. A paper stands (or does not stand) on the strength of the results. A thesis passes (or fails) on the strength of your demonstration of competence; even if good results are not achieved, the thesis should pass if you have shown the ability to undertake high-quality research. Questions that examiners might be asked to address include whether you have demonstrated command of the fundamentals of the discipline, whether you have the ability to correctly interpret results, and whether you have sufficiently strong communication skills.

That is, fundamentally it is the student that is being examined, rather than the research. In a paper, the primary element is the contribution: whether the research is novel, interesting, and correct. In a thesis, the primary element is the competence: whether the student has demonstrated that they are capable of undertaking independent research.

A particular element of theses that is often weak is the analysis of the outcomes. All too often the discussion can be summarized as "the code ran", "it seems plausible", or "look at the pretty feature". To a greater degree than in a paper, it is necessary to probe why the outcomes occurred or what factors or variables were significant in the experiments. The guidelines to examiners issued by many universities state that the candidate must demonstrate critical thinking. Application of critical thinking and skeptical questioning to the work is an excellent way of persuading an examiner that the candidate understands their own methods and results.

Examiners are unlikely to be impressed by students who make grandiose claims about their work. Many researchers—and not just students—are reluctant to admit that their discoveries have any limitations; yet one of the clearest demonstrations of research ability is to ask incisive questions. Was the algorithm an improvement because of better cache use or fewer CPU cycles? What else would explain these results? In what circumstances is the theorem not applicable?

A thesis with negative results can, if appropriately written, demonstrate the ability of the candidate just as well as a thesis with positive results. The outcomes may be less interesting, but the capability to undertake research has still been shown. Examiners focus on, for example, whether there is a clear, consistent presentation and a thorough critical analysis: What do the results imply? Where did the research succeed? Where did it fail? What problems were *not* solved? What questions are suggested?

Examiners are also unlikely to be impressed by a student who accepts the word of established authority without question, or rejects other ideas without giving them due consideration, or appears reluctant to suggest any change or to make unfavourable comment. If you have a relevant point to make, and can defend it by reasonable argument, then make it. Be thorough. A Ph.D. is an opportunity to do research in depth; shortcuts and incomplete experiments suggest shoddy work.

For an extended research degree such as a Ph.D., another difference between a thesis and a paper is that the former may report on a series of more or less independent research discoveries. In contrast, a typical paper concerns a single consistent investigation. A thesis may, moreover, include work drawn from multiple papers. For this reason, there is more variation in structure from thesis to thesis than from paper to paper. An example of the problems faced in organizing a thesis is how to consolidate descriptions of new algorithms. It may make sense to bring all of them into a single chapter and then evaluate and compare them in subsequent chapters, or it may be preferable to describe them one by one, evaluating each in turn. Factors to consider in choosing an organization include how cohesive the algorithms are (for example, whether they address the same problem) and whether an explanation of one algorithm is meaningful if the previous one has not yet been evaluated.

As the scope of a thesis is more substantial than that of a paper, the introduction should be broad in topic and conversational in tone. It could introduce a whole area rather than a single problem. Another reason to develop a substantial introduction is that a thesis is a more thorough, detailed document than is a paper. Why was the problem worth investigating in depth? How do the parts of the investigation relate to each other? What are some practical, concrete ways in which the outcomes of the work might be used? Running examples may be outlined in the introduction, to give unity to the thesis overall. The role of a thesis's introduction is, however, much the same as in a paper. As in the introduction of a paper, theory, jargon, and notation are inappropriate.

Take the time to learn about the challenges that are specific to thesis writing.[3] Browse other theses, from your own institution, other institutions, and other disciplines. Form views about the strengths and weaknesses of these theses; these views will help to shape your own work. Critically, remember that an examiner may only have hours to read your work—you need to help them to spend that time well.

Getting It Wrong

Over the next few chapters I look at the details of how to write well and also some common mistakes that researchers make; these largely concern details, that is, individual elements that are poor.

Some problems in papers, however, are at a higher level, and concern the quality of the work as a whole. As a journal editor, conference chair, and referee, I see defects of this kind again and again—problems that make it certain that the paper will be rejected, and which in some cases are obvious to the referee in the first few moments of reading.

Common ways in which authors "get it wrong" are below. Many of these issues are also discussed elsewhere in this book, but it is valuable, I think, to consider

[3] There are plenty of good textbooks on this topic, and a couple by me.

them together. Something that can be confronting about these issues is that, often, the authors appear to have worked hard over a long period of time to produce a substantial document; and yet it immediately obvious that there is no chance of the work being accepted. An experienced researcher may feel baffled that this has occurred—as the work progressed, did no one see that it was going badly?

Be alert to the potential faults in your own work, and have the courage to abandon or refocus activity that has little chance of leading to a valuable outcome. And, while the examples below are in some respects extreme—which makes them easy to understand—they do highlight the kinds of issues that readers become alert to, and which authors should therefore avoid.

Irrelevance

When I first see a paper, impressions form in a minute or two, influenced by layout, readability, and so on. With some papers, though, a positive initial response is gradually followed by a sinking feeling: *I cannot figure out what this paper is about*. Something elementary is utterly missing.

What that "something" is can vary. Sometimes there is a lack of connection to the literature on any particular topic, and thus no sense of what the author is trying to achieve. In some cases the author has proposed an elegant solution, but it is not obvious what the problem is, or the problem is so unrealistic that it is impossible to grasp.[4] It may be that the author has given a clear motivation for the work, but the bulk of the paper concerns something else entirely; an example was a paper whose starting point was the challenge faced by teachers who wish to ensure that Web searches only return pages that are appropriate for children, but the contribution concerned mechanisms for selectively highlighting passages that were relevant to the query.

Another form of this are those papers that are submitted to an inappropriate venue:[5] work on file compression submitted to a conference on database modelling, or work on face recognition submitted to a journal on data visualization.[6] Even more

[4] I once struggled with a paper that concerned relational databases, but in which each record—and I do mean record, not table—had an arbitrary number of fields. So, not relational then, but in some places relational properties were assumed. (Like many of the examples in this text, this instance is "real" but altered to disguise its origins, and also to make it easy to explain in a sentence or two.) And another in which the authors assumed that Web queries are the result of a random walk through a weighted graph representing mental representations of related concepts, and wished to use a log of queries to infer the graph. Some rather arbitrary use of terminology ("actuation maps can be made explicit through provocation by deliberative stimuli") was at first intimidating, until I realised the authors were using it to disguise the fact that they hadn't figured out how to achieve anything.

[5] Which is not the same thing as venues that are inappropriate. A consequence of publication pressures has been the rise of journals and conferences that seem little more than opportunistic, with glossy web presences, plausible editorial boards or program committees, and even affiliations with major professional societies—but with low standards of refereeing, high publication or registration costs, and, ultimately, no citations.

[6] At a journal where most of the submissions were on Web search, I received respectable papers on automated migration of software between operating systems and on a method for evaluating a

surprising are papers where the authors have utterly misunderstood the norms of research or presentation for the field, such as papers where the authors have made no use of standard resources such as data sets, or, for example, a paper on search technology written as a narrative from the imagined perspective of a document.

Most curiously of all, in some papers there is no obvious research question, no statement of aims or goals, and no claimed contribution. A more subtle problem of this kind is when a paper appears to tell a coherent story, but on inspection it becomes clear that, say, the experimental results are unrelated to the conclusions. In some cases they seem to be on a different topic altogether. An example was a paper that gave results for the efficiency of a string search method but drew the conclusion that the method enhanced data privacy. Stated so concisely, the paper sounds absurd! And yet such problems are not rare.

Inconsistency, Inadequacy, and Incompleteness

Some papers seem reasonable in parts, but the parts don't belong in the same document. A sensible, well-organised paper may be framed in terms of grandiose, ambitious claims that can only be described as ridiculous.[7] Or there may be a detailed, insightful literature review, but it is either disconnected from the contribution, or, bizarrely, the contribution is less interesting than the previous work that was described so well.

For papers that are overall at a high standard, perhaps the single commonest problem that leads to rejection is that the experiments are inadequate. There may be an interesting method, but the experiments are trivial or uninformative, and fall far short of supporting the claims; often, in these cases, the problem is that the data set used is too artificial to allow any interesting conclusion to be drawn. Or a small data set may be used to support claims for applications at an entirely different scale, such as a set of a few thousand documents being used to make claims about Web search. Or the data set may not be relevant to the problem at all. It is as if the researchers

(Footnote 6 continued)
dictionary for medical practitioners, among others. And many that were not so respectable; topics included image enhancement for ancient rock carvings (evaluated on a single image), use of XML for storing machine maintenance logs (utterly trivial), automated translation of eighteenth-century English text into modern usage (only arguably modern, but unarguably garbled and harder to read), and a tool for distinguishing between kinds of spider (use of a computer for a task does not mean the task is computer science).

[7] An example was a Ph.D. thesis that concerned how to develop software specifications in terms of a particular way of describing assertions and tests. The work was ambitious, but did appear to achieve reasonable initial outcomes. However, the motivation was that the work would ultimately make it unnecessary to write programs, and that the specifications could be automatically inferred from transcripts of human conversation. (This condensation of several pages of rambling text into a single sentence doesn't convey the full eccentricity of these claims.) No connection was made between the claims and the actual contribution.

are so excited about the ideas that they fail to see the need for validation, and offer results that have no plausible relevance to the paper's claims.

Another variety of inadequacy is when parts of the paper are missing, or dealt with in a few brief lines rather than pages. Strikingly, some papers have no literature review, or are based on a single out-of-date textbook, as if previous or recent work was of no relevance. But if the author cannot be troubled to properly place the work in context of what is already known, a reader cannot learn what the contribution is. Another common failing is papers where the reader cannot identify what the data is (there may be 200 documents, but where from? what content? what size? and so on), or who ran the experiments, or what techniques were tried. In the mind of the reader, moreover, there may not be much of a distinction between information that is missing and information that is concealed—a line of thought that is unlikely to lead to belief that the work was done well. And some papers just aren't ready to be refereed; the underlying work is unfinished and the paper is incomplete.

Incomprehensibility

In the cases above, the shortcomings are not always immediately apparent. In other cases, the paper's problems are obvious straight away. For example, when presented with an incoherent abstract or introduction, the reader immediately feels that the work cannot be of value. A reader can have no hope for a paper whose abstract begins with the sentence: "Internet supports all type of the forms of information that are digital, such as the pages of the Web everywhere and also libraries and email, so it is a language of all our information sources in a world repository that is our knowledge."[8]

In such cases, there seems to be a wide gap between what the writer wants to say and the actual words on the page. I've observed similar writing in students who are perfectly clear in conversation, but in documents seem to want to pour all of their thoughts into a few sentences. But incomprehensibility stems from many causes, and takes many forms. Regardless of cause, if the result is a document that cannot be read, it won't be.

Ugliness

The look of a document is another respect in which problems can be immediately obvious. I suspect that many authors do not realise how much impact defects in

[8] Or this opening sentence: "We have new networks of wireless like wired networks that our method makes use of in computers connected to each other but heterogeneously and distributed."

Or this one: "With the explosioning of documents on the internet, systems of finding documents that are the answers of users their queries have become important in the recent years."

Or this: "An ideal vector space is the base of IR research, so the basic problem of IR is to set up a suitable vector space, in this suitable vector space, query and document can be represented well by vectors."

presentation have on readers, but the message is clear: if something looks terrible, then the author doesn't care about the content; and if the author doesn't care, then the reader certainly shouldn't.

There are several common forms of this ugliness. One is in illustrations and tables: graphs that are badly designed or badly rendered, tables that are irregular or chaotic, diagrams in which the parts are unrelated, and so on. Another form is in layout, with, for example, absurdly sized headings or columns that overlap. A third form is the presence of dramatic formatting glitches, such as font and font size changing from paragraph to paragraph. Each of these conveys an impression of laziness.

Another kind of mistake conveys an impression of bad judgement: the decision to use inappropriate styles of presentation. The *comic sans* font has been widely mocked for its use in slides; it is even more mockable in a paper. Other examples include use of colours instead of italics for emphasis, comical drawings,[9] and peculiar over-the-top jokes.[10]

A more subtle form of ugliness is when a paper is dense with errors. These may be errors of fact, spelling errors, garbled citations, incomplete sentences, or any of a range of such things. They show that the author is indifferent to the work, and the reader will respond likewise.

Ignorance

All of the issues noted in this section make it difficult to see a paper as being of value, but, as a way of persuading the reader that a paper is worthless, nothing is more certain than a display of ignorance.

An example of this is when much of a paper is spent explaining an elementary concept that will be familiar to any likely reader and maybe even to undergraduates. While a few lines of review may be appropriate (to ensure that terminology is correctly understood, for example), why spend six pages of an algorithms paper explaining the difference between random-access memory and hard disks? Moreover, when the author gets the details wrong—and uses 1980s literature on memory technology in a 2000s paper, to consider one particular paper—the main effect is to reveal that the work is unreliable.

A similar example is when the author discusses at great length a statement that is either blindingly obvious or, worse, clearly false. "Web pages from a single website may be more like each other than pages drawn from different locations", besides being

---

[9] Particularly memorable (not in a good way) was a submission that included a photograph of Britain's Queen Elizabeth II, on which the author had superimposed a cartoon of a smiling mouth and a thumbs-up, to illustrate the Queen's happiness at the result of a successful Web search for "corgi". There was no mention of royalty or corgis in the text.

[10] Such as a paper in which the methods were named after kinds of duck, explanations made frequent (laboured) reference to duck behaviour, there were jokes about ducks (for example: Why did the duck cross the road? Because it wasn't chicken), the points in the graphs were duck shapes instead of circles or crosses, and one of the authors was allegedly Bill Feathertail of Poll Tree College.

clumsily worded, does not require fifteen hundred words of amplifying explanation. Some authors seem to like to make, and then explain, remarkably asinine observations, such as “Users today frequently use a search engine to answer queries”. But this is not as bad as asserting, say, that “with the growth of data online, users have given up trying to use Web search engines”, or that “the development of large RAM means that there is no longer a need for offline storage”.[11] Nor is it as bad as basing a program of research on a fallacy or delusion, such as “users download software illegally as a form of protest against corporate business practices”, or “recent advances in machine learning mean that soon it may no longer be necessary to write software”, or “increases in the efficiency of processors are encouraging people to develop their own database utilities, based on open-source toolkits”.[12]

What these examples have in common is that they are dogmatic—and in some instance, foolish—opinions expressed as fact. They embody a catastrophic misjudgement of the likely readership, and possibly of reality.

Another display of ignorance is in relation to past literature, as noted above. All too often, authors have either paid no attention to key past work, or have not troubled to understand it properly and dismiss it in a few words. It is astonishing how often authors don’t seem to have searched for related papers; a common exercise of mine is to paste a paper’s title directly in to a search engine, because highly relevant but uncited papers are often in the first page of results. When the background and literature review are crushed into a few brief paragraphs, it is almost certain that the author has nothing to contribute.

A “Writing Up” Checklist

Regarding *the scope of the work*,

- In what forum, or kind of forum, do you plan to publish?
- Is the scope of the work well defined?
- Is there a single, clearly articulated research question or goal? Have you identified which aspect of the work is of greatest impact, or of greatest interest?
- What would success in the project look like? What would failure look like? Can you anticipate the form of the outcomes in either case?

---

[11] Some papers in the area of in-memory databases rest on a similar assertion, namely that with the falling cost of memory technologies databases no longer need be maintained on disk. I’m intrigued by the fact that these claims have now been made for well over twenty years, a period in which memory and disk capacities have grown by a factor of at least 10,000. While there is some validity to the assertion—it is undoubtably true that some things can be done in memory today that required disk in the past—it does illustrate that such claims depend not just on technology, but also on the context in which the technology is used.

[12] If the gross logical failures of these statements are not obvious, pause here and take the time to analyze them yourself.

- Who is the readership? How deep or thorough will the background need to be to ensure that the readers fully appreciate the work?
- Do you and your co-authors have an agreed methodology for sharing the work of completing the write-up?
- Are the roles of the participants clear? What are your responsibilities? What activities will the others undertake?

Regarding *how the write-up is organized and presented*,

- What form will your write-up take? What other paper or thesis should your write-up resemble?
- Are you writing to a well-defined structure and organization? What are the sections, and how do they relate to each other?
- Do the title, abstract, and introduction appropriately set the context for the work?
- Have you identified a structure for the argument? A format for the results?
- Have you established a connection between the question, background, methods, and results? That is, have you identified the shape that the narrative will take?
- Is there anything unusual about the organization of the write-up, and, if so, is there a reason for this organization and how will it be explained to the reader?
- If you are writing a paper, are you working to defined length limits or a specified format?
- If you are writing a thesis, are there formatting requirements?

Regarding *your approach to the work*,

- Are you maintaining a log and notebook?
- Do you work to an explicit schedule with dates and targets?
- Do the deadlines leave enough time for your advisor to provide feedback on your drafts, or for your colleagues to contribute to the material?
- Do you have an effective approach to writing that lets you quickly generate text?
- How are results being selected for presentation? How do these results relate to your original aims? How do the selected results relate to the complete body of evidence you are gathering?
- Have the results been critically analyzed?
- In a thesis with multiple contributions, are they explicitly linked by an overarching goal?
- For a thesis, do you know how it will be examined? How is that knowledge shaping your writing?

Chapter 6
Good Style

Everything written with vitality expresses that vitality; there are no dull subjects, only dull minds.

Raymond Chandler
The Simple Art of Murder

It is a golden rule always to use, if possible, a short old Saxon word. Such a sentence as "so purely dependent is the incipient plant on the specific morphological tendency" does not sound to my ears like good mother-English—it wants translating.

Charles Darwin
Letter to John Scott

There are many ways in which an idea can be expressed in English; writing can be verbose or cryptic, flowery or direct, poetic or literal. The manner of expression is the writing style. Style is not about correct use of grammar, but about how well you communicate with likely readers.

Conventions and styles are valuable because some forms of presentation are difficult to understand or are simply boring. They are also valuable because conformity to commonly used styles reduces the effort required from readers. Breaking an established convention has the impact of this exclamation! It arrests attention and distracts from the message.

Science writing must by its nature be plain and straightforward—the need for it to be accurate and clear makes poetry inappropriate. But this does not mean that science writing has to be dull. It can have style, and moreover the desire to communicate clearly is not the only reason to make good use of English. Lively writing suggests a lively mind with interesting ideas to discuss.

In contrast, poor usage is distracting, suggests disorganized thinking, and prejudices readers against whatever is being said. It may seem unjust, but good writing and presentation can persuade readers that work is of value. Poorly presented material carries a strong subconscious message; for example, readers tend to mistrust statements if they contain numerous spelling errors. Layout issues such as font and spacing are also important: a lazy or amateurish presentation suggests to the reader that little care has been taken with the work.

This chapter, and Chaps. 7 and 8, concern writing style, including issues that are specific to science and general issues that many scientists ignore. Good style for

© Springer-Verlag London 2014

J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9_6

science is, ultimately, nothing more than writing that is easy to understand. Most of the points in these chapters are about the fundamental aims of science writing: to be clear, unambiguous, correct, interesting, and direct.

Economy

Text should be taut. The length of a paper should reflect its content—it is admirable to use only as many words as are required. Every sentence should be necessary. Papers are not made more important by padding with long-winded sentences; they are made less readable. In the following example, the italicized text can be discarded without affecting the intent.

> *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*.

Waffle, such as the italicized material above, is deadwood that readers must cut away before they can get to the meaning of the text.

Taut writing is a consequence of careful, frequent revision. Aim to delete superfluous words, simplify sentence structure, and establish a logical flow. That is, convey information without unnecessary dressing. Revise in a critical frame of mind, and avoid a sense of showing off or being clever. Be egoless—ready to dislike anything you have previously written. Expect to revise several times, perhaps many times.

If someone dislikes something you have written, remember that it is readers you need to please, not yourself. Again, it helps to set aside your ego. For example, when you are making changes to a paper in response to comments from a reader, you may find that the reader has made a claim that is wrong or meaningless. However, rather than telling yourself "the reader is wrong", ask yourself "what did I write that led the reader astray?" Even misguided feedback can tell you something about your writing.

Text can be condensed too far. Don't omit words that make the writing easier to understand.

✗ Bit-stream interpretation requires external description of stored structures. Stored descriptions are encoded, not external.

✓ Interpretation of bit-streams requires external information such as descriptions of stored structures. Such descriptions are themselves data, and if stored with the bit-stream become part of it, so that further external information is not required.

Tone

Science writing should be objective and accurate. Many of the elements that give literature its strength—nuance, ambiguity, metaphor, sensuality—are inappropriate for technical work. In contrast to popular science writing, the primary objective is to inform, not entertain. On the other hand, use of awkward, convoluted language is perhaps the most common fault in scientific writing; a direct, uncomplicated style is appropriate. Aim for austerity, not pomposity.

Simple writing follows from a few simple rules:

- Have one idea per sentence or paragraph and one topic per section.
- Have a straightforward, logical organization.
- Use short words.
- Use short sentences with simple structure.
- Keep paragraphs short.
- Avoid buzzwords, clichés, and slang.
- Avoid excess, in length or style.
- Omit unnecessary material.
- Be specific, not vague or abstract.
- Break these rules if there is a good reason to do so.

Sometimes a long word or a complex sentence is the best option. Use these when necessary, but not otherwise.

Another common fault in science writing is to overqualify, that is, to modify every claim with caveats and cautions. Such writing is a natural consequence of the scientist's desire to not make unfounded claims, but it can be taken too far.

✗ The results show that, for the given data, less memory is likely to be required by the new structure, depending on the magnitude of the numbers to be stored and the access pattern.

✓ The results show that less memory was required by the new structure. Whether this result holds for other data sets will depend on the magnitude of the numbers and the access pattern, but we expect that the new structure will usually require less memory than the old.

The first version is vague; the author has ventured an opinion that the new structure is likely to be better, but has buried it.

Use direct statements and expressions involving "we" or "I"—that is, the active voice—to make reading more pleasant and to help distinguish new results from old. (Voice is discussed later in this chapter.) There is nothing wrong with using a casual or conversational tone in technical writing, so long as it does not degenerate into slang.

Technical writing is not a good outlet for artistic impulses. The following is from a commercial software requirements document.

✗ The system should be developed with the end users clearly in view. It must therefore run the gamut from simplicity to sophistication, robustness to flexibility, all in the context of the individual user. From the first tentative familiarization steps, the consultation process has been used to refine the requirements by continued scrutiny and rigorous analysis until, by some alchemical process, those needs have been transmuted into specifications. These specifications distill the quintessence of the existing system.

The above extract has the excuse that it forms part of a sales pitch, but the following is from a scientific paper on concurrent database systems.

✗ 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.

Deciphering this paper was not easy. The following is a rough translation, with no guarantee that the intended meaning is preserved.

✓ We have seen that the usual assumptions lead to a tractable model, but this model is only a poor representation of real behaviour. We therefore proposed richer assumptions, which are however difficult to analyze. Now we consider whether there is any way in which our assumptions can be usefully applied.

Novice writers can be tempted to imitate the style of, not science writing, but popular science writing.

✗ As each value is passed to the server, the "heart" of the system, it is checked to see whether it is in the appropriate range.

✓ Each value passed to the central server is checked to see whether it is in the appropriate range.

Don't dress up your ideas as if they were on sale. In the following I have changed the author's name to "Grimwade".

✗ Sometimes the local network stalls completely for a few seconds. This is what we call the "Grimwade effect", discovered serendipitously during an experiment to measure the impact of server configuration on network traffic.

✓ Sometimes the local network stalls for a few seconds. We first noticed this effect during an experimental measurement of the impact of server configuration on network traffic.

But consider the following extract from a paper on some pragmatics for indexing, which illustrates that it is not necessary to write in a literary or pedantic style. It is colloquial, poorly punctuated, and there were spelling errors (not reproduced here), but it is direct and frank.

✓ To improve the chance of a cache hit almost a complete recode was necessary to the data structure routines but no run with the new code showed any improvement. The cache may have been too small but more likely the problem was the operating system and instruction prefetch getting the cache dirty. Also after recoding a couple of extra machine instructions were needed for each access so the saving of having a few more hits was lost.

For researchers educated in an English-speaking country, it is easy to forget that English is not the first language of a great many readers. Simple writing allows these readers to easily understand your work. Also, popular writing often uses shared cultural elements as references. Slang ("home run"), values ("cool"), analogies ("like turning left from the right lane"), and events ("the Northeastern power outage") may well be meaningless to readers living in other countries, or in other times. Even dates can be confusing: in the United States, dates are often written month–day–year, but elsewhere this notation almost invariably means day–month–year. Write for everybody.

Examples

Use an example whenever it adds clarification. A small example often means the difference between communication and confusion, particularly if the concept being illustrated is fundamental to understanding the paper. People learn by generalizing from concrete instances, and examples can give substance to abstract concepts.

✓ In a semi-static model, each symbol has an associated probability representing its likelihood of occurrence. For example, if the symbols are characters in text, then a common character such as "e" might have an associated probability of 12 %.

Each example should be an illustration of one concept; if you don't know what an example is illustrating, change it.

Examples can be blocks of text with a heading such as "Example 3.5" or detailed discussions of specific instances where a technique can be used, but often an informative example is just a few words.

✓ Large document collections, such as a repository of newspaper articles, can be managed with the same techniques.

✓ Special cases, such as the empty set, need to be handled separately.

✓ Algorithms that involve bit manipulation cannot be efficiently implemented in these languages. For example, Huffman coding is impractical because it involves processing a stream one bit at a time.

Motivation

Many authors take considerable trouble over the structure of their papers but don't make the structure obvious to the reader. Not only should the parts of a paper be ordered in a logical way, but this logic needs to be communicated.

The introduction usually gives some indication of the organization of the paper, by outlining the results and their context, and may include a list of the parts of the paper, but these measures by themselves are not sufficient. Brief summaries at the start and end of each section are helpful, as are sentences connecting one section to the next; for example, a well-written section might conclude with:

✓ Together these results show that the hypothesis holds for linear coefficients. The difficulties presented by non-linear coefficients are considered in the next section.

Link text together as a narrative—each section should have a clear story to tell. The connection between one paragraph and the next should be obvious. This principle is sometimes expressed as: Tell the reader what you are going to say, then say it, and then tell the reader that you have said it.

A common error is to include material such as definitions or theorems without indicating why the material is useful. Usually the problem is lack of explanation; sometimes it is symptomatic of an ordering problem, such as including material before the need for it is obvious. Never assume that a series of definitions, theorems, or algorithms—or even the need for the series—is self-explanatory. Motivate the reader at each major step in the exposition: explain how a definition (theorem, lemma, whatever) is to be used, or why it is interesting, or how it fits into the overall plan.

The authors of a paper are almost always better informed than their readers. Even expert readers won't be familiar with some of the details of a problem, whereas the author has probably been studying the problem intimately for months or years, and takes many difficult issues for granted. You should explain everything that is not common knowledge to the paper's readership; what constitutes common knowledge depends on the paper's subject and on where it is published. At each part of a paper you should consider what the reader has learnt so far, whether this knowledge is sufficient to allow understanding of what follows, and whether each part follows from what has already been said.

Motivation is essential, but do motivate the right thing. Don't, say, set the scene by explaining that certain algorithms are too slow for massive databases, and then test your method on a few thousand records; or argue that "Web search needs to be semantically aware" and then propose the use of spelling correction to amend queries. The big-picture topic may well be the inspiration for your overall work, but that does not mean that it is necessarily the right inspiration for a particular paper.[1]

[1] And don't let the motivation take over. In a paper on routing of traffic in a local network, the author was attempting to introduce the topic of congestion measurement, but digressed into a history of network topology, configuration, scale, and hardware. It took two pages to reach what appeared to

Balance

Within a paper, each topic should be discussed to a similar depth. A paragraph on a previous algorithm followed by seven pages on your refinements to it is unbalanced. If one paper merits half a page, other papers of equal relevance should not be dismissed in a line. An algorithm that is only sketched does not merit twenty graphs and tables; an algorithm that is described in detail needs a substantial analysis or other justification. A four-page rambling introduction is unlikely to be readable.

The length of a paper is a consequence of how much material is included and of how much detail is given, that is, the depth to which each topic is discussed. When a paper must be kept within a length limit, some compromise is required. Some of the discussion must be omitted, or the graphs selected more carefully, or the text condensed. Perhaps it will even be necessary to omit a proof or a series of results. Such changes should not be used as an excuse for unbalancing the paper.

Voice

Avoid excessive use of indirect statements (*passive voice*), particularly descriptions of actions that don't indicate who or what performs them.

✗ The following theorem can now be proved.

✓ We can now prove the following theorem.

The direct style (*active voice*) is often less stilted and easier to read.

Another unpleasant indirect style is the artificial use of verbs like "perform" or "utilize", perhaps in the false belief that such writing is more precise or scientific. These words can often be removed.

✗ Tree structures can be utilized for dynamic storage of terms.

✓ Terms can be stored in dynamic tree structures.

✗ Local packet transmission was performed to test error rates.

✓ Error rates were tested by local packet transmission.

Other words often used in this way include "achieved", "carried out", "conducted", "done", "occurred", and "effected".

Change of voice sometimes changes meaning and often changes emphasis. If passive voice is necessary, use it. Complete absence of active voice is unpleasant, but that does not mean that all use of passive voice is poor.

Use of "we" is valuable when trying to distinguish between the contribution made in your paper and existing results in a field, particularly in an abstract or introduction.

(Footnote 1 continued)
be the goal, which was to introduce new approaches to measurement of packet delays. A sentence would have been sufficient.

For example, in “it is shown that stable graphs are closed”, the reader may have difficulty deciding who is doing the showing, and in “it is hypothesized that”, the reader will be unsure whether the hypothesis was posed in your paper or elsewhere. Use of “we” can allow some kinds of statements to be simplified—consider “we show” versus “in this paper it is shown that”. “We” is preferable to pretentious expressions such as “the authors”.

Some authors use phrases such as “this paper shows” and “this section argues”. These phrases, with their implication that the paper, not the author, is doing the arguing, should generally be avoided.

In some cases the use of “we” is wrong.

✗ When we conducted the experiment it showed that our conjecture was correct.

Here, the use of “we” seems to hint that the outcome might be different if the experiment was run by someone else.

✓ The experiment showed that our conjecture was correct.

I do not particularly like the use of “I” in scientific writing, except when it is used to indicate that what follows is the author’s opinion. The use of “I” in place of “we” in papers with only one author is uncommon.

Use of personal pronouns has been a contentious issue in technical writing. Some people argue that it undermines objectivity by introducing the author’s personality and is therefore unacceptable, even unscientific. Others argue that to suggest that a paper is not the work of individuals is intellectually dishonest, and that use of personal pronouns makes papers easier to read. Although opinions on this topic are divided, use of “we” is an accepted norm.

The Upper Hand

Some authors seem to have a superiority complex—a need to prove that they know more or are smarter than their readers. Perhaps the most appropriate word for this behaviour is swagger. One form of swagger is implying familiarity with material that most scientists will never read; an example is reference to philosophers such as Wittgenstein or Hegel, or statements such as “the argument proceeds on Voltarian principles”. Another form is the unnecessary inclusion of difficult mathematics, or offhand remarks such as “analysis of this method is of course a straightforward application of tensor calculus”. Yet another form is citation of obscure, inaccessible references.

This kind of showing off, of attempting to gain the upper hand over the reader, is snobbish and tiresome. Since the intention is to make statements the reader won’t understand, the only information conveyed is an impression of the author’s ego. Write for an ordinary reader, as your equal.

Obfuscation

Obfuscation is the making of statements in ambiguous or convoluted terms, with the intention of hiding meaning, or of appearing to say much while actually saying little. It can be used, for example, to give the impression of having done something without actually claiming to have done it.

✗ Experiments, with the improved version of the 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.

Note the use of bland statements such as "experiments… are the step that confirms our speculation" (true, but not informative) and "improvements lead to better performance" (tautologous). The implication is that experiments were undertaken, but there is no direct claim that experiments actually took place.

In science writing, vague statements are regrettably common. It is always preferable to be specific: exceptions are or are not possible, data was transmitted at a certain rate, and so on. Stating that "there may be exceptions in some circumstances" or "data was transmitted fast" is not helpful.

✗ Amelioration can lead to large savings.

✓ Amelioration led to savings of 12 %–33 % in our experiments.

Obfuscation can arise in other ways: exaggeration, omission of relevant information, or bold statements of conclusions based on flimsy evidence. Use of stilted or long-winded sentences—often due to an unnecessary attempt to introduce formality—can obfuscate.

✗ 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.

Some obfuscation arises because processes are unnecessarily complex, are presented in unnecessary detail, or are outright unnecessary. The following was written as part of a tender process.

✗ These draft guidelines are part of a process for seeking comments on the proposed stages for identifying the officers responsible for participating in the development of the initial specification.

Analogies

Analogies are curious things: what seems perfectly alike or parallel to one person may seem entirely unalike to another.

✗ Writing a program is like building a model with connector blocks.

What are "connector blocks"? How are they like programming? Even if the similarity is obvious to a programmer, is it obvious to a novice? This analogy (made in an introductory computer science textbook) seems to me to fail because it captures neither logic nor repetition. For an analogy to be worthwhile, it should significantly reduce the work of understanding the concept being described.

Another drawback to analogies is that they can take your reasoning astray—two situations with marked similarities may nonetheless have fundamental differences that the analogy leads you to ignore. I have seen more bad analogies than good in computing research papers; however, simple analogies can undoubtedly help illustrate unfamiliar concepts.

✓ Contrasting look-ahead graph traversal with standard approaches, look-ahead uses a bird's-eye view of the local neighbourhood to avoid dead ends, but at significant cost: it is necessary to feed the bird and wait for it to return after each observation.

Beware of analogies with situations that may be unfamiliar to the reader.

✗ One-sided protocols are like signals in football.

Straw Men

A straw man is an indefensible hypothesis that an author describes for the sole purpose of criticizing it. A paraphrasing of an instance in a published paper is "it can be argued that databases do not require indexes", in which the author and reader are well aware that a database without an index is as practical as a library without a catalogue.

Such writing says more about the author than it does about the subject. For example, occasionally an author will write something like:

✗ As the scale of data on the Web grows to billions of pages, searchers can no longer find answers to queries.

Sweeping statements of this kind, in this case thoroughly contradicted by our daily experience of search, tell the reader that the work is not based on rigorous argument or clear thinking.

Another form of straw man is the contrasting of a new idea with some impossibly bad alternative, to put the new idea in a favourable light. This form is obnoxious because it can lead the reader to believe that the impossibly bad idea might be worthwhile, and that the new idea is more important than is in fact the case. Contrasts should be between the new and the current, not the new and the fictitious.

✗ Query languages have changed over the years. For the first database systems there were no query languages and records were retrieved with programs. Before then data was kept in filing cabinets and indexes were printed on paper. Records were retrieved by getting them from the cabinets and queries were verbal, which led to many mistakes being made. Such mistakes are impossible with new query languages like QIL.

A more subtle form of straw man is comparison between the new and the ancient. For example, criticisms based on results in old papers are sometimes unreasonable, because many of the factors that affected the results (architecture, scale, kinds of data, beliefs about algorithms, and so on) have changed dramatically in the meanwhile. Likewise, decisions that look poor in retrospect may have been perfectly rational at the time.

Historical fallacies are another form of the same issue.

✗ Since the invention of the internet, researchers have been using the Web to publish data.

The Web was invented in 1989 and became generally available within academia in 1993 or so; the internet as we now know it began to develop in the 1970s and was in wide use long before 1990. And the two are not equivalent—one is infrastructure, the other is traffic.

✗ Researchers working on computer image generation initially failed to consider the benefits of parallel processing.

Considering that those researchers were working in the 1970s, they were unlikely to have had access to more than a single, sequential computer.

A straw man is an example of rhetoric—of attempting to win an argument through presentation rather than reasoning. Other forms of rhetoric are appeal to authority, appeal to intuitively obvious truth, and presentation of received wisdom as fact.

✗ We did not investigate partial interpretation because it is known to be ineffective.

If there is evidence—a study or proof, not someone else's opinion—then cite it. Unsubstantiated claims should be clearly noted as such, not dressed up as accepted knowledge.

✗ Most users prefer the graphical style of interface.

✓ We believe that most users prefer the graphical style of interface.

✗ Another possibility would be a disk-based method, but this approach is unlikely to be successful.

✓ Another possibility would be a disk-based method, but our experience suggests that this approach is unlikely to be successful.

Avoid nonsense, absurdity, and over-generalization.

✗ Execution was almost instantaneous.

✗ The Web is infinitely large.

✗ There is no limit to the possible efficiency gains.

Reference and Citation

You need to explain the relationship of your new work to existing work, showing how your work builds on previous knowledge and how it differs from contributions in other, relevant papers. The existing work is identified by reference to published theses, articles, and books. All papers include a bibliography, that is, a list of such references in a standardized format, and embedded in each paper's text there are citations to the publications.

References, and discussion of them, serve three main purposes. They help demonstrate that work is new: claims of originality are much more convincing in the context of references to existing work that (from the reader's perspective) appears to be similar. They demonstrate your knowledge of the research area, which helps the reader to judge whether your statements are reliable. And they are pointers to background reading.

Before including a reference, consider whether it will be of service to the reader. A reference should be relevant, up-to-date, and reasonably accessible; and it should be necessary. Don't add citations just to pad the bibliography. Refer to an original paper in preference to a secondary source; to well-written material in preference to bad; to a book, conference paper, or journal article in preference to a workshop paper; to a workshop paper in preference to a manuscript (which have the disadvantage of being unrefereed); and to formally published documents rather than Web pages. Avoid reference to private communications and information provided in forums such as seminars or talks—such information cannot be accessed or verified by the reader. In the rare circumstance that you must refer to such material, do so via a footnote, parenthetical remark, or acknowledgement.

If you discuss a paper or note some particular contribution it makes, it must be cited. Otherwise, consider whether a reader needs the paper for knowledge in addition to that in the other papers you cite. If the answer is no, perhaps it should be omitted. At the same time, ensure that it is clear to the reader that you know all the pertinent background literature.

Don't cite to support common knowledge. For example, use of a binary tree in an algorithm doesn't require a reference to a data structures text. But claims, statements of fact, and discussion of previous work should be substantiated by reference if not substantiated within your write-up. This rule even applies to minor points. For some readers the minor points could be of major interest.

In many papers, some of the references are to previous publications by the same author. Such references establish the author's credentials as someone who understands the area, establish a research history for the paper, and allow the interested reader to gain a deeper understanding of the research by following it from its inception. Gratuitous self-reference, however, undermines these purposes; it is frustrating for readers to discover that references are not relevant.

On rare occasions it is necessary to refer to a result in an inaccessible paper. For example, suppose that in 1981 Dawson wrote “Kelly (1959) shows that stable graphs are closed”, but Kelly (1959) is inaccessible and Dawson (1981) does not give the details. In your write-up, do not refer directly to Kelly—after all, you can’t check the details yourself, and Dawson may have made a mistake.

✓ According to Dawson (1981), stable graphs have been shown to be closed.

✓ According to Kelly (1959; as quoted by Dawson 1981), stable graphs are closed.

The second form tells readers who originated the result without the effort of obtaining Dawson first. Kelly’s entry in the bibliography should clearly show that the reference is second-hand.

Regardless of whether you have access to original sources, be careful to attribute work correctly. For example, some authors have referred to “Knuth’s Soundex algorithm”, although Knuth is not the author and the algorithm was at least fifty years old when Knuth discussed it.

Some readers of a paper will not have access to the publications it cites, and so may rely on the paper’s description of them. For this reason alone you should describe results from other papers fairly and accurately. Any criticisms should be based on sound argument. That is, it is acceptable to make reasoned criticisms, and a careful assessment of past work is of great value because ultimately it is how a paper is regarded that determines its worth. However, only rarely is it acceptable to offer opinions, and it is never acceptable to use flattery or scorn. Neither belittle papers, regardless of your personal opinion of their merits, nor overstate their significance; and beware of statements that might be interpreted as pejorative.

✗ Robinson’s theory suggests that a cycle of handshaking can be eliminated, but he did not perform experiments to confirm his results [22].

✓ Robinson’s theory suggests that a cycle of handshaking can be eliminated [22], but did not report experimental confirmation.

✓ Robinson’s theory suggests that a cycle of handshaking can be eliminated [22], but as yet there is no experimental confirmation.

Careful wording is needed in these circumstances. When referring to the work of Robinson, you might write that “Robinson thinks that…”, but this implies that you believe he is wrong, and has a faint odour of insult; you might write that “Robinson has shown that…”, but this implies that he is incontrovertibly right; or you might write that “Robinson has argued that…”, but then should make clear whether you agree.

A simple method of avoiding such pitfalls is to quote from the reference, particularly if it contains a short, memorable statement—one or two sentences, say—that is directly pertinent. Quotation also allows you to clearly distinguish between what you are saying and what others have said, and is far preferable to plagiarism.

Cited material often uses different terminology, spelling, or notation, or is written for an entirely different context. When you use results from other papers, be sure to show the relationship to your own work. For example, a reference might show a general case, but you use a special case; then you need to show that it is a special case. If you claim that concepts are equivalent, ensure that the equivalence is clear to the reader.

References that are discussed should not be anonymous.

✗ Other work [16] has used an approach in which…

✓ Marsden [16] has used an approach in which…
Other work (Marsden 1991) has used an approach in which…

The modified versions provide more information to the reader, and "Marsden" is easier to remember than "[16]" if the same paper is discussed later on.

Likewise, self-references should not be anonymous—it should be clear to the reader that references used to support your argument are your own papers, not independent authorities.

✗ Smith et al. [10] found compressed lists to be…

✓ In Smith et al. [10], we found compressed lists to be…

Other references that are not discussed can just be listed.

✓ Better performance might be possible with string hashing techniques that do not use multiplication [11, 30].

Avoid unnecessary discussion of references.

✗ Several authors have considered the problem of unbounded delay. We cite, for example, Hong and Lu (1991) and Wesley (1987).

✓ Several authors have considered the problem of unbounded delay (Hong and Lu 1991; Wesley 1987).

Two styles of citation are illustrated above. One is the ordinal-number style, in which entries in the reference list are numbered and are cited by their number, as in "… is discussed elsewhere [16]". The other is the name-and-date or Harvard style—my preferred style—in which entries are cited by author name using either square or round brackets:

✓ … is discussed by Whelks and Babb (1972).

✓ … is discussed elsewhere (Whelks and Babb 1972).

✓ … is discussed by Whelks and Babb [1972].

✓ … is discussed elsewhere [Whelks and Babb 1972].

A third common style is to use superscripted ordinal numbers, as in "… is discussed elsewhere[16]". Another style is use of uppercase abbreviations, where references are denoted by strings such as "[MAR91]". This is not a good style: the abbreviations seem to encourage poor writing such as "… is discussed in [WHB72]" and, because uppercase characters stand out from text, they are rather distracting.

Note, however, that many publishers insist on a particular style. (Some also insist that bibliographic entries be ordered alphabetically, which is convenient for the reader, or that they appear by order of citation, which is convenient for traditional typesetting.) Your writing should be designed to survive a change in the style of citation.

When discussing a reference with more than two authors, all but the first author's name can be replaced by "et al."

✓ Howers, Mann, Thompson, and Wills [9] provide another example.

✓ Howers et al. [9] provide another example.

In a variant of this style, the full list is given at the first citation, and the abbreviated form thereafter. Note the stop: "et al." is an abbreviation.

Each entry in the reference list should include enough detail to allow readers to find the paper. Other than in extreme cases, the names of all authors should usually be given—don't use "et al." in the reference list. Be guided by the common practices in your research area.[2]

Format fields of the same type in the same way. For example, don't list one author as "Heinrich, J.", the next as "Peter Hurst", the next as "R. Johnson", and the next as "SL Klows". Capitalization, explained in Chap. 8, should be consistent. Don't use unfamiliar abbreviations of journal names. (One that has puzzled me is "*J. Comp.*")

Journal articles. The journal name should be given in full, and author names, paper title, year, volume, number, and pages must be provided. Consider also giving the month. Thus:

✗ T. Wendell, "Completeness of open negation in quasi-inductive programs", *J. Dd. Lang.*, 34.

is inadequate. Revise it to, say:

✓ T. Wendell, "Completeness of open negation in quasi-inductive programs", *ICSS Journal of Deductive Languages*, 34(3):217–222, November 1994.

Conference papers. The conference name should be complete, and authors, title, year, and pages must be provided. Information such as publisher, conference location, month, and editors should also be given.

Books. Give title, authors, publisher, year, and, where relevant, edition and volume. If the reference is to a specific part of the book, give page numbers; for example,

[2] An exception is the rare case in which the authors list themselves as "et al." I have only seen one paper with such an author list: "The Story of O_2" by O. Deux et al.

write "(Howing 1994; pp. 22–31)" rather than just "(Howing 1994)". If the reference is to a chapter, give its title, pages, and, if applicable, authors.

Technical reports. In addition to title, authors, year, and report number, you need to provide the address of the publisher (which is usually the authors' home institution). If the report is available online, give its URL. Given that technical reports are now so rare, and are relatively unreliable as a source of scientific knowledge, consider removing such references from the bibliography and citing directly from the text via a footnote.

Web pages. If you cite a Web page, attempt to find a durable URL that is unlikely to change if, for example, a researcher changes institution. In addition to the usual details, give the URL; they can include unusual characters, so make sure you represent these correctly. It is a preferred practice to also include the date on which the Web page was accessed. However, as for technical reports, it is usually better to refer to such material in a footnote rather than as a formal entry in a bibliography. You should still provide sufficient details, of course, and a durable URL.

Personal communications. An email from a colleague is not a durable document, and should never be cited. If you have to mention it, use a footnote.

All references. Take particular care to provide as much information as possible. If the reference is obscure, but you feel that you must refer to the First Scandinavian Workshop on Backward Compatibility, consider explaining how to obtain the proceedings or a copy of the paper. Any of these kinds of references might have a DOI, or some other form of permanent URL. If such a URL is available, you should include it.

Punctuation of citations is considered in Chap. 8, and ethical issues with regard to citations and references are discussed in Chap. 17.

Quotation

Quotations are text from another source, usually included in a paper to support an argument. The copied text, if short, is enclosed in double quotes (which are more visible than single quotes and cannot be confused with apostrophes). Longer quotes are set aside in an indented block.

✓ Computer security forensics is "the study of matching an intrusion event to an IP address, location, and individual" (Brinton 1997).

✓ As described by Kang [16], there are three stages:

> First, each distinct word is extracted from the data. During this phase, statistics are gathered about frequency of occurrence. Second, the set of words is analyzed, to decide which are to be discarded and what weights to allocate to those that remain. Third, the data is processed again to determine likely aliases for the remaining words.

The quoted material should be an exact transcription of the original text; some syntactic changes are permissible, so long as the meaning of the text is unaltered, but the changes should be held to a minimum. Changes of font, particularly addition of emphasis by changing words to italics, should be explicitly identified, as should changes of nomenclature.

The expression “[sic]” is used to indicate that an error is from the original quote, as in “Davis regards it as ‘not worty [sic] of consideration’ [11]”. It is not polite to point out such errors, which are insignificant and can be silently corrected; avoid such use of “[sic]”, and, perhaps, avoid quotes that seem to require it. More rarely “[sic]” is used to indicate that terminology or jargon is being used in a different way.

✗ Hamad and Quinn (1990) show that “similarity [sic] is functionally equivalent to identity”; note that similarity in this context means homology only, not the more general meaning used in this paper.

The long explanation renders the quote pointless.

✓ Hamad and Quinn (1990) show that homology “is functionally equivalent to identity”.

Moreover, for a short, natural statement of this kind the quotes are not essential.

✓ Hamad and Quinn (1990) show that homology is functionally equivalent to identity.

Other changes are insertions, replacements, or remarks, delimited by square brackets; and short omissions, represented by ellipses. In strict usage, the ellipses are themselves placed in square brackets, to make it clear they were not in the original text.

✓ They describe the methodology as “a hideous mess […] that somehow manages to work in the cases considered [but] shouldn’t”.

(Note that an ellipsis consists of three stops, neither more nor less.) Ellipses are unnecessary at the start of quotes, and at the end of quotes except where they imply “et cetera” or “and so on”, or where the sentence is left hanging. For long omissions, don’t use an ellipsis; separate the material into two quotations. Material in square brackets is used for comments or to make the quote parse when read in its new context.

Don’t mutilate quotations.

✗ According to Fier and Byke such an approach is “simple and … fast, [but] fairly crude and … could be improved” [8].

It would be better to paraphrase.

✓ Fier and Byke describe the approach as simple and fast, but fairly crude and open to improvement [8].

Long quotations, and quotation in full of material such as algorithms or figures, require permission from the publisher and from the author of the original. (Plagiarism and inappropriate quotation are discussed in Chap. 17.)

Words can be quoted to show that they are inadequately defined.

✗ This language has more “power” than the functional form.

Here the author must assume that “power” will be understood in a consistent way by the reader. Such use of quotes indicates woolly thinking—that the author is not quite sure what “power” means, for example.

✓ This language allows simpler expression of queries than does the functional form.

More rarely, words can be quoted to indicate irony. The expression “in their ‘methodology’ ” can be interpreted as *in their so-called methodology*, and is therefore insulting. This is not an appropriate use of quotes.

Acknowledgements

In the acknowledgements of a write-up you should thank everyone who made a contribution, whether advice, proofreading, coding, or whatever: include research students, research assistants, technical support, and colleagues. Funding sources should also be acknowledged. It is usual to thank only those who contributed to the scientific content—don’t thank your parents or your cat unless they really helped with the research. Books and theses often have broader acknowledgements, however, to include thanks for people who have helped in non-technical ways. Consider showing your acknowledgement to the people you wish to thank, in case they object to the wording or to the presence of their name in the paper.

There are two common forms of acknowledgement. One is to simply list the people who have helped with the paper.

✓ I am grateful to Dale Washman, Kim Micale, and Dong Wen. I thank the Foundation for Science and Development for financial support.

Even in this little example there is some scope for bruised egos—Kim might wonder why Dale was listed first, for example.

The other common form is to explain each person’s contribution. On the one hand, don’t make your thanks too broad; if Kim and Dong constructed the proof, why aren’t they listed as authors? On the other hand, too much detail can damn with faint praise.

✗ I am grateful to Dale Washman for discussing aspects of the proof of Proposition 4.1, to Kim Micale for identifying some technical errors in Theorem 3, and to Dong Wen for helping with use of the debugging tools. I thank the Foundation for Science and Development for a year of financial support.

✓ I am grateful to Dale Washman and Kim Micale for our fruitful discussions, and to Dong Wen for programming assistance. I thank the Foundation for Science and Development for financial support.

This form has the advantage of identifying which of your colleagues contributed to the intellectual content.

Some authors write their thanks as “I would like to thank” or “I wish to thank”. To me this seems to imply that *I wish to thank … but for some reason I am unable to do so*. Consider instead using “I am grateful to” or simply “I thank” or “Thanks to”.

Grammar

In this book I have not given advice on grammar, because the clarity of writing largely depends on whether it conforms to accepted usage. One aspect of grammar is, however, worth considering: that some people use what they believe to be traditional grammar to criticize other people’s text, based on rules such as *don’t split infinitives* or *don’t begin a sentence with “and” or “but”*. I dislike this attitude to writing: grammatical rules should be observed, but not at the cost of clarity or meaning. Be aware, though, that an excess of sloppy grammar annoys some readers.

Beauty

Authors of style guides like to apply artistic judgements to text. This does not mean that scientific writing should be judged as literary prose; indeed, such prose would be inappropriate. But some authorities on writing style argue that text should be crystalline, transparent, and have good rhythm and cadence; and that one should dislike stuffy, soft, stodgy, and sagging sentences.

How useful such judgements are to most writers is not clear. Doubtless, well-crafted text is a pleasure to read, ill-written text can be hard going, and good rhythm in text helps us to parse. But appreciation of well-written text does not always help a novice to write it, nor is it evident that, to a poor writer, the argument that text should be elegant is meaningful. It is sufficient to aim for simplicity and clarity.

Chapter 7
Style Specifics

Those complicated sentences seemed to him very pearls… "The reason for the unreason with which you treat my reason, so weakens my reason that with reason I complain of your beauty" … These writings drove the poor knight out of his wits.

Cervantes
Don Quixote

Underneath the knocker there was a notice that said:

PLES RING IF AN RNSER IS REQIRD

Underneath the bell-pull there was a notice that said:

PLEZ CNOKE IF AN RNSR IS NOT REQID

These notices had been written by Christopher Robin, who was the only one in the forest who could spell.

A.A. Milne
Winnie the Pooh

Good style is about clear, easy-to-read writing, which can be achieved by following well-defined guidelines. These are not arbitrary rules, but are principles that experienced writers follow. In the previous chapter, some of these principles were reviewed. This chapter concerns a range of specific problems that are common in technical writing.

Titles and Headings

Titles of papers and sections should be concise and informative, have specific rather than general terms, and accurately describe the content. Complicated titles with long words are hard to digest.

✗ A New Signature File Scheme Based on Multiple-Block Descriptor Files for Indexing Very Large Data Bases

✓ Signature File Indexes Based on Multiple-Block Descriptor Files

© Springer-Verlag London 2014

J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9_7

✗ An Investigation of the Effectiveness of Extensions to Standard Ranking Techniques for Large Text Collections

✓ Extensions to Ranking Techniques for Large Text Collections

Don't make the title so short that it is contentless. "Limited-Memory Huffman Coding for Databases of Textual and Numeric Data" is awkward, but it is superior to "Huffman Coding for Databases", which is far too general.

Accuracy is more important than catchiness—"Strong Modes Can Change the World!" is excessive, not to mention uninformative. The more interesting the title, however, the more likely that the text underneath it will be read. The title is the only part of your paper that most people see; if the title does not reflect the paper's contents, the paper will not be read by the intended readership.

The title is meant to capture something of the flavour of the contribution, and should not be misleading as to the scope or outcome of the work. For example, titles that begin "Towards ..." can be disconcerting. If the title of the paper is "Towards Effective Blog Search", it suggests that effective blog search is not actually achieved in the paper, or that only one specific part of the broader problem has been tackled. It would make more sense to capture just that specific part of the problem in the title, and consider the broader context only when discussing the motivation for the work.

Titles and section headings do not have to be complete sentences; indeed, such titles can look rather odd.

✗ Duplication of Data Leads to Reduction in Network Traffic

✓ Duplication of Data to Reduce Network Traffic

Section headings should reflect the paper's structure. If a section is headed "Lists and Trees" and the first subsection is "Lists", another should be "Trees"; don't use, say, "Other Data Structures". If a section is headed "Index Organizations" the subsection heading should probably be "B-trees" rather than "B-tree indexes".

Headings may or may not be numbered. In a paper, my preference is to use only two levels of headings, major and minor, and to only number major headings. In a thesis, numbered chapters, sections, and possibly subsections, are appropriate. Deeper numbering allows more precise referencing, but often seems fussy. If all headings are unnumbered—as is required in some journals—make sure that major and minor headings are clearly distinguished by font, size, or placement.

Also, headings may or may not be displayed, that is, on a line by themselves. Typically, major headings—of say chapters, sections, or subsections—are displayed, while in-line headings may be used for briefer segments of text where some naming of the content is needed. For example, in-line headings are used in the section "Misused words" of this chapter.

A paper (or thesis chapter) consists of sections and possibly subsections. There is rarely any need to break subsections into sub-subsections. Avoid breaking text into small blocks; three displayed headings on a page is too many. Headings below the level of subsections should be in-line, not lines by themselves (but note that publishers often have specific rules on such aspects of formatting). Beware of having too few headings, though, because it is difficult to maintain the logical flow of a section over more than a few pages.

The Opening Paragraphs

The opening paragraphs can set the reader's attitude to the whole paper or thesis, so begin well. All of a document should be created and edited with care, but take the most care with the opening, to create the best possible first impression. The abstract should be written especially well, without an unnecessary word, and the opening sentence should be direct and straightforward.

✗ Trees, especially binary trees, are often applied—indeed indiscriminately applied—to management of dictionaries.

✓ Dictionaries are often managed by a data structure such as a tree. However, trees are not necessarily the best choice for this application.

The following, which was the first sentence of a paper, is an example of how to begin badly.

✗ This paper does not describe a general algorithm for transactions.

Only later does the reader discover than the paper describes an algorithm for a special case.

✓ General-purpose transaction algorithms guarantee freedom from deadlock, but can be inefficient. In this paper, we describe a new transaction algorithm that is particularly efficient for a special case, the class of linear queries.

The first paragraphs must be intelligible to any likely reader; save technicalities for later on, so that readers who don't understand the details of your paper are still able to appreciate your results and the importance of your work. That is, describe what you have done without the details of how it was done.

Starting an abstract or introduction with "This paper concerns" or "In this paper" often means that results are going to be stated out of context.

✗ In this paper we describe a new programming language with matrix manipulation operators.

✓ Most numerical computation is dedicated to manipulation of matrices, but matrix operations are difficult to implement efficiently in current high-level programming languages. In this paper we describe a new programming language with matrix manipulation operators.

The second version describes the context of the paper's contribution.

Beginning a paper by stating that a topic is popular or that a problem is important is flat and uninspiring.

✗ Use of digital libraries is increasingly common.

✗ It is important that the cost of disk accesses be reduced in query processing.

Such openings succeed in establishing context but fail in motivation, often because they are an assertion that a reasonable person might disagree with. A simpler or more specific statement may well be preferable.

✓ Digital libraries provide fast access to large numbers of documents.

✓ Query processing can involve many disk accesses.

A typical organization for the introduction of a paper is to use the first paragraphs to describe the context. It is these paragraphs that convince the reader that the paper is likely to be interesting. The opening sentences should clearly indicate the topic.

✗ Underutilization of main memory impairs the performance of operating systems.

✓ Operating systems are traditionally designed to use the least possible amount of main memory, but such design impairs their performance.

The second version is better for several reasons. It is clear; it states the context, which can be paraphrased as *operating systems don't use much memory*; and, in contrast to the first version, it is positive.

Take care to distinguish description of existing knowledge from the description of the paper's contribution.

✗ Many user interfaces are confusing and poorly arranged. Interfaces are superior if developed according to rigorous principles.

✓ Many user interfaces are confusing and poorly arranged. We demonstrate that interfaces are superior if developed according to rigorous principles.

In most papers, the introduction should not flow on from the abstract, which is a summary of a paper rather than its opening. The paper should be complete even with the abstract removed.

Variation

Diversity—in organization, structure, length of sentences and paragraphs, and word choice—helps to keep the reader's attention.

✗ The system of rational numbers is incomplete. This was discovered 2000 years ago by the Greeks. The problem arises in squares with sides of unit length. The length of the diagonals of these squares is irrational. This discovery was a serious blow to the Greek mathematicians.

✓ The Greeks discovered 2000 years ago that the system of rational numbers is incomplete. The problem is that some quantities, such as the length of the diagonal of a square with unit sides, are irrational. This discovery was a serious blow to the Greek mathematicians.

Note how, in the second version, the final statement is more effective although it hasn't been changed.

Paragraphing

A paragraph should discuss a single topic or issue. The outline or the argument is typically captured in the first sentence of each paragraph, with the rest of the paragraph used for amplification or example. Every sentence in a paragraph should be on the topic announced in the opening. The last sentence has higher impact than those in the body, so pay attention to sentence order.

Long paragraphs can indicate that several lines of argument are being followed simultaneously. If a long paragraph can be broken, break it. Lack of variation in paragraph length makes the page monotonous, however, so don't divide your text into paragraphs of uniform size.

Contextual information can be forgotten between paragraphs, and references between paragraphs can be difficult to follow. For example, if a paragraph discusses a fast sorting algorithm, the next paragraph should not begin "This algorithm" but rather "The fast sorting algorithm"; if one paragraph refers to Harvey, the next should not refer to "his" but rather to "Harvey's". Link paragraphs by re-use of key words or phrases, and with expressions that connect the content of one paragraph to that of the next.

Formatted lists can be used as an occasional alternative to paragraphs. Lists are useful for the following reasons:

- They highlight each main point clearly.
- The context remains obvious, whereas, in a long list of points made in a paragraph, it is hard to tell whether the later points are part of the original issue or belong to some subsequent discussion.
- An individual point can be considered in detail without confusing the main thread of narrative.
- They are easy to refer to.

List points can be numbered, named, or tagged. Use numbers only when ordering or reference is important. If it is necessary to refer to an individual point, use numbers or names. Otherwise use tags, as in the list above. Acceptable tags are bullets and dashes; fancy symbols such as $\hookrightarrow$, $\star$, or graphic icons look childish.

A disadvantage of lists is that they highlight rather too well: a list of trivia can be more attention-getting than a paragraph of crucial information. Reserve lists for material that is both significant and in need of enumeration.

Ambiguity

Check carefully for ambiguity. It is often hard to detect in your own text because you know what is intended.[1]

✗ The compiler did not accept the program because it contained errors.

✓ The program did not compile because it contained errors.

The next example is from a manual.

✗ There is a new version of the operating system, so when using the "fetch" utility, error messages can be ignored.

✓ There is a new version of the operating system, so the "fetch" utility's error messages can be ignored.

Part of the confusion comes from the redundant phrase "when using": there would be no error messages if the utility was not being used.

When using pronouns such as "it", "this", and "they", ensure that the reader knows what is being referred to.

✗ The next stage was the test of the complete system, but it failed.

What failed, the test or the system?

✗ In addition to skiplists we have tried trees. They are superior because they are slow in some circumstances but have lower asymptotic cost.

✓ In addition to skiplists we have tried trees. Skiplists are superior because, although slow in some circumstances, they have lower asymptotic cost.

Another problem with "it" is that it is overused.

✗ The machine crashed and it was necessary to reboot it.

✓ The machine crashed and had to be rebooted.

---

[1] A safe-sex guide issued by the Australian Government included "a table on which sexual practices are safe"; it transpired that this was not a piece of furniture.

Government guidelines on planning for emergencies had a list of "events that emergency recovery agencies have assisted", including "destruction of homes … toxic chemical spillage … holding of hostages".

Newspaper headlines can be a rich source of ambiguity, such as the following well-known examples:

> Enraged Cow Injures Farmer with Axe
>
> Miners Refuse to Work After Death

While not exactly ambiguous, the report that

> the pilot of a plane that crashed killing six people was flying "out of his depth"

does convey the wrong impression.

The first sentence is not ambiguous, but "it" has been used in two senses. Use a more specific term whenever doing so doesn't make the text too clumsy.

Premature pronouns also lead to difficulties.

✗ When it was first developed, recursive compilation was impractically slow and required too much memory.

✓ When recursive compilation was first developed, it was impractically slow and required too much memory.

A common source of confusion is between speed and time. Although not ambiguous, the phrase "increasing speed" is easily read as *increasing time*, which has the opposite meaning. There are similar problems with phrases such as "improving affordability".

A clumsy sentence is preferable to an ambiguous one. But remember that stilted sentences slow the reader, and it is difficult to entirely avoid ambiguity.[2]

Sentence Structure

Sentences should have simple structure, which usually means that they will be no more than a line or two. Don't say too much all at once.[3]

[2] The following my-dog-has-no-nose joke, due to Andy Clews, is not ambiguous.

First circumlocutionist: I have in my possession a male animal belonging to the family Canidae, and it appears that he does not possess any extra-facial olfactory organs.

Second circumlocutionist: Could you therefore impart to me such knowledge as is necessary to describe how that animal circumvents the problem of satisfying his olfactory senses?

First circumlocutionist: Unfortunately the non-ambiguity of your enquiry does not easily permit me to provide a clever answer, but I am in fact thinking of referring the animal to an olfactologist. However, the animal does have an unpleasant body odour, should you be interested.

[3] The following quote is a single sentence from a version of the standard lease agreement of the Real Estate Institute of Victoria, Australia. It is 477 words long, but the punctuation amounts to only three pairs of parentheses, one comma, and one stop. This clause is an example of "the fine print"—for example, the holder of a lease containing this clause has agreed not to take action if, in circumstances such as failure to pay rent, assaulted by the property's owner.

> If the Lessee shall commit a breach or fails to observe or perform any of the covenants contained or implied in the Lease and on his part to be observed and performed or fails to pay the rent reserved as provided herein (whether expressly demanded or not) or if the Lessee or other person or persons in whom for the time being the term hereby created shall be vested, shall be found guilty of any indictable offence or felony or shall commit any act of bankruptcy or become bankrupt or make any assignment for the benefit of his her or their creditors or enter into an agreement or make any arrangement with his her or their creditors for liquidation of his her or their debts by composition or otherwise or being a company if proceedings shall be taken to wind up the same either voluntarily or compulsorily under any Act or Acts relating to Companies (except for the purposes of reconstruction or amalgamation) then and in any of the said cases the Lessor notwithstanding the waiver by the Lessor of any previous breach or default by the Lessee or the failure of the Lessor to have taken advantage of any previous breach or default at any time thereafter (in addition to its other power) may forthwith re-enter either by himself or

✗ 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 roles is to deliver a message from a sending application process to a receiving 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.

That the kernel process is the default state is irrelevant here, and should have been explained elsewhere.

This example also illustrates the consequence of having too many words between related phrases. The original version said that "the time that is required for *something* is assumed to be …", where *something* was 34 words long. The main reason that the revision is clearer is that *something* has been reduced to two words, "this activity"; the structure of the sentence is much easier to see.

It is likewise helpful to avoid nested sentences, that is, information embedded within a sentence that is not part of its main statement.

✗ In the first stage, the backtracking tokenizer with a two-element retry buffer, errors, including illegal adjacencies as well as unrecognized tokens, are stored on an error stack for collation into a complete report.

First, this is poor because crucial words are missing. The beginning should read "In the first stage, which is the backtracking tokenizer". Second, the main information—how errors are handled—is intermixed with definitions. Nested content, particularly

(Footnote 3 continued)
by his agent upon the Premises or any part thereof in the name of the whole and the same have again repossess and enjoy as in their first and former estate and for that purpose may break open any inner or outer doorfastening or other obstruction to the Premises and forcibly eject and put out the Lessee or as permitted assigns any transferees and any other persons therefrom and any furniture property and other things found therein respectively without being liable for trespass assault or any other proceedings whatsoever for so doing but with liberty to plead the leave and licence which is hereby granted in bar of any such action or proceedings if any such be brought or otherwise and upon such re-entry this Lease and the said term shall absolutely determine but without prejudice to the right of action of the Lessor in respect of any antecedent breach of any of the Lessee's covenants herein contained provided that such right of re-entry for any breach of any covenant term agreement stipulation or condition herein contained or implied to which Section 146 of the Property Law Act 1958 extends shall not be exercisable unless and until the expiration of fourteen days after the Lessor has served on the Lessee the Notice required by Sub-section(1) of the said Section 146 specifying the particular breach complained of and if the breach is capable of remedy requiring the Lessee to remedy the breach and make reasonable compensation in money to the satisfaction of the Lessor for the breach.

if in parentheses, should be omitted. If such content really is required, then put it in a separate sentence.

✓ The first stage is the backtracking tokenizer with a two-element retry buffer. In this stage possible errors include illegal adjacencies as well as unrecognized tokens; when detected, errors are stored on a stack for collation into a complete report.

Watch out for fractured "if" expressions.

✗ If the machine is lightly loaded, then response time is acceptable whenever the data is on local disks.

✓ If the machine is lightly loaded and data is on local disks, then response time is acceptable.

✓ Response time is acceptable when the machine is lightly loaded and data is on local disks.

The first version is poor because the conditions of the "if" have been separated by the consequent.

It is easy to construct long, winding sentences by, for example, stating a principle, then qualifying it—a habit that is not necessarily bad, but does often lead to poor sentence structure—then explaining the qualification, the circumstances in which it applies, and in effect allowing the sentence to continue to another topic, such as the ideas underlying the principle, cases in which the qualification does or does not apply, or material which no longer belongs in the sentence at all, a property that is arguably true of most of this sentence, which should definitely be revised.

Sometimes longer sentences can be divided by, say, simply replacing an "and" or a semicolon with a full stop. If there is no particular reason to join two sentences, keep them separate.

Beware of misplaced modifiers.

✗ We collated the responses from the users, which were usually short, into the following table.

✓ The users' responses, most of which were short, were collated into the following table.

Double negatives can be difficult to parse and are ambiguous.

✗ There do not seem to be any reasons not to adopt the new approach.

The impression here is of condemnation—*we don't like the new approach but we're not sure why*—but praise was intended; the quote is from a paper advocating the new approach. This is another example of the academic tendency to overqualify. The revision "There is no reason not to adopt the new approach" is punchier, but still negative. It is difficult to suggest further improvement with the same meaning, because the meaning was probably unintended, but the following better reflects the original aims.

✓ The new approach is at least as good as the old and should be adopted.

Sometimes a double negative is a reasonable choice—and there are several such instances in this book. For example, the phrase "the two outcomes are not inconsistent" has the feel of a double negative, but it may be that the intention was to convey the sense that *the two outcomes do not contradict each other*, which is not the same as the negative-free *the two outcomes are consistent*. Whether this choice is the right one will depend on the surrounding text; for example, the word "consistent" may have a specific technical meaning, and thus shouldn't be used in its more general sense. But the fewer negatives you use, the clearer your writing will be.

Sing-song phrases can be distracting, as can rhymes and alliteration.

✗ We propose that the principal procedure of proof be use of primary predicates.

✗ Semantics and phonetics are combined by heuristics to give a mix that is new for computational linguistics.

Organize your sentences so that they can be parsed without too much backtracking. Ambiguous words or phraseology, even if clear in the context of a whole sentence, can slow the reader down.

✗ Classifying handles can involve opening the files they represent.

The opening phrase can, without the context provided by the rest of the sentence, be interpreted as *handles for classifying*.

✓ Classification of handles can involve opening the files they represent.

This is an instance of a more general problem. If an "-ing" suffix can be replaced by "-ation of", as in this example, then it is probably a good idea to do so. Other similar examples are as follows.

✗ The final line in the table shows that removing features with low amplitude can dramatically reduce costs.

✓ The final line in the table shows that removal of features with low amplitude can dramatically reduce costs.

✗ In this context, developing tools is not an option.

✓ In this context, development of tools is not an option.

In the first example, the reader may briefly wonder what a "removing feature" is—the construct feels like a name, not an action.

Know your limits. Experienced writers can construct complex sentences that are easy to read, but don't make the mistake of believing that something is easy to understand because you—the author—understand it.

Build your text from simple sentences and concise paragraphs. To guide analysis of your writing, ask elementary questions about it. Is each sentence motivated by the preceding text? Can you identify the sentence's purpose, that is, is it necessary? Could it be simplified? And so on. The habit of careful examination of your text can greatly improve your writing.

Tense

In science writing, most text is in past or present tense. Present tense is used for eternal truths. Thus we write "the algorithm has asymptotic cost $O(n)$", not "the algorithm had asymptotic cost $O(n)$". Present tense is also used for statements about the text itself. It is better to write "related issues are discussed below" than to write "related issues will be discussed below".

Past tense is used for describing work and outcomes. Thus we write "the ideas were tested by experiment", not "the ideas are tested by experiment". It follows that it is occasionally correct to use past and present tense together.

✓ Although theory suggests that the Klein algorithm has asymptotic asymptotic cost $O(n^2)$, in our experiments the trend observed was $O(n)$.

Either past or present tense can be used for discussion of references. Present tense is preferable but past tense can be forced by context.

✓ Willert (1999) shows that the space is open.

✓ Haast (1986) postulated that the space is bounded, but Willert (1999) has since shown that it is open.

Other than in conclusions, future tense is rarely used in science writing.

Repetition and Parallelism

Text that consists of the same form of sentence used again and again is monotonous. Watch out for sequences of sentences beginning with "however", "moreover", "therefore", "hence", "thus", "and", "but", "then", "so", "nevertheless", or "nonetheless". Likewise, don't overuse the pattern "First, … Second, … Last, …".

Complementary concepts should be explained as parallels, or the reader will have difficulty seeing how the concepts relate to each other.

✗ In SIMD, the same instructions are applied simultaneously to multiple data sets, whereas in MIMD different data sets are processed with different instructions.

✓ In SIMD, multiple data sets are processed simultaneously by the same instructions, whereas in MIMD multiple data sets are processed simultaneously by different instructions.

Parallels can be based on antonyms.

✗ Access is fast, but at the expense of slow update.

✓ Access is fast, but update is slow.

Lack of parallel structure can result in ambiguity.

✗ The performance gains are the result of tuning the low-level code used for data access and improved interface design.

✓ The performance gains are the result of tuning the low-level code used for data access and of improved interface design.

This can be further improved. It is kinder to the reader to move the longer clauses in a list to the end.

✓ The performance gains are the result of improved interface design and of tuning the low-level code used for data access.

There are some standard forms of parallel. The phrase "on the one hand" should have a matching "on the other hand". A sentence beginning "One …" suggests that a sentence beginning "Another …" is imminent. If you flag a point with "First" then every following point should have a similar flag, such as "Second", "Next", or "Last".

Parallel structures should be used in lists.

✗ For real-time response there should be sufficient memory, parallel disk arrays should be used, and fast processors.

The syntax can be fixed by adding "should be used" at the end but the result is clumsy. A complete revision is preferable.

✓ Real-time response requires sufficient memory, parallel disk arrays, and fast processors.

Note the use in this example of the serial comma, as discussed further in Chap. 8.

Comparisons and relative statements should be complete. If "the Entity-Relationship model is a better method for developing schema", then it is better than something else. Say what that something is.

Emphasis

The structure of a sentence places implicit emphasis, or stress, on some words. Reorganization of a sentence can change the emphasis.

✗ A static model is appropriate because each item is written once and read often.

It is not clear what makes the model's behaviour appropriate; the emphasis should be on the last two words, not the last five.

✓ A static model is appropriate because each item is only written once but is read often.

Inappropriate stress can lead to ambiguity.

✗ Additional memory can lead to faster response, but user surveys have indicated that it is not required.

✓ Faster response is possible with additional memory, but user surveys have indicated that it is not required.

The first version, which has the stress on "additional memory", incorrectly implies that users had commented on memory rather than response. Since the sentence is about "response", that is where the stress should be.

Explicit stress can be provided with italics, but is almost never necessary. Don't italicize words *unnecessarily*—let sentence structure provide the emphasis. Few papers require explicit stress more than once or twice. DON'T use capitals for emphasis. Some authors use the word "emphatic" to provide emphasis, as in "which are emphatically not equivalent". Other words used in this way are "certainly" and "indeed". The resulting wordiness weakens rather than strengthens; use of this form of emphasis should be rare.

Italicized passages of any length are hard to read. Rather than italicize a whole sentence, say, stress it in some other way: italicize one or two words only, or make it the opening sentence of a paragraph.

When a key word is used for the first time, consider placing it in italics.

✓ The data structure has two components, a *vocabulary* containing all of the distinct words and, for each word, a *hit list* of references.

Definitions

Terminology, variables, abbreviations, and acronyms should be defined or explained the first time they are used. Definitions should be specific and concrete. Don't create questions by giving definitions that refer to concepts that are unknown or uncertain.

Use a consistent format for introducing new terminology. Implicit or explicit emphasis on the first occurrence of a new word is often helpful, because it stresses what is being introduced.

✗ We use homogeneous sets to represent these events.

The reader has not been told that "homogeneous" is a new term that is about to be defined, and may look back for an explanation.

✓ We use *homogeneous* sets to represent these events.

✓ To represent these events we use homogeneous sets, whose members are all of the same type.

It can be helpful to give multiple explanations or illustrations of unfamiliar concepts.

✓ Compaction, in contrast to compression, does not preserve information; that is, compacted data cannot be exactly restored to the original form.

Sometimes a discursion—a discussion that is not part of the main thread of argument—is needed to motivate a definition. The discursion might consider negative examples, showing what happens in the absence of the definition, or it might lead the reader by steps to agree that the definition is necessary.

Choice of Words

Use short, direct words rather than long, circumlocutionary ones; the result is vigorous, emphatic writing. For example, use "begin" rather than "initiate", "first" and "second" rather than "firstly" and "secondly", "part" rather than "component", and "use" rather than "utilize". Use short words in preference to long, but use an exact long word rather than an approximate short one.

The words you choose should be specific and familiar. Abstract, vague, or broad terms have different meanings for different readers and can lead to confusion.

✗ The analysis derives information about software.

The "information" could be anything: optimizations, function-point descriptions, bug reports, or asymptotic cost.

✓ The analysis estimates the resource costs of software.

Other abstract terms that are overused include "important", "intelligent", "method", "paradigm", "performance", and "semantic". "Difficult" is often used when a better term is available: if something is "difficult to compute", does that mean that it is slow, or memory-hungry, or requires double precision, or something else altogether? "Hard" is sometimes used poorly too, including cases when "difficult" would be a better choice; remember that "hard" also means *inflexible* or *rigid* and can be misunderstood. "Efficient" is another word that is often vague. Use the most precise term available.

A common reason for using vague terms is that some authors feel they are writing badly if they use the same word twice in a sentence or paragraph, and thus substitute a synonym, which is usually less specific.

✗ The database executes on a remote machine to provide better security for the system and insulation from network difficulties.

✓ The database executes on a remote machine to provide better security for the database and insulation from network difficulties.

The "don't repeat words" rule might apply to creative writing, but not to technical terms that must be clearly understood.

Some sequences of words are awkward because they can be run together to form another, valid word.

✗ There are some times that appear inconsistent.

✓ Some of the times appear inconsistent.

This form is awkward for another reason—"some of the time" is a common phrase.

✓ Several of the times appear inconsistent.

Language is not static. Words enter the language, or go out of vogue, or change in meaning. A word whose meaning has changed—at least, some people still use the old meaning, but most use the new—is "data". Since "data" is by etymology a plural, expressions such as "the data is stored on disk" have been regarded as grammatically incorrect, but "the data are stored on disk" simply seems wrong. Correspondingly, "datum" is now rare. "Data" is appropriate for both singular and plural. On the other hand, use "automaton" rather than "automata" for the singular case.

Use a word only if you are sure that you know the meaning and can apply it correctly. Some words are familiar because of their use in a certain context—perhaps in a saying such as "hoist by his own petard" or a cliché such as "critical juncture"—but have otherwise lost their meaning. Other words, such as "notwithstanding", "whilst", and "amongst", have an archaic feel and can seem out of place in new writing. Some words have acquired meanings in computing that are distinct from their meaning elsewhere. Besides re-use of nouns such as "bus" and "record", there are more subtle cases. For example, "iterate" in computing means *to loop*, but in other writing it can mean *to do again*.

If you are unsure about a word, check it in a dictionary. There are many good online dictionaries, but be sure to check that the meaning is appropriate to computing or mathematics. Sometimes it can be helpful to use the Web to find examples of the word being used in context.

Some choices of word or phrase are cultural. For example, I've noticed that Indian writers sometimes write "different from" where a British writer would write "in contrast to", and moreover would argue that the Indian usage was wrong. With the globalization of English, however, it is often not logical to defend one usage over another. While I find "different from" irritating, and would never use it in my own writing, it is perfectly clear; the important thing is that your usage be consistent.

Slang should not be used in technical writing. Nor should the choice of words suggest that the writing is careless; avoid sloppy-looking abbreviations and contractions. Use "cannot" in preference to "can't", for example.

Don't make excessive claims about your own work. Phrases such as "our method is an ideal solution to these problems" or "our work is remarkable" are not acceptable. Claims about your own work should be unarguable.

Qualifiers

Don't pile qualifiers on top of one another. Within a sentence, use at most one qualifier such as "might", "may", "perhaps", "possibly", "likely", "likelihood", or "could". Overuse of qualifiers results in text that is lame and timid.

✗ It is perhaps possible that the algorithm might fail on unusual input.

✓ The algorithm might fail on unusual input.

✓ It is possible that the algorithm would fail on unusual input.

Here is another example, from the conclusions of a paper.

✗ We are planning to consider possible options for extending our results.

✓ We are considering how to extend our results.

Double negatives are a form of qualifier; they are commonly used to express uncertainty.

✗ Merten's algorithm is not dissimilar to ours.

Such statements tell the reader little.

Qualifiers such as "very" and "quite" should be avoided, because they are in effect meaningless. If an algorithm is "very fast", is an algorithm that is merely "fast" deficient in some way? Writing is invariably more forceful without "very".

✗ There is very little advantage to the networked approach.

✓ There is little advantage to the networked approach.

Likewise, "simply" can often be deleted.

✗ The standard method is simply too slow.

✓ The standard method is too slow.

Other words of this kind are "totally", "completely", "truly", "highly", "usually", "accordingly", "certainly", "necessarily", and "somewhat".

Misused Words

The Table 7.1 lists words that are often used incorrectly because of confusion with another word of similar form or sound. The "usually correct" form is shown on the left; the form with which each word gets confused is shown on the right. Some other problem words are as follows.

Which, that, the. Many writers use "which" when "that" is appropriate. Use "which" only when it cannot be replaced by "that".

Table 7.1 Misused words

Usual	Other
Alternative	Alternate
Coarse	Course
Comparable	Comparative
Complement	Compliment
Dependent	Dependant
Descendant	Descendent
Discrete	Discreet
Elusive	Illusive
Emit	Omit
Ensure	Insure
Ensure	Assure
Envelope	Envelop
Excerpt	Exert
Foregoing	Forgoing
Further	Farther
Insight	Incite
Lose	Loose
Omit	Emit
Partly	Partially
Practice	Practise
Principle	Principal
Simple	Simplistic
Solvable	Soluble
Stationary	Stationery

✗ There is one method which is acceptable.

✓ There is one method that is acceptable.

✓ There are three options, of which only one is tractable.

The word “that” is often underused. Use of “that” can make a sentence seem stilted, but its absence can make the sentence unclear.

✗ It is true the result is hard to generalize.

✓ It is true that the result is hard to generalize.

On the other hand, “the” is often used unnecessarily; delete it where doing so does not change the meaning.

Less, fewer. Use “less” for continuous quantities (“it used less space”) and “fewer” for discrete quantities (“there were fewer errors”).

Affect, effect. The “effect”, or *consequence*, of an action is to “affect”, or *influence*, outcomes.

Alternate, alternative, choice. The word “alternate” means *other* or *switch between*, whereas an “alternative” is something that can be chosen. Strictly speaking, if there is only one alternative, no choice is available; “alternative” and “choice” are not synonyms.

Assume, presume. “Assume” means *for now, take as being true*, while “presume” means *take for granted*. A fact is assumed as the basis of an argument, an event is presumed to have occurred.

May, might, can. Many writers use “may” or “might” when they mean “can”. Use “may” to indicate personal choice, and “can” to indicate capability.

✓ Users can access this facility, but may not wish to do so.

Basic, fundamental. Some writers confuse “basic” with “fundamental”: the former means *elementary* as well as *a foundation*. A result should only be described as “basic” if *elementary* is meant, or readers may get the wrong idea.

Novel, complex, sophisticated. “Sophisticated” does not mean *new* or *novel*, but either *advanced* or *complex*. Use “novel” or “complex” if these meanings are intended.

Will, shall. The word “shall” can seem archaic and is rarely preferable to “will”. Both “will” and “shall” are often used unnecessarily and in many cases can be deleted.

Compile, compose. In general usage, “compile” means *assemble*, *gather*, or *collect*, but it has such a strong specific meaning in computing that it should not be used for other purposes. To “compose” is to *invent* or perhaps *prepare*; it is not a synonym of “compile”, even though “composed of” means *made up of*.

Conflate, merge, confuse. The word “conflate” means *regard distinct things as similar*, while “merge” means *join distinct things to form one new thing*. If two things are “confused”, then one has been mistaken for the other. These three terms are not equivalent.

Continual, continuous. “Continual” is not equivalent to “continuous”. The former means *ceaselessly*; the latter means *unbroken*.

Conversely, inversely, similarly, likewise. Only use “conversely” if the statement that follows really is the opposite of the preceding material. Don’t use “similarly” or “likewise” unless whatever follows has a strong parallel to the preceding material. Some authors use “inversely”, but the meaning is rarely clear; avoid it.

Fast, quickly, presently, timely, currently. A process is “fast” if it *runs quickly*; “quickly” means *fast*, but does not necessarily mean *in the near future*. Something is “timely” if it is *opportune*; timeliness has nothing to do with rapidity. Also on the subject of time, “presently” means *soon*, whereas “currently” means *at present*.

Optimize, minimize, maximize. Absolute terms are often misused. One such word is "optimize", which means *find an optimum* or *find the best solution*, but is often used to mean *improve*. The latter usage is now so common that it could be argued that the meaning of "optimize" has changed, but as there is no synonym for "optimize" such a change would be unfortunate. Other absolute terms that are misused are "maximize" and "minimize".

Overlook, oversee, oversight. To "overlook" is to fail to notice, or to *ignore*. To "oversee" is to *manage* or look after. They are not synonyms! Even more confusingly, "oversight" means both of these things.[4]

Theory, hypothesis, proposition, supposition. These words are used in a wide variety of ways across the discipline. In some areas, "theory" is used in a strict sense, of a *hypothesis* that has been confirmed by analysis or experiment. But in some areas it is used more or less equivalently to "proposition", in the sense of *a concept that is to be tested*. Sometimes "proposition" is used to mean *assumption*, as is "supposition". That is, these terms are used both formally and loosely, in ways that can be deeply inconsistent with each other. As in other cases, be alert to the conventions within your discipline, but it is helpful to use these terms in ways that are consistent with their formal meaning, as they are part of the fundamental principles of science.

Spelling Conventions

A finished manuscript should as nearly as possible be free of spelling errors. As is also true for serious grammatical errors and poor formatting, the presence of spelling errors signals to the reader—perhaps subconsciously—that the work is unreliable and has been undertaken in a lazy way.

To ensure that all errors have been found it is essential to use a spell checker, but you should also take the effort to find mistakes by hand. It has been claimed that writers who depend solely on spell checkers tend to have more errors in their work than writers who don't use spell checkers at all, perhaps because the discipline of detailed examination means that the work is more carefully scrutinized overall.

Some words don't have a single fixed spelling. An example is "disk"; both this spelling and "disc" are so common that either is acceptable, but be consistent. However, "hard disk" is more common than "hard disc", and "compact disk" is incorrect. Other words that don't have a stable spelling include "enquire" ("inquire"), "biased" ("biassed"), and "dispatch" ("despatch"). In these examples, while one or the other spelling is more common in different cultures, both are in wide use.

---

[4] "Oversight" is an example of an auto-antonym, a word that means the opposite of itself. The common examples are not much used in computer science, but—to a wordaholic—they are fascinating. For example, "cleave" means both *divide* and *adhere*, and "sanction" means both *allow* and *penalize*. A "nice distinction" is probably *insightful* but might be *nit-picking*.

These are not the same as the words whose meaning has reversed over time. For example, "flammable" and "inflammable" now mean the same thing, where they once were opposites.

The English-speaking countries have different spelling conventions. For example, the American "traveler" becomes the British "traveller" while "fulfill" becomes "fulfil". In Britain it is incorrect to spell "-our" words as "-or", but, for example, "vigour" and "vigorous" are both correct.[5] The American "center" is the British "centre", "program" is "programme" (except for *computer program*), "catalog" and "analog" are "catalogue" and "analogue", "acknowledgment" is "acknowledgement", and "judgment" is "judgement". Another confusion is with regard to the suffixes "-ize" and "-yze", which have the same recommended spelling in both countries, but are often spelt as "-ise" and "-yse" outside the United States.[6] As discussed in Chap. 1, British spelling has largely been used throughout this book.

Science is international—technical writing is usually for a readership that is accustomed to reading text from around the world—and it is accepted that a national of one country won't necessarily use the spelling of another. The most important thing is to spell consistently and to be consistent with suffixes such as "-ize" without introducing errors such as "expertize" or "otherwize". Note that many journals insist on their own standards for spelling and presentation, or insist that the spelling be consistently of one nationality or another, and thus may choose to modify anything they publish.

The best authority for national spelling is a respectable dictionary written for that country. However, dictionaries are primarily a record of current non-technical spelling; the presence of a particular spelling in a dictionary does not prove that it is used in your discipline. The choice of spelling for a technical term may be dictated by the usage in other papers, not by your nationality.

Jargon

The word "jargon" means *terms used in a specialized vocabulary* or *mode of speech familiar only to a group or profession*.[7] As such, the use of jargon is an important part of scientific communication—how convenient it is to be able to say "CPU" rather than "the part of the computer that executes instructions". Some use of technical language, which inevitably makes the writing inaccessible to a wider readership, is essential for communication with specialists. But the more technical the language in a paper, the smaller the audience will be.

In mathematical writing, formal notation is a commonly used jargon. Mathematics is often unavoidable, but that doesn't mean that it must be impenetrable.

---

[5] An editor of the first edition of this book suggested that the material should have "an international flavor".

[6] However, these problems are overstated. For example, of the 6,000 or so distinct words used to write this book (modulo suffixes), other than "-ize" words only 20 or so have a nationality-specific spelling.

[7] From the Oxford Shorter Dictionary, which also lists *unintelligible or meaningless talk or writing; nonsense; gibberish; twittering*.

✗ **Theorem.** Let $\delta_1, \ldots, \delta_n$, $n > 2$ be such that $\delta_1 \mapsto_{\Omega_1} \delta_2$, …, $\delta_{n-1} \mapsto_{\Omega_{n-1}} \delta_n$. Let $\eta', \eta'' \in \mathscr{R}$ be such that $\Omega_1 \models \eta'$ and $\Omega_{n-1} \models \eta''$. Then

$$\exists (\eta', \eta_1)(\eta_1, \eta_2) \cdots (\eta_{r-1}, \eta_r)(\eta_r, \eta'') \in L$$

such that $\forall \eta_i$, $1 \le i \le r$, $\exists \Omega_j$, $1 \le j < n$, such that $\Omega_j \models \eta_i$.

Mathematics as jargon is discussed further in Chap. 9.

Jargon does not have to consist of obscure terms. It can be at its most confusing when words in common use are given a new meaning; and some words have multiple meanings even within computing.

✗ The transaction log is a record of changes to the database.

✓ The transaction log is a history of changes to the database.

The first version is confusing because databases consist of records. Likewise, consider "the program's function". Synonyms also cause such problems.

✗ Hughes describes an array of algorithms for list processing.

✓ Hughes describes several algorithms for list processing.

New jargon inevitably arises during research, as ideas are debated and simple labels are attached to new concepts. Consider whether your terminology conveys the intended meaning (or any meaning at all) to likely readers.

The need to name variants of existing ideas or systems presents a dilemma, because if the new name is dissimilar to the old then the relationship is not obvious, but prefixing a modifier to the old name—for example, to obtain "binary tree" from "tree"—can result in ridiculous constructs such as the "variable-length bitstring multiple-descriptor floating bucket extensible hashing scheme". If you need to qualify a name, choose a meaningful adjective. There are already too many "intelligent" algorithms, for example.

Where new terminology or jargon is introduced, use it consistently. Existing terminology or notation should only be changed with good reason. Sometimes your problem requires new terminology that is inconsistent with the terminology already being used, thus making change essential; but remember that any change is likely to make your paper harder to read.

Cliché and Idiom

Some expressions are clichés, that is, stock phrases whose meaning has little relationship to their words. Many readers, especially those from other cultures, may misunderstand such phrases. Examples include "follow suit", "up to scratch", "reinvent the wheel", "go through with a fine-tooth comb", "flat out", "cut and dried", and "bells and whistles". Idiomatic phrases are also poor choices in scientific writing, for similar reasons. Examples include "crop up", "lose track", "come to grips with", "it turned out that", "play up", "stacked deck", and "right out". Do not use such phrases.

Foreign Words

If you use a foreign word that you feel needs to go in italics, consider instead using an English equivalent. Some writers feel that use of foreign words is *de rigueur* because it lends the work a certain *je ne sais quoi* and shows *savoir-vivre*, but such writing is hard to understand.

Latin expressions are occasionally used—but more often misused—in technical writing. Examples include *mutatis mutandis*, *prima facie*, *circa*, *mea culpa*, and *vice versa*. Such phrases are not universally understood, and should only be used if you are confident of the meaning.

It is polite to use appropriate characters for foreign names, if they are natively written in a Latin character set. Don't write "Børstëdt" as "Borstedt", for example. But "张" may have to be written as "Zhang".

Overuse of Words

Repetition of a word is annoying when it makes the reader feel they have read the same phrase twice, or have read a phrase and an inversion of it.

✗ Ada was used for this project because the underlying operating system is implemented in Ada.

✓ Ada was used for this project because it is the language used for implementation of the underlying operating system.

Repetition should be eliminated when the same word is used in different senses, or when a word and a synonym of it are used together.

✗ Values are stored in a set of accumulators, each initially set to zero.

✓ Values are stored in a set of accumulators, each initialized to zero.

Some words grate when they are used too frequently. Common offenders include "this", "very", and "case". Other words are even more memorable—unusual words, other than technical terms, should only be used once or twice in a paper. Watch out for tics: excessive use of some stock word or phrase. Typical tics include "so", "also", "hence", "note that", and "thus".

There are cases in which repetition is useful. In the phrase "discrete quantities and continuous quantities", the first "quantities" can be omitted, but such omissions are often ambiguous and can result in text that is difficult to parse. What, for example, is intended by "from two to four hundred"? A common error relating to this form of expression is to shorten phrases by deleting adjectives, such as the second "long" in the expression "long lists and long arrays". Overuse of a word can lead to ambiguity,[8] but technical concepts should always be described in the same way, not by a series of synonyms.

Some phrases are worn out from overuse and, like clichés and the words listed earlier, should be avoided. Examples include "vicious circle", "as a matter of fact", "tip of the iceberg", "knotty problem", "in the final analysis", "every effort was made", and "vexed question".

Padding

Padding is the unnecessary use of pedantic phrases such as "in general", which should usually be deleted, not least because they are irritating. The phrase "of course" can be patronizing or even insulting—"*of course* it is now clear that that the order is stable". The phrases "note that" and "the fact that" are not padding, but are often used to introduce something that readers should be able to deduce for themselves.

Phrases involving the word "case" ("in any case", "it is perhaps the case") are also suspect. There is rarely a reason to use "it is frequently the case that" instead of "often", for example. Unnecessary introduction of quantities, or the concept of quantities, is a form of padding. For example, the phrase "a number of" can be replaced by "several", and "a large number of" by "many".

Adjectives are another form of padding.

✗ A well-known method such as the venerable quicksort is a potential practical alternative in instances of this kind.

In all likelihood, the context has made clear that impractical alternatives are not being discussed.

✓ A method such as quicksort is a potential alternative.

Use the minimum number of words, of minimum length, in your writing. The Table 7.2 lists common redundant or wordy expressions and possible substitutes for them. The list is illustrative rather than exhaustive; there are some typical forms of redundancy, such as "completely unique" for "unique", for which there are hundreds of examples. Sometimes a wordy expression is the right choice—to emphasise a key point, for example, or to lend the writing a conversational style—but in most cases a concise form is preferable.

---

[8] The following requirement was once in the Australian Tax Act.

> Where the amount of an annuity derived by the taxpayer during a year of income is more than, or less than, the amount payable for a whole year, the amount to be excluded from the amount so derived is the amount which bears to the amount which, but for this sub-section, would be the amount to be so excluded the same proportion as the amount so derived bears to the amount payable for a whole year.

Table 7.2 Examples of redundant or wordy expressions

Wordy	Concise
Adding together	Adding
After the end of	After
In the region of	Approximately
Cancel out	Cancel
Conflated together	Conflated
Let us now consider	Consider
Cooperate together	Cooperate
Currently … today	Currently …
Divided up	Divided
Give a description of	Describe
During the course of	During
Totally eliminated	Eliminated
Of fast speed	Fast
First of all	First
For the purpose of	For
Free up	Free
In view of the fact	Given
Joined up	Joined
Of large size	Large
Semantic meaning	Meaning
Merged together	Merged
The vast majority of	Most
It is frequently the case that	Often
Completely optimized	Optimized
Separate into partitions	Partition
At a fast rate	Quickly
Completely random	Random
Reason why	Reason
A number of	Several
Such as … etc.	Such as …
Completely unique	Unique
In the majority of cases	Usually
Whether or not	Whether
It is a fact that	—

Plurals

A common problem in English for writers educated in another language is agreement of plurals—a plural noun can require a differently formed verb to that required by a singular noun. For example, "a parser checks syntax" whereas "compilers check programs". Simple errors such as "the instructions is" are easy to identify, but care needs to be taken with complex sentence constructions. A particular problem is with collectives.

✗ The set of positive matches are then discarded.

✓ The set of positive matches is then discarded.

✗ The range of numbers that must be considered are easy to identify.

✓ The range of numbers that must be considered is easy to identify.

Consider proofreading your paper just to check for plural agreement.

When describing classes of things, excessive use of plurals can be confusing. The following is from a paper on minimum redundancy codes.

✗ Packets that contain an error are automatically corrected.

✗ Packets that contain errors are automatically corrected.

The first version implies that packets with a particular error are corrected, the second that packets with multiple errors are corrected. Both of these interpretations are wrong. Whenever it is reasonable to do so, convert plurals to singular.

✓ A packet that contains an error is automatically corrected.

Classes may not need a plural.

✗ These kinds of algorithms are irrelevant.

✓ These kinds of algorithm are irrelevant.

✓ Algorithms of this kind are irrelevant.

The use of variant plurals is becoming less common. Where once it was thought correct to base the plural form on that of the language of the root of the word, now it is almost always acceptable to use “-s” or “-es”. Thus “schemata” can be “schemas”, “formulae” can be “formulas”, and “indices” can be “indexes”; but, while “indices” is used in the context of arrays, it is almost never used in the context of databases. However, “radii” is not yet “radiuses”, and “matrices” is not yet “matrixes”. Special cases remain, in particular where the plural form has replaced the singular as in “data”, and in old-English forms such as “children”.

Abbreviations

It is often tempting to use abbreviations such as “no.”, “i.e.”, “e.g.”, “c.f.”, and “w.r.t.” These save a little space on the page, but slow readers down. It is almost always desirable to expand these abbreviations, to “number”, “that is”, “for example”, “compared with” (or more accurately “in contrast to”, since that is the sense in which “c.f.” should be used), and “with respect to”, or synonyms of these expressions. Where such abbreviations are used, the punctuation should be as if the expanded form were used. Also consider expanding abbreviations such as "Fig." and "Alg." (but note that the contracted form is the preferred style for some journals), and don't use concoctions such as "1st" or "2nd". Months should not be abbreviated. Make sure that all abbreviations and acronyms are explained when they are first used.

Avoid use of "etc." and "and so on". They are clumsy, and sometimes patronizing, as they can imply that the reader ought to be able to complete the list without the author's help.

✗ Methods available are random probing, extrapolation, etc.

✓ Methods available include random probing and extrapolation.

✓ Methods such as random probing and extrapolation can be used.

Never write "etc., etc." or "etc. …".

The ellipsis is a useful notation for indicating that text has been omitted. It should, therefore, only be used in quotations.

A slash, also known as a virgule or solidus, is often used for abbreviation, as in "save time and/or space" or "used for list/tree processing". Use of slashes betrays confusion, since it is often not clear whether the intended meaning is *or* (in the usual English sense of *either but not both*), *or* (in the usual computing sense of *either or both*), *and*, or *also*. If you want to be clear, don't use slashes.

An exception is "I/O", meaning *input and output*. There was once a variety of forms for this expression; now, all forms other than "I/O" are rare.

Acronyms

In technical documents with many compound terms it can be helpful to use acronyms, but as with abbreviations they can confuse the reader. An acronym is desirable if it replaces an otherwise indigestible name such as "pneumonoultramicroscopicsilico-volcanoconiosis" (miner's black lung disease), in which case the acronym becomes the name—as has happened with DNA for "deoxyribonucleic acid". Frequently used sequences of ordinary words, such as "central processing unit", are usually more convenient as acronyms; in a paper about a "dynamic multiprocessing operating system", it is probably best to introduce the DMOS right at the start. However, a surfeit of acronyms will force readers to flip back and forth through the paper to search for definitions. Don't introduce an acronym unless it is to be used frequently.

Acronyms can be fashionable. It was once common to write "WWW" to denote the World Wide Web, but today it is usually denoted by "the Web" or "the web"—often, it isn't even capitalized. And watch out for redundant acronyms, such as "the CPU unit". How, exactly, does a "local area LAN network" differ from a "LAN"?

Abbreviations end with a stop but it is unusual to put stops in acronyms. Thus "CPU" is correct, "C.P.U." is acceptable, and "CPU." is incorrect. Plurals of acronyms don't require an apostrophe; write "CPUs" rather than "CPU's".

Sexism

Forms of expression that unnecessarily specify gender are widely regarded as sexist. In technical writing, sexist usage is easy to avoid.

✗ A user may be disconnected when he makes a mistake.

✓ A user may be disconnected when they make a mistake.

✓ Users may be disconnected when they make a mistake.

The first use of “they”, as a singular pronoun, is acceptable but, to some readers, jarring. The second use, as a plural, removes sexism at the cost of clarity. It is preferable to recast the sentence.

✓ A user who makes a mistake may be disconnected.

Don’t use ugly constructs such as “s/he” or engage in reverse sexism by using “she” unless it is absolutely impossible to avoid a generic reference. Remember that some readers find use of “he” or “his” for a generic case offensive and dislike writing that employs such usage.

Chapter 8
Punctuation

Taste and common sense are more important than any rules; you put in stops to help your readers to understand you, not to please grammarians.

Ernest Gowers
The Complete Plain Words

Punctuation is a fundamental skill. Anyone reading this book is familiar with the functions of spaces, commas, stops, and capital letters. This chapter concerns stylistic issues of punctuation and errors that are common in science writing.

Fonts and Formatting

There is no obligation to use fancy typesetting just because a word processor provides it. Most computing or mathematical writing uses three fonts (plain, italic, and bold) or four (if, say, a fixed-width font is used for the text of programs) but use of more is likely to be annoying, and all but the plain font should be used sparingly. Overuse of fonts results in messy-looking text. Some authors prefer **bold** to *italic* for emphasis, but bold print is distracting. Use of underlining for emphasis, once common because of the limitations of typewriters as typesetting devices, is obsolete.

Use standard fonts for the text of papers. Two standard choices are Times-Roman and Cambria; note that these are too similar to be used in the same document. Fonts such as Helvetica and Courier are not as easy to read. Sans-serif fonts such as Calibri are widely used in advertising and to some readers don't seem sufficiently serious. An elaborate or unfamiliar font is almost always inappropriate.

Visual clutter of any kind is distracting and should be eliminated unless there is a clear need for it. Emphasis is one kind of clutter. Another is the use of graphic devices such as boxes around important points or icons next to results. Yet another kind of clutter is punctuation: excessive use of parentheses, quotes, italics, hyphens, semicolons, and uppercase letters.

Indentation is an important tool of layout, used primarily to indicate the start of a new paragraph. Some authors prefer to use a blank line instead, a decision that is

© Springer-Verlag London 2014

J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9_8

often unwise; the meaning is unclear at a page break, for example. Changes of topic should be signalled by headings. Indentation is also used to offset material that is not part of the textual flow, such as quotes, programs, and displayed mathematics.

Text looks tidier if it is right-justified as well as left-justified (although it is not always easier to read). Consider using a running header, of say the authors' surnames or the paper's title. Pages should be numbered. When submitting a paper, note that some journals and conferences ask that the author information be on a separate page, and many venues have specific formatting guidelines in their "Information for Authors".

Stops

Stops (or full-stops or periods) end sentences. Some writers don't use any other punctuation. Sentences should usually be short but commas and other marks give variety. Lack of other marks makes text telegrammatic. Such text can be tiring to read. Rather like this paragraph.

Stops are also used in abbreviations, acronyms, and ellipses. When these occur at the end of sentence, the sentence's stop is omitted. Problems with stops are a good reason to avoid abbreviations.

✗ The process required less than a second (except when the machine was heavily loaded, the network was saturated, etc.).

✓ The process required less than a second (unless, for example, the machine was heavily loaded or the network was saturated).

It is not usual to put a stop at the end of a heading.

✗ 3. Neural Nets for Image Classification.

✓ 3. Neural Nets for Image Classification

Commas

The primary uses of commas are to mark pauses, indicate the correct parsing, form lists, and indicate that a phrase is a parenthetical remark (that is, a comment) rather than a qualifier. Thus "the four processes that use the network are almost never idle" means *of the processes, the four that use the network are almost never idle*, while "the four processes, which use the network, are almost never idle" means *the four processes use the network and are almost never idle*. Incorrect use of commas in parenthetical remarks, in particular omission of the first of a pair of commas, is a frequent error.

✗ The process may be waiting for a signal, or even if processing input, may be delayed by network interrupts.

✓ The process may be waiting for a signal, or, even if processing input, may be delayed by network interrupts.

Use the minimum number of commas needed to avoid ambiguity. Sentences with many commas often have strangulated syntax; if the commas seem necessary, consider breaking the sentence into shorter ones or rewriting it altogether. But don't omit too many commas.

✗ When using disk tree algorithms were found to be particularly poor.

✓ When using disk, tree algorithms were found to be particularly poor.

✗ One node was allocated for each of the states, but of the nine seven were not used.

✓ One node was allocated for each of the states, but, of the nine, seven were not used.

✓ Nine nodes were allocated, one for each of the states, but seven were not used.

Another exception to the minimal-commas rule is in lists. A simple example of a list is “the structures were arrays, trees, and hash tables”. Many authors (and editors) prefer to omit the last comma from a list, known as the *serial comma*, a process that rarely adds clarity and often does it serious damage.

✗ At this stage, the alternatives were to branch to the left, back up one step and branch to the right, insert a new value or increment the failure counter and exit with error.

✓ At this stage, the alternatives were to branch to the left, back up one step and branch to the right, insert a new value, or increment the failure counter and exit with error.

Commas can be used to give the reader time to breathe.

✗ As illustrated by the techniques listed at the end of the section there are recent advances in parallel algorithms and multiprocessor hardware that indicate the possibility of optimal use of shared disk arrays by indexing algorithms such as those of interest here.

✓ As illustrated by the techniques listed at the end of the section, recent advances in parallel algorithms and multiprocessor hardware may allow optimal use of shared disk arrays by some algorithms, including indexing algorithms such as those of interest here.

Cutting this into several sentences would undoubtedly improve it further.

Colons and Semicolons

Colons are used to join related statements.

✓ These small additional structures allow a large saving: the worst case is reduced from $O(n)$ to $O(\log n)$.

Colons are also used to introduce lists.

✓ There are three phases: accumulation of distinct symbols, construction of the tree, and the compression itself.

The elements in a list can be separated by semicolons, allowing commas or other marks within each element.

✓ There are three phases: accumulation of distinct symbols in a hash table; construction of the tree, using a temporary array to hold the symbols for sorting; and the compression itself.

A semicolon can also be used to divide a long sentence, or to set off part of a sentence for emphasis.

✓ In theory the algorithm would be more efficient with an array; but in practice a tree is preferable.

Colons and semicolons are valuable but should not be overused.

Apostrophes

Many people seem to have trouble with apostrophes. Even professional writers get them wrong now and again. But the rules are quite simple.

- Singular possessives such as “the student’s algorithm”, “Brandt’s book”, and “Su and Ling’s method” require an apostrophe and an “s”. (Some people would write “Su’s and Ling’s method”, which is fine too.) For some names ending in “-s”, such as in “Williams’s book”, you can optionally omit the “s” after the apostrophe. If you are unsure, then the “s” should be given.
- Plural possessives such as “students’ passwords” require an apostrophe but no “s”.
- Pronoun possessives such as “its” (as in “its speed”) and “hers” do not require an apostrophe.
- Contractions such as “it’s” (as in “it is blue”) and “can’t” require an apostrophe; but note that contractions should be avoided in technical writing.[1]

Other than in the cases above, apostrophes are not required. The uses “in the 1980’s”, “each of the CPU’s”, “the computers’s power supplies”, and “Goss’ approach” are all incorrect.

[1] This book is a text, not a technical contribution, and I’ve used contractions without shame.

Exclamations

Avoid exclamation marks! Never use more than one!!

The proper place for an exclamation mark is after an exclamation (such as “Oh!”—not a common expression in technical writing), or, rarely, after a genuine surprise.

✓ Performance deteriorated after addition of resources!

This is acceptable but not particularly desirable. It would be better to omit the exclamation and add emphasis some other way.

✓ Remarkably, performance deteriorated after addition of resources.

Hyphenation

Many compound words, such as “website”, would originally have been written as two separate words, “web site”. When the combination becomes common, it is hyphenated, “web-site”, then eventually the hyphen is dropped to give the final form. Some words are in a state of transition from one form to another. In the database literature, for example, all three of “data base”, “data-base”, and “database” are used, and in general writing both “co-ordinate” and “coordinate” are common. Make sure that you are consistent.

Hyphens are also used to override right associativity. By default we parse phrases such as “randomized data structure” into *randomized data-structure*, and thus realize that the topic is not a structure for randomized data. In some phrases that are not right-associative, such as “skew-data hashing”, we need the hyphen to disambiguate (although in this case it might be better to write “hashing for skew data”).[2] Sometimes there is no correct hyphenation and the sentence has to be rewritten. The phrase “array based data structure” should be written “array-based data structure”, but “binary tree based data structure” should probably be written, albeit awkwardly, as “data structure based on binary trees”.

Good word-processors hyphenate words when they run over the end of a line, to preserve right-justification. Automatic hyphenation is not always correct and should be checked, to ensure that none of the syllables are broken or that the break is not too close to the word’s end. For example, the hyphenations “mac-hine” and “availab-le” should be changed (to “mach-ine” and “avail-able”), and “edited” should probably not be hyphenated at all.

Note that there are three different “dash” symbols: the hyphen “-” used for joining words, the minus sign or en-dash “–” used in arithmetic and for ranges such as “pages 101–127”, and the em-dash “—” used for punctuation.

[2] There is a hyphen missing in the headline “Squad helps dog bite victim”.

Capitalization

Capital letters were once used more liberally than they are now; in the eighteenth century writers commonly used capitalization (that is, an initial capital letter) to denote nouns. Today, only proper names are capitalized, and even these can be in lowercase if the name is in common use; for example, the capitals in the phrase "the Extensible Hashing method" should be in lowercase.

Some names are not consistently capitalized, particularly those of programming languages. Acronyms that cannot be sounded, such as "APL", should always be written that way, but the only general rule for other cases is to follow other authors. For example, both of the names "FORTRAN" and "Prolog" are abbreviations derived from truncated words. These are however proper names and should always have an initial capital; "lisp" and "pascal" are incorrect.

In technical writing it is usual to capitalize names such as "Theorem 3.1", "Figure 4", and "Section 11". In other writing, lowercase is preferred, but in technical writing lowercase looks sloppy to some readers.

Headings can be either minimally or maximally capitalized. In the former, words are capitalized as they would be in normal text, except that the word following a colon is capitalized.

✓ The use of jump statements: Advice for Prolog programmers

In the latter, words other than articles, conjunctions, or prepositions are capitalized; even these may be capitalized if they are over three letters long.

✓ The Use of Jump Statements: Advice for Prolog Programmers

The same rules apply to captions and titles of references.

Be consistent in your style of capitalization. It is acceptable to use maximum capitalization for sections and minimum capitalization for subsections, but not the other way around.

Quotations

One convention for quotations is that some punctuation marks are placed inside the quotation even when they are not part of the original material. An alternative is to place a punctuation mark within the quotation only if it was used in the original text—such as when a complete sentence is being quoted—as is done throughout this book.

✓ Crosley (2000) argues that "open sets are of insufficient power", but Davies (2002) disagrees: "If a concept is interesting, open sets can express it."

(But note that it is not essential to quote such a dull statement as “open sets are of insufficient power”; paraphrase, or even simply omitting the quote symbols, would be more appropriate. Omission of quotation marks in this case is acceptable—that is, not plagiarism—because Crosley’s statement is a natural way to express the concept.)

If the material in the quotation marks is a literal string, the punctuation must go outside. Since most punctuation symbols have meaning in programming languages, when programming statements are quoted the matter in the quote will be syntactically incorrect if the punctuation is in the wrong place.

✗ One of the reserved words in C is “for.”

✓ One of the reserved words in C is “for”.

Some editors change this to the wrong form. You may prefer to avoid the problem altogether.

✓ One of the reserved words in C is `for`.

Note that the angled or smart quotation symbols (“ and ”) are not the same as the straight ASCII double-quote symbol (").

Parentheses

A sentence containing a statement in parentheses should be punctuated exactly as if the parenthetical statement was not there.

✗ Most quantities are small (but there are exceptions.)
✓ Most quantities are small (but there are exceptions).
✗ (Note that outlying points have been omitted).
✓ (Note that outlying points have been omitted.)

Parenthetical remarks should be asides that the reader can ignore—important text should not be in parentheses. In particular, don’t introduce information in parentheses that is referred to later. The same rule applies to footnotes. If you think some text should be relegated to a footnote, perhaps it can be deleted.

Overuse of parentheses looks crowded. Avoid having more than one parenthesized remark per paragraph, or more than a couple per page. Parentheses within parentheses are hard to read and look like typing errors. Get rid of them.

The use of “(s)” to denote the possibility of a plural, as in “any observed error(s)”, is ugly and unnecessary; omit the parentheses or recast the sentence.

Citations

Citations should be punctuated as if they were parenthetical remarks.

✗ In [2] such cases are shown to be rare.

✗ In (Wilson 1984) such cases are shown to be rare.

Some journals typeset citation numbers as superscripts, in which case this example becomes "In[2] such cases are shown to be rare". Never treat a bracketed expression, whether a citation or otherwise, as a word.

✓ Such cases have been shown to be rare [2].

✓ Such cases have been shown to be rare (Wilson 1984).

✓ Wilson [2] has shown that such cases are rare.

✓ Wilson has shown that such cases are rare [2].

✓ Wilson (1984) has shown that such cases are rare.

The cite should be close to the material it relates to—poor placement of cites can be ambiguous.

✗ The original algorithm has asymptotic cost $O(n^2)$ but low memory usage, so it is not entirely superseded by Ahlberg's approach, which although of cost $O(n \log n)$ requires a large in-memory array (Ahlberg 1996; Keele 1989).

Since Ahlberg did not recognize the array as a problem and does not describe the old approach, this sentence is misleading.

✓ The original algorithm has asymptotic cost $O(n^2)$ but low memory usage (Keele 1989). Thus it is not entirely superseded by Ahlberg's approach (Ahlberg 1996), which, although of cost $O(n \log n)$, requires a large in-memory array.

The placement of citations depends partly on the citation style used. With the superscript style, for example, it is usual to try and place citations at the end of the sentence.

Chapter 9
Mathematics

Mathematics is no more than a symbolism. But it is the only symbolism invented by the human mind which steadfastly resists the constant attempts of the mind to shift and smudge the meaning …. Our confidence in any science is roughly proportional to the amount of mathematics it employs—that is, to its ability to formulate its concepts with enough precision to allow them to be handled mathematically.

Jacob Bronowski and Bruce Mazlish
The Western Intellectual Tradition

If you want to learn about nature, to appreciate nature, it is necessary to understand the language that she speaks in.

Richard P. Feynman
The Relation of Mathematics to Physics

Mathematics gives solidity to abstract concepts. As for writing in general, there are well-established conventions of presentation for mathematics and mathematical concepts. Reading mathematics is difficult work at the best of times, unpleasant work if the mathematics is badly presented, and pointless if the mathematics does not make sense.

Often, a statement can be made clearer by using mathematics to express the ideas. Mathematical notation can be used to describe algorithms, data structures, automata, or just about any of the objects that computer scientists study. The discipline of describing work in a mathematical form can expose inconsistencies and gaps, and provides a basis for making formal statements about the ideas being studied. While mathematics should not be used unnecessarily—to dress up uninteresting ideas, for example—ultimately a great deal of computer science has a mathematical foundation.

Clarity

In mathematical writing it is essential to be precise. For example, an ambiguous statement of a theorem can make its proof incomprehensible.

© Springer-Verlag London 2014

J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9_9

✗ An inverted list for a given term is a sequence of pairs, where the first element in each pair is a document identifier and the second is the frequency of the term in the document to which the identifier corresponds.

✓ An inverted list for a term t is a sequence of pairs of the form $\langle d, f \rangle$, where each d is a document identifier and f is the frequency of t in d.

In the first version, the author has had to struggle to avoid ambiguity.

Many terms have well-defined mathematical meanings and are confusing if used in another way.

Normal, usual, typical. The word "normal" has several mathematical meanings; it is often best to use, say, "usual" or "typical" if a non-mathematical meaning is intended.

Definite, strict, proper, all, some. Avoid "definite", "strict", and "proper" in their non-mathematical meanings, and be careful with "all" and "some".

Any. Avoid the word "any" in mathematical writing: sometimes it means "all" and sometimes it means "some".

Intractable, infeasible. An algorithm or problem is "intractable" only if it is NP-hard, that is, the asymptotic cost (or computational complexity) is believed to be worse than polynomial. In the context of asymptotic cost, "infeasible" sometimes has the same meaning as "intractable"; in the context of an optimization problem, it might mean that the problem has no (feasible) solution.

In general writing, either "infeasible" and "intractable" is sometimes used to mean *hard to do*, which is acceptable if there is no possibility of confusion.

Formula, expression, equation. A "formula", or an "expression" is not necessarily an "equation"; the latter involves an equality.

Equivalent, similar. Two things are "equivalent" if they are indistinguishable with regard to some criteria. If they are not indistinguishable, they are at best "similar".

Element, partition. An "element" is a member of a set (or list or array) and should not be used to refer to a subpart of an expression. If a set is "partitioned" into subsets, the subsets are disjoint and form the original set under union.

Average, mean. "Average" is used loosely to mean *typical*. Only use it in the formal sense—of *mean*, that is, the arithmetic mean—if it is clear to the reader that the formal sense is intended. Otherwise use "mean" or even "arithmetic mean".

Subset, proper subset, strict subset. "Subset" should not be used to mean *subproblem*. Orderings (or partial orderings) specified in writing are assumed to be non-strict. For example, "A is a subset of B" means that $A \subseteq B$; confusingly, this is sometimes written $A \subset B$. To specify $A \subsetneq B$ use "A is a proper (or strict) subset of B".

Similar rules apply to "less than", "greater than", and "monotonic".

Metric, measure. "Metric" is sometimes used informally to mean *measure*, but both have specific meanings in mathematics. In particular, when used in a formal context a metric is expected to satisfy conditions such as the triangle inequality. While "measure" also has a formal meaning, it is usually the less confusing of the two words, as it also has an appropriate informal usage. In mathematical contexts, use "measure" unless *metric* is intended.

Theorems

When you submit a paper containing a proof of a theorem, you should be satisfied that the proof is correct. However, the details of the proof may not be important to the reader and can often be omitted. Steps in the logic of a proof should be simple enough that the gaps can be completed by a reader mechanically, without too much invention. A common mistake is to unnecessarily include mechanical algebraic transformations; you need to work through these to check the proof, but the reader is unlikely to find them valuable.

Theorems, definitions, lemmas, and propositions should be numbered, even if there are only two or three of each in the paper, and you could consider numbering key examples. Not only does numbering allow reference within the paper, but it simplifies discussion of the paper later on. It is much easier for a correspondent to refer to "Definition 4.2" than "the definition towards the bottom of page 6". Many readers skim papers to find theorems (or other results such as illustrations or tables). For this reason, and because they may be quoted verbatim in other papers, statements of theorems should be as complete as possible.

In some theoretical contexts, authors choose to end a paper with a proof of some theorem or lemma. This style of writing can be unsatisfying for the reader. All papers can sensibly have an introduction and a conclusion, and it is worth reminding the reader of the main lessons of the paper in its final paragraphs.

Some presentation problems are not easily resolved. For a theorem with a complex proof, if the lemmas are proved early they appear irrelevant, and if they are proved late the main proof is harder to understand. One approach is to state the main theorem first, then state and prove the lemmas before giving the main proof, but in other cases all that can be done is to take extra care in the motivation and make liberal use of examples. Explain the structure of long proofs before getting to the detail, and explain how each part of the proof relates to the structure.

When stating your proof in a paper—that is, making it comprehensible to a reader—remember that you are presenting a reasoned argument. Use any available means to convey your argument with the greatest possible clarity; a diagram, for example, is perfectly acceptable. The end of each proof, example, or definition can be marked with a symbol such as a box. Alternatively, proofs and so on can be indented to set them apart from the running text.

Readability

Mathematics is usually presented in italics, to distinguish it from other text. For example, in the expression "of length *n*" it is clear that *n* is a variable of some kind. The main exception is function names such as log or sin, which are written in an upright font. Always use the same font for the same variable, and use the same font for all variables unless there is a good reason not to. There are several standards for representing a vector variable, such as $\mathbf{u}$, $\underset{\sim}{u}$, and $\vec{u}$. (The tilde was originally an instruction to the typesetter to choose a bold font.) Follow the conventions of your area rather than invent your own.

Brackets or square brackets [ ], parentheses ( ), and braces { } are used to delimit subexpressions, but braces can be confusing because they are also used to denote sets. To ensure that sets are distinguished from sequences, use braces for sets and parentheses for sequences. Angle brackets $\langle\ \rangle$ can also be used; these are not the same symbols as the relational operators $< >$. Note that some authors choose $\langle\ \rangle$ for the inner product of vectors: $\langle \vec{u}, \vec{v} \rangle = \sum_i u_i v_i$.

Use parentheses of appropriate size; they should stand out from the expressions they enclose.

✗ $(p \cdot (\sum_{i=0}^{n} A_i))$

✓ $\big(p \cdot (\sum_{i=0}^{n} A_i)\big)$

Sentences with embedded mathematics should be structured as if each formula is a simple phrase. Phrases indicate how the following text will be structured, but mathematics does not, and so should not be used at the start of a sentence.

✗ $p \leftarrow q_1 \wedge \cdots \wedge q_n$ is a conditional dependency.

✓ The dependency $p \leftarrow q_1 \wedge \cdots \wedge q_n$ is conditional.

The same rule applies to digits; sentences should begin with a word.

✗ 7 of the runs were successful, but 46 failed.

✗ Seven of the runs were successful, but 46 failed.

✓ There were 7 successful runs and 46 failures.

The first attempt at rephrasing fixes one problem but creates another, by introducing a mismatch in the way the numbers are described. In this case, a full rewrite is required.

Give the type of each variable every time it is used, so that the reader doesn't have to remember as many details and can concentrate on understanding the content. Watch out for misplaced types or variables.

✗ The values are represented as a list of numbers L.

✓ The values are represented as a list L of numbers.

✓ The values are represented as L, a list of numbers.

The former version is ambiguous—the symbol L might denote an individual member of the set.

Mathematics should not take the place of text: readers quickly get lost if they need to decipher a stream of complex expressions.

✗ Let $\langle S\rangle = \{\sum_{i=1}^{n} \alpha_i x_i \mid \alpha_i \in F, 1 \leq i \leq n\}$. For $x = \sum_{i=1}^{n} \alpha_i x_i$ and $y = \sum_{i=1}^{n} \beta_i x_i$, so that $x, y \in \langle S\rangle$, we have $\alpha x + \beta y = \alpha(\sum_{i=1}^{n} \alpha_i x_i) + \beta(\sum_{i=1}^{n} \beta_i x_i) = \sum_{i=1}^{n}(\alpha\alpha_i + \beta\beta_i)x_i \in \langle S\rangle$.

Although the mathematics in this example is straightforward, there is no motivation, and the thicket of symbols is daunting.

✓ Let $\langle S\rangle$ be a vector space defined by

$$\langle S\rangle = \left\{ \sum_{i=1}^{n} \alpha_i x_i \mid \alpha_i \in F \right\}.$$

We now show that $\langle S\rangle$ is closed under addition. Consider any two vectors $x, y \in \langle S\rangle$. Then $x = \sum_{i=1}^{n} \alpha_i x_i$ and $y = \sum_{i=1}^{n} \beta_i x_i$. For any constants $\alpha, \beta \in F$, we have

$$\begin{aligned} \alpha x + \beta y &= \alpha \left(\sum_{i=1}^{n} \alpha_i x_i \right) + \beta \left(\sum_{i=1}^{n} \beta_i x_i \right) \\ &= \sum_{i=1}^{n} (\alpha\alpha_i + \beta\beta_i) x_i \ , \end{aligned}$$

so that $\alpha x + \beta y \in \langle S\rangle$.

Note the vertical alignment of the equality symbols.

Mathematical expressions should not run together.

✗ For each x_i, $1 \leq i \leq n$, x_i is positive.

✓ Each x_i, where $1 \leq i \leq n$, is positive.

If a formula is complex, or is a key result, it should be displayed. In such displays, the formula can be either centred or indented; choose either, but be consistent. However, if part of the display is an algorithm or program, centering can look peculiar. Displayed formulas (or graphs or diagrams) should be positive results, not counter-examples, so that readers who skim through the paper won't be misled. If a displayed formula is sufficiently important it should be numbered, to allow discussion of it elsewhere in the paper and for reference once the paper is published. As in the example above, a displayed formula should be treated as a phrase; and remember to add trailing punctuation to the formula, with appropriate spacing.

✓ The weighting function can then be simplified to

$$w_\mu = (1 - \mu) \frac{f_x}{\Delta + l_x} + \mu \frac{f_y}{\Delta + l_y} \ ,$$

thus showing that μ controls the blending.

Mathematical symbols should, if possible, be the same font size as other characters. For example, the expression $(n(n+1)+1)/2$ is more legible than $\frac{n(n+1)+1}{2}$ even though the former uses more characters; but take care with expressions such as $a/b+c$ that are easy to misread. Font sizes should be consistent (though not necessarily identical) in text and displayed equations; large symbols are ugly and tiny symbols are illegible.

Consider breaking down expressions to make them more readable, especially if doing so enlarges small symbols.

$$\times \quad f(x) = e^{2^{-\frac{b}{a}x\sqrt{1-\frac{a^2}{x^2}}}}$$

$$\checkmark \quad f(x) = e^{2^{g(x)}} \quad \text{where} \quad g(x) = -\frac{b}{a}x\sqrt{1-\frac{a^2}{x^2}}$$

Avoid unnecessary subscripts: use x and y rather than x_1 and x_2. Also, don't nest subscripts on top of one another: the symbol i is legible in x_i, barely acceptable in x_{j_i}, and ridiculous in $x_{k_{j_i}}$. Mix subscripts and superscripts with caution: the expression $x_{j'}^{p_k}$ is a mess. Be careful with choice of letters for subscripts: in some small fonts, the letters i, j, and l are not easy to distinguish.

Some mathematics should be rewritten to remove or reduce the use of subscripts. For example, if $W = \{w_1, \ldots, w_k\}$ then you might write $\sum_{i=1}^{k} f_{w_i}$, but the equivalent expression $\sum_{w \in W} f_w$ is easier to read.

As illustrated in these examples, even simple mathematical expressions require competent typesetting. Such typesetting may involve use of advanced word-processing facilities, but failure to learn such facilities is no excuse for sloppy presentation.

Notation

Ensure that the symbols you use will be correctly understood by, and familiar to, the reader. For example, there are several symbols (including $\Rightarrow$, $\mapsto$, $\vdash$, $\supset$, $\sqsupset$, $\models$, and probably others) that are used in one context or another for logical implication. These symbols also have other meanings, so there is plenty of scope for confusion. The symbols $\sim$, $\simeq$, and $\approx$ are all used to mean *approximately equal to*, but $\sim$ may also represent an equivalence relation. The symbol $\cong$ means *is congruent to*, not *approximately equal*. Don't be lazy; use $\leq$, not $<=$, for *less than or equal to*.

The symbols for *floor* ($\lfloor\ \rfloor$) and *ceiling* ($\lceil\ \rceil$) seem to cause particular problems in typesetting. In one of my papers the typesetters changed these to square brackets ([ ]), and in the process utterly destroyed the meaning of the equations. Similarly, mistakes in placement are common with subscripts. Watch out for such errors.

Symbols such as $\forall$, $\exists$, $<$, $>$, $=$, and $\Rightarrow$, and abbreviations such as "iff", should not be used as substitutes for words. These symbols may be compact but they are difficult for readers to digest.

✗ Here, $\gcd(x, y) = \max\{z \mid \exists u, v : x = uz \wedge y = vz\}$

✓ Here, the value $\gcd(x, y)$ is the largest integer that is a divisor (or factor) of both x and y.

As this example also illustrates, clear writing rather than mathematics may be preferable for explanation of a familiar concept.

But don't replace symbols by words unnecessarily; for example, write "$a \leq b$" rather than "a is less than or equal to b". Concocted or amusing symbols are not a good idea; don't use $\clubsuit$ as an operator, for example. Use each operator for one purpose only; compilers may understand overloading, but people do not.

Don't re-use notation: an excellent way of confusing readers is to use N for one quantity on page 6 and for another on page 13. But expressions with similar meaning should have similar notation that follows consistent rules. Adhere to conventions such as using i and j for integer subscripts and calligraphic letters for classes. And don't vary an existing notation without good reason.

Take care with accents. Don't use $\hat{a}$, $\tilde{a}$, $\bar{a}$, and $\vec{a}$ together. Statistics texts sometimes use formulations such as $\bar{\hat{a}}$—the mean of the estimator of a—but this is better avoided by reformulating. And don't pile up primes: the symbol a'' may be clear, but what about $D_{i'}^{l'\prime}$? Some authors put powers on primes, as in a'^4 to represent a'''', but this notation is often unclear. If you have such deep primes, consider reworking your notation to get rid of them.

Ranges and Sequences

The closed range of real numbers r where $a \leq r \leq b$ is represented by "$[a, b]$"; the open range $a < r < b$ is represented by "(a, b)"; the range $a \leq r < b$ is represented by "$[a, b)$"; and the range $a < r \leq b$ is represented by "$(a, b]$".

It is common practice to use an ellipsis to describe a sequence of integers; thus $m, \ldots, n$ represents all integers between m and n inclusive. An infinite sequence is usually represented by $m_1, m_2, \ldots$, where it is assumed that the reader can extrapolate from the initial values to the other members of the sequence. Thus "2, 4, 8, ..." would be assumed to be the sequence of positive powers of 2. Always state both the lower and the upper bound if the sequence is finite and ensure that the intended sequence is clear.

An expression such as $1 \leq i \leq 6$ should be replaced by $i = 1, \ldots, 6$ if it is not clear that i should be an integer.

Some computer scientists, particularly in the context of computing theory, represent $1, 2, \ldots, n$ by $[n]$. Others let $[0, n]$ stand for $0, 1, 2, \ldots, n - 1$, to mimic the behaviour of programming languages such as C. Your usage should be made clear; unexplained notation, even if common in the specific field of the paper, is unnecessarily exclusive.

Alphabets

Characters from the Greek alphabet provide a clear way of denoting variables and mathematical quantities, as they cannot be confused with English text, and are widely used for this purpose.

Most readers are familiar with only a few Greek letters, so use of unfamiliar letters should be minimized, if only because new notation should be minimized. Many people find it easier to remember that a letter denotes a certain quantity if they already know the name of the letter; if they do not know the name they invent one, but this invention is generally not as effective a label as a real name. For example, reading the statement "sets are denoted by α" might result in the thought *sets are denoted by alpha* while reading "sets are denoted by ζ" (zeta) might result in the thought *sets are denoted by a squiggle-that-looks-like-a-deformed-s*. Other characters that have this effect are the Greek letters ξ and Ξ (xi), and symbols such as $\aleph$, $\Re$, and $\Im$.

Some mathematical symbols and characters from other alphabets have a superficial resemblance to more familiar symbols. Some pairs that can cause confusion, particularly after imperfect reproduction, shown in Table 9.1.

Line Breaks

Avoid letting a number, symbol, or abbreviation appear at the start of the line, particularly if it is the end of a sentence.

Table 9.1 Symbols that can be confused with each other

Symbol	Confused with
ε epsilon	e
ε epsilon	$\in$ (element of)
η eta	n
ι iota	i
μ mu	u
ρ rho	p
τ tau	t
υ upsilon	v
ν nu	v
ω omega	w
$\vee$ or	v
$\cup$ union	U
$\propto$ proportional	α alpha
$\top$ top (of lattice)	T
$\emptyset$ empty set	0 zero
$\times$ multiplication	x

✗ We therefore have to make use of a further class variable, denoted by x. It allows ...

✓ We therefore have to make use of a further class variable, denoted by x. It allows ...

A line that begins with a variable can look clumsy even if the variable is not at the end of a phrase.

✗ The remaining values are irreducible, in which case it is clear that set $\mathscr{D}$ is not empty.

Nor should you let a quantity become detached from its unit.

✗ Accesses to the new kind of file system typically require about 12 ms using our techniques.

✓ Accesses to the new kind of file system typically require about 12 ms using our techniques.

Most word processors provide an unbreakable-space character that prevents this behaviour. However, some word processors insist on breaking lines at awkward places in mathematical expressions.

✗ In this case the problem can be simplified by using the term $f(x_1, \ldots,$ $x_n)$ as a descriptor.

Sometimes the only solution is to rewrite the surrounding text.

✓ The problem is simplified if the term $f(x_1, \ldots, x_n)$ is used in this case as a descriptor.

Numbers

General writing guides recommend that large numbers should be written out in digits, such as 1,401 or 23, and that small numbers or round numbers should be spelt out: "one" rather than 1, and "a hundred" or "one hundred" rather than 100. In technical writing, however, digits are generally preferred when quantities are being reported, and in particular when numbers are being compared. First, digits are easier to locate when scanning a text, as a reader may do to remind themselves of numerical results reported earlier. Second, varying the presentation for figures that are meant to be compared introduces a false and perhaps misleading distinction between them.

However, words are sometimes preferable, for approximate numbers and for numbers at the start of a sentence, although, as noted earlier, it is generally better to recast the sentence so that the number is elsewhere. Percentages should always be in figures.

✗ 1024 computers were linked into the ring.

✗ Partial compilation gave a 4-fold improvement.

✗ The increase was over five per cent.

✗ The method requires 2 passes.

✓ There were 1024 computers linked into the ring.

✓ Partial compilation gave a four-fold improvement.

✓ The increase was over 5 per cent.

✓ The increase was over 5 %.

✓ Method 2 is illustrated in Fig. 1.

✓ The leftmost 2 in the sequence was changed to a 1.

✓ The method requires two passes.

Don't mix modes.

✗ There were between four and 32 processors in each machine.

✓ There were between 4 and 32 processors in each machine.

In English-speaking countries, the traditional method for separating long sequences of digits into groups of three is to use commas, as in "1,897,600". This method has two disadvantages: it can be confusing if the numbers are part of a comma-separated list, and decimal points are denoted by commas in many countries, so a number such as "1,375" could be misinterpreted. It is for these reasons that the alternative of using thin spaces was introduced, as in "1 897 600" or "73 802". But the comma-separated style remains popular and the use of thin spaces, although widespread in resources such as Wikipedia, has not become established in computer science papers. Comma separation is used throughout this book.

Fractions are only rarely used for values, and should not be used as abbreviations.

✗ About 1/3 of the data was noise.

✓ About one-third of the data was noise.

Numbers should not be adjacent.

✗ There were 14 512-Kb sets.

✓ There were fourteen 512-Kb sets.

Always include the leading 0 in numbers whose magnitude is less than 1; write "the size was 0.3 Kb", not "the size was .3 Kb".

Avoid the phrase "orders of magnitude".

✗ The new algorithm is at least two orders of magnitude faster.

In this example, is the unit of magnitude binary or decimal? It would be better to be explicit.

✓ The new algorithm is at least a hundred times faster.

Be clear about which base a number is in.[1]

Quantities that are in the same unit should, for consistency, be represented to the same precision. In physical experiments, it is usual to represent quantities to the same relative precision, that is, the same number of digits. In computer science, in which values are usually measured to the same absolute precision, it is more logical to represent quantities to the same number of decimal places.

✗ The sizes were 7.31 and 181 Kb, respectively.

✓ The sizes were 7.3 and 181.4 Kb, respectively.

A paper gave the same figure in different places as "almost 200,000", "about 170,000", "173,000", and "173,255"—an entirely unnecessary inconsistency.

Be realistic about accuracy and error. Your system may report that a process required 13.271844 CPU seconds, but in all likelihood the last four or five digits are meaningless. You should not imply accuracy by including spurious numbers. For example, "0.5 s" is not equivalent to "half a second", since the former implies that careful measurements were taken. Guesses and approximations should be clearly indicated as such, with words such as "roughly", "nearly", "approximately", "about", "almost", or "over"; but don't use wordy phrases such as "in the region of".

Percentages

Use percentages with caution.

✗ The error rate grew by 4 %.

This example is ambiguous because an error rate is presumably a percentage. It is better to be explicit, and to avoid mixing kinds of percentages.

✗ The error rate grew by 4 %, from 52 % to 54 %.

✓ The error rate grew by 2 %, from 52 % to 54 %.

When stating a percentage, ensure that the reader knows what is a percentage of what. If you write that "the capacity decreased by 30 %", is this 30 % of the old figure or the new? The convention is to use 100 % as the starting point, but in a series of statements of percentages it is easy to get lost. Use percentages rather than odds to express probabilities.

✗ The likelihood of failure is 2:1.

✓ The likelihood of failure is one in three.

✓ The likelihood of failure is about 30 %.

[1] It is said that there are 10 kinds of people in the world, those that understand binary and those that don't. (And how true it is. A reader suggested that "10" be changed to "ten".)

Don't use probabilities to describe small sets of observations. Success in two of five cases does not mean that the method "works 40 % of the time". The percentage gives the result an authority it might not deserve.

Units of Measurement

Two quantities are commonly measured in computer science: space and time. For time, the basic units are the second (s), minute (min) and hour (h); note that it is unusual to give the abbreviated forms of these units.[2] For the divisions of the second—the millisecond (ms or msec), microsecond (μs or μsec), and nanosecond (ns or nsec)—some readers may be unsure of the notation. For example, "ms" might be interpreted as *microsecond*. State such units in unabbreviated form at least once. When writing about hours or minutes use a colon rather than a stop to separate the components of the time. That is, write "3:30 min" rather than "3.30 min".

For space, the basic units are bit and byte. These are usually combined in tenth powers of 2 rather than third powers of 10,[3] although, confusingly, in IEEE standards it is powers of 10 that are correct (Table 9.2).

If there is any likelihood that, for example, a reader could interpret "Mb" as *megabit*, use "Mbyte" or "megabyte" instead. The larger units, especially "Pb", "Eb", "Zb", and "Yb", are unfamiliar to most readers and should be written in full at least once.

There are few derived units in computing other than the transfer rate of bytes per second, as in "18 Mb/sec". It was surprising to see "millibits" in a paper on arithmetic coding (in which symbols can be represented in a fraction of a bit). The unit is so unusual that "thousandths of a bit" is preferable.

Choose units that are easy to understand. For example, seconds can be preferable to minutes because fractions of a minute can be confusing: does "1.50 min" mean *one and a half minutes* or *one minute and fifty seconds*? (This problem can be avoided by using colons instead of stops, as discussed above.) Also, as values such as clock speeds and transfer rates are quoted in seconds, use of minutes makes comparison more difficult. On the other hand, "13:21 h" is perhaps kinder to the reader than "47.8×10^3 s".[4]

Some units, although in general use, are not well-defined. For example, MIPS (a million instructions per second) and gigaflops (a billion instructions per second) are increasingly meaningless; they cannot be used to compare machines of different

---

[2] Though variations of these units are common, with "sec" for seconds, for example, and "hr" for hours. While SI usage is to be encouraged, consistency is essential; choose an abbreviation and stick to it.

[3] Except, apparently, when the capacity of disk drives is being discussed. For example, a megabyte of disk is $10^6 = 1{,}000{,}000$ bytes, or somewhat less than a megabyte or $2^{20} = 1{,}048{,}576$ bytes of memory.

[4] As an engineer might write it. Or "4.78×10^4 s", as it might be written by a mathematician.

Table 9.2 Units of data volume

Unit	Value (bytes)	Denotation
Kilobyte	$2^{10} \approx 10^3$	Kb, Kbyte
Megabyte	$2^{20} \approx 10^6$	Mb, Mbyte
Gigabyte	$2^{30} \approx 10^9$	Gb, Gbyte
Terabyte	$2^{40} \approx 10^{12}$	Tb, Tbyte
Petabyte	$2^{50} \approx 10^{15}$	Pb, Pbyte
Exabyte	$2^{60} \approx 10^{18}$	Eb, Ebyte
Zettabyte	$2^{70} \approx 10^{21}$	Zb, Zbyte
Yottabyte	$2^{80} \approx 10^{24}$	Yb, Ybyte

architectures, in particular asynchronous processors without a central clock. Also, "gigaflops" strictly means a billion floating point operations per second, but is widely used to mean a billion instructions of any kind. Architecture-independent measures such as benchmarks may be more appropriate.

For quantities greater than 1, the unit is plural. For smaller quantities, the convention is that the unit is singular, but in computer science this convention is often not observed.

✓ The average run took 1.3 seconds, and the fastest took 0.8 second.

✓ The average run took 1.3 seconds, and the fastest took 0.8 seconds.

Units should be typeset in the font used in the paper for text, even when they are part of a mathematical expression.

✗ The volume is $r^p\ Kb$ in total.

✓ The volume is r^p Kb in total.

Put white space between values and units. Write "11.2 Kbytes" rather than "11.2Kbytes"; the second form may be common, but it is much harder to read. Numbers and their units should be hyphenated when used as an adjective.

✓ We also tried the method on the 2.7-Kb input.

Chapter 10
Algorithms

Mostly gobbledygook …
Eric Partridge, defining computation
Usage and Abusage

In many computer science papers, the core contribution is a family of algorithms. These algorithms are often the product of months of work; the version that the researchers have decided to submit for publication is typically based on a great deal of discussion, brainstorming, prototyping, testing, analysis, and debate over details. Yet in many cases this effort is not reflected in the presentation. Not only are the steps of the algorithm often not made clear, but there is no discussion of why the reader should believe that the algorithm is correct, or believe that its behaviour is reasonable. An algorithm by itself is uninteresting; what is of value is an algorithm that has been shown to solve a problem.

The topic of this chapter is effective description, analysis, and explanation of algorithms—perhaps the most challenging single task in writing of papers in computing. Experimental assessment of algorithms is considered in Chap. 14. Here the focus is on helping the reader to understand what the algorithm does, what effect it has, what its value is, and what its properties are.

Presentation of Algorithms

When an algorithm is presented in a computer science paper, the details of the algorithm by themselves—the program steps, for example—do not show that it is of value. You must demonstrate that the algorithm is a worthwhile contribution: for example, show that given appropriate input, it terminates with appropriate outcomes; or perhaps show, by a combination of proof and experiment, that it meets some claimed performance bound.

© Springer-Verlag London 2014

J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9_10

There are many reasons why you might choose to describe an algorithm. One is that it provides a new or better way to compute a result. What is usually meant by "better" is that, compared to previous approaches, the algorithm can compute the result with asymptotically fewer resources as demonstrated by mathematical analysis:[1] less time or memory, or some more desirable tradeoff of time and memory. It may be that the worst case is improved, at no saving in the average case; or that the average case is improved, but at the expense of space; or that all cases are improved, asymptotically, but with constant factors so large that there will no improvement in any conceivable practical situation. Each of these is a valid result, but it is crucial that the scope of the improvement be clearly specified—"better" is too vague.

As part of a description of an algorithm, a reader would expect to find of some or all of the following:

- The steps that make up the algorithm.
- The input and output, and the internal data structures used by the algorithm.
- The scope of application of the algorithm and its limitations.
- The properties that will allow demonstration of correctness, which might be formally expressed as pre- and post-conditions and loop invariants,
- A demonstration of correctness.
- A formal analysis of cost, for both space and time requirements.
- Experiments confirming the theoretical results.

Validation by experiment is often a critical part of the presentation in such cases. The experiment provides concrete evidence that, for some data, the algorithm terminates correctly and performs as predicted. (Experiments are discussed in Chap. 14.) However, while experiments on an algorithm may support an asymptotic analysis, they cannot replace it.

Another reason for describing an algorithm is to explain a complex process. For example, a paper about a distributed architecture might include a description of the steps used to communicate a packet from one processor to another. These steps certainly constitute an algorithm, and, while readers would not expect an asymptotic cost analysis, you would have to give an argument to show that the steps did indeed result in packet transmission. Similarly, when describing a parser, say, there is no blanket requirement that a mathematical analysis must be given (different norms apply to different areas and readerships), but that does not excuse you from giving an analysis where it is appropriate to do so.

Yet another reason for describing an algorithm is to show that it is feasible to compute a result, regardless of the cost, or to show that a problem is decidable. Again, different norms apply. In such cases a formal proof of correctness is essential, while an asymptotic analysis may be of little interest.

[1] Terminology in the area of analysis is both precise and inconsistent. A common usage is to refer to a *complexity analysis*, and to describe algorithms as having a certain *complexity*. Some people regard such usage as sloppy, and instead use the terminology *asymptotic cost*, as determined by a *cost analysis*, a terminology I have broadly kept to here. However, other people also object to this usage, and argue for yet other wordings. It may be that my usage is not quite in keeping with that of some mathematical computer scientists, but it is at least reasonably unambiguous.

In summary, in the presentation of an algorithm it is usual to give a formal demonstration of correctness and performance, and perhaps an experimental validation. When such demonstrations are absent, the reason for the absence should be clear.

Formalisms

The description of an algorithm usually consists of the algorithm itself and the environment it requires. There are several common formalisms for presenting algorithms. One is the *list* style, in which the algorithm is broken down into a series of numbered or named steps and loops involving several steps are represented by "go to step X" statements. This form has the advantage that the algorithm can be discussed as it is presented: there is no restriction on the amount of text used to describe a step (although a step should be a single activity), so there is room for a clear statement of each step and for remarks on its properties. But the control structure is often obscure and it is all too easy for the discussion to bury the algorithm.

Another common formalism is *pseudocode*, in which the algorithm is presented as if written in a block-structured language and each line is numbered. An example is shown in Fig. 10.1. Pseudocode has the advantage that the structure of the algorithm is immediately obvious; but each statement is forced by formatting considerations to be fairly terse, and it is not easy to include detailed comments. Also, the use of programming language constructs and notation is usually a mistake. It takes experience to present algorithms well in pseudocode, and, although it is straightforward to translate such pseudocode into an imperative programming language, pseudocode is unnecessarily difficult to understand.

A better option is to use what might be called *prosecode*, in which the algorithm is described by text with embedded code, rather than as code with textual annotations; structure is given by numbered lists in which loops are presented as sublists with nested numbering. An example is shown in Fig. 10.2. In the example, input and output are described in the preamble, and "code" statements and explanatory text are mixed freely in the algorithm itself. Despite the informality, the specification of the algorithm is direct and clear. The assignment symbol "$\leftarrow$" is a good choice because it is unambiguous, in contrast to "$=$". However, the prosecode style of presentation is only effective when the concepts underlying the algorithm have been discussed before the algorithm is given.

Another effective approach to description of algorithms is what might be called *literate code*, in which the detail of the algorithm is introduced gradually, intermingled with discussion of the underlying ideas and perhaps with the asymptotic analysis and proof of correctness. An example is shown in Fig. 10.3. (This example is incomplete—most algorithms worth presenting need a substantial explanation that can't be condensed into a page or two.)

The **WeightedEdit** function computes the edit distance between two strings, assigning a higher penalty for errors closer to the front.

Input:	$S1, S2$: strings to be compared.
Output:	weighted edit distance
Variables:	$L1, L2$: string lengths
	$F[L1, L2]$: array of minimum distances
	W: current weighting
	M: maximum penalty
	C: current penalty

```
WeightedEdit(S1,S2):
1.   L1 = len(S1)
2.   L2 = len(S2)
3.   M = 2 × (L1 + L2)
4.   F[0,0] = 0
5.   for i from 1 to L1
6.        F[i,0] = F[i − 1,0] + M − i
7.   for j from 1 to L2
8.        F[0, j] = F[0, j − 1] + M − j
9.   for i from 1 to L1
10.       C = M − i
11.       for j from 1 to L2
12.            C = C − 1
13.            F[i, j] = min(F[i − 1, j] + C,
                             F[i, j − 1] + C,
                             F[i − 1, j − 1] + C × isdiff(S1[i], S2[j]))
14.  WeightedEdit = F[L1, L2]
```

Fig. 10.1 Example of pseudocode. This is not the best style of presentation: the algorithm is cryptic and the numbering does not reflect the indentation. Also, the author has unnecessarily introduced a trivial optimization (at lines 10 and 12) and the notation for variables is ugly. It is like a program meant for a machine, not an explanation meant for a reader

Regardless of the presentation form chosen, you need to consider the extent to which the algorithm, or its components, can be presented as mathematical abstractions. Can a loop be described as an operation on a set? Does the order in which array elements are processed matter? To understand pseudocode, a reader must reinterpret the sequence of statements as a higher-level abstraction; the algorithm should be presented at such a level.

WeightedEdit(s,t) compares two strings s and t, of lengths $k(s)$ and $k(t)$ respectively, to determine their edit distance—the minimum cost in insertions, deletions, and replacements required to convert one into the other. These costs are weighted so that errors near the start of the strings attract a higher penalty than errors near the end.

We denote the ith character of string s by s_i. The principal internal data structure is a 2-dimensional array F in which the dimensions have ranges 0 to $k(s)$ and 0 to $k(t)$, respectively. When the array is filled, $F_{i,j}$ is the minimum edit distance between the strings $s_1 \cdots s_i$ and $t_1 \cdots t_j$; and $F_{k(s),k(t)}$ is the minimum edit distance between s and t.

The value p is the maximum penalty, set to $2(k(s)+k(t))$, and the penalty for a discrepancy between positions i and j of s and t, respectively, is $p-i-j$, so that the minimum penalty is $p-k(s)-k(t) = p/2$ and the next-smallest penalty is $p/2+1$. Two errors, wherever they occur, will outweigh one.

1. (Set penalty.) Set $p \leftarrow 2 \times (k(s)+k(t))$.
2. (Initialize data structure.) The boundaries of array F are initialized with the penalty for deletions at start of string; for example, $F_{i,0}$ is the penalty for deleting i characters from the start of s.
   (a) Set $F_{0,0} \leftarrow 0$.
   (b) For each position i in s, set $F_{i,0} \leftarrow F_{i-1,0} + p - i$.
   (c) For each position j in t, set $F_{0,j} \leftarrow F_{0,j-1} + p - j$.
3. (Compute edit distance.) For each position i in s and position j in t:
   (a) The penalty is $C = p - i - j$.
   (b) The cost of inserting a character into t (equivalently, deleting from s) is $I = F_{i-1,j} + C$.
   (c) The cost of deleting a character from t is $D = F_{i,j-1} + C$.
   (d) If s_i is identical to t_j, the replacement cost is $R = F_{i-1,j-1}$. Otherwise, the replacement cost is $R = F_{i-1,j-1} + C$.
   (e) Set $F_{i,j} \leftarrow \min(I,D,R)$.
4. (Return.) Return $F_{k(s),k(t)}$.

Fig. 10.2 Example of prosecode. The longer introduction and use of text in the presentation help make the algorithm easy to understand

WeightedEdit(s,t) compares two strings s and t, of lengths $k(s)$ and $k(t)$ respectively, to determine their edit distance—the minimum cost in insertions, deletions, and replacements required to convert one into the other. These costs are weighted so that errors near the start of the strings attract a higher penalty than errors near the end.

The major steps of the algorithm are as follows.
1. Set the penalty.
2. Initialize the data structure.
3. Compute the edit distance.

We now examine these steps in detail.

1. Set the penalty.

   The main property that we require of the penalty scheme is that costs reduce smoothly from start to end of string. As we will see, the algorithm proceeds by comparing each position i in s to each position j in t. Thus a diminishing penalty can be computed with the expression $p-i-j$, where p is the maximum penalty. By setting the penalty thus

   (a) Set $p \leftarrow 2 \times (k(s)+k(t))$

   the minimum penalty is $p-k(s)-k(t) = p/2$ and the next-smallest penalty is $p/2+1$. This means that two errors—regardless of position in the strings—will outweigh one.

2. Initialize data structures ...

Fig. 10.3 Example of literate code. The algorithm is explained and presented simultaneously. This is the most verbose style, but, usually, the clearest. Note that this example is incomplete

Level of Detail

Algorithms should be specified in sufficient detail to allow them to be implemented without undue inventiveness.

× 5. (Matching.) For each pair of strings $s, t \in S$, find $N_{s,t}$, the maximum number of non-overlapping substrings that s and t have in common.

The way in which a step of this kind is implemented may greatly affect the behaviour of the final algorithm, so the matching process needs to be made explicit.

But don't provide too much detail. For example, loops are sometimes used unnecessarily in specification of algorithms.

✗ 3. (Summation.) *sum* $\leftarrow 0$. For each j, where $1 \le j \le n$,
  (a) $c \leftarrow 1$; the variable c is a temporary accumulator.
  (b) For each k, where $1 \le k \le m$, set $c \leftarrow c \times A_{jk}$.
  (c) *sum* $\leftarrow$ *sum* $+ c$.

This is cumbersome and no more informative than the equivalent mathematical expression. It is safe to assume that most programmers know how to use loops to implement sums and products.

✓ 3. (Summation.) Set *sum* $\leftarrow \sum_{j=1}^{n} \left(\prod_{k=1}^{n} A_{jk} \right)$.

Written this way, it is unclear that a separate variable *sum* is even required: the mathematical expression may suffice in future references to the same value. Giving this step explicitly, however, might help if, say, the matrix A was sparse and stored as a list rather than as a two-dimensional array, thus requiring an explanation of how to compute the summation efficiently.

In specifications of algorithms, use text rather than mathematics if the former is sufficiently clear.

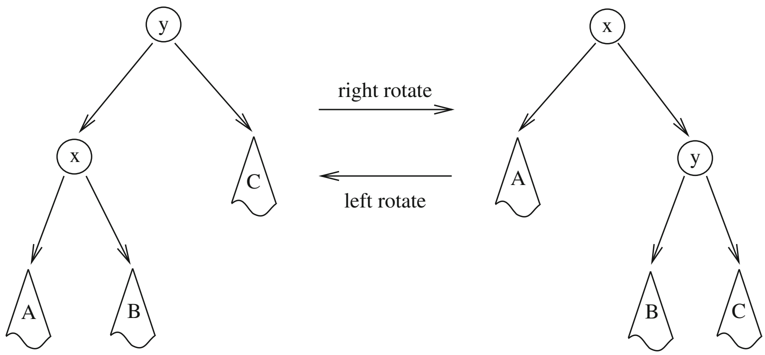
✗ 2. for $1 \le i \le |s|$
  (a) set $c \leftarrow s[i]$
  (b) set $A_c \leftarrow A_c + 1$

✓ 2. For each character c in string s, increment A_c.

Figures

Figures are an effective way of conveying the intricacies of data structures; and even quite simple structures can require complex descriptions. General guidelines for figures are given in Chap. 11.

✓ A single rotation can be used to bring a node one level closer to the root. In a left-rotation, a parent node x and its right child y are exchanged as follows: as B is the left child of y, assign B to be the new right child of x and assign x to be the new left child of y. The left child of x and the right child of y remain unchanged. The complementary operation is a right-rotation. Left- and right-rotations are shown in the following diagram.



Notation

Mathematical notation is preferable to programming notation for presentation of algorithms. Use "x_i" rather than "`x[i]`", for example. Don't use "`*`" or "`x`" to denote multiplication; most word processors provide a multiplication symbol such as "$\times$" or "$\cdot$", and in any case multiplication is often implicit.

Likewise, avoid using constructs from specific programming languages. For example, expressions such as `==`, `a=b=0`, `a++`, and `for (i=0; i<n; i++)` may have little meaning, or even the wrong meaning, to readers who are unfamiliar with C. Block-bounding statements such as `begin` and `end` are usually unnecessary; nesting can be shown by indentation or by the numbering style, as in the examples in Figs. 10.1 and 10.2.

Mathematics provides numerous handy conventions and symbols that can be used in description of algorithms, including set notation, subscripts and superscripts, and symbols such as $\lceil$ and $\rceil$, $\sum$, and $\prod$. But remember that such notation has a widely understood formal meaning that should not be abused. Also, good programming style does not necessarily imply good style for description of algorithms. For example, take care with variable names of more than one character—don't use "pq" if it might be interpreted as "$p \times q$".

It was once common to include the text of a program in a paper, in addition to a description of the algorithm it embodies. Inclusion of the code was valuable because, for short programs at least, it was the simplest way for readers to obtain the code. However, this practice is now extremely rare.

Environment of Algorithms

The steps that comprise an algorithm are only part of its description. The other part is its environment: the data structures on which it operates, input and output data types, and, in some cases, factors such as properties of the underlying operating system and hardware. If the environment of an algorithm is not described, the algorithm is likely to be difficult to understand. For example, a presentation of a list-processing algorithm should include descriptions of the list type, the input, and the possible outputs. If the list is stored on secondary storage and speed is being analyzed, it might also be appropriate to describe assumed disk characteristics. For algorithms in which there are hardware considerations, such as memory size or disk throughput, for the environment to seem realistic any assumptions about the hardware should reflect current technology or likely improvements in the near future.

Specify the types of all variables, other than trivial items such as counters; describe expected input and output, including assumptions about the correctness of the input; state any limitations of the algorithm; and discuss possible errors that are not explicitly captured by the algorithm. Most importantly, say what the algorithm does.

Describe data structures carefully. This does not mean that you should give record definitions in a pseudo-language; instead, use, say, a simple mathematical notation to unambiguously specify the structure.

✓ Each element is a triple

```
(string, length, positions)
```

in which `positions` is a sequence of byte offsets at which `string` has been observed.

Be consistent. When presenting several algorithms for the same task, they should as far as possible be defined over the same input and output. It may be the case that some of the algorithms are more powerful than the others—perhaps they can process a richer input language, for example. Variations of this kind should be made explicit.

Asymptotic Cost

The performance of algorithms is often measured by asymptotic analysis; the reader should learn how an algorithm behaves as the scale of the problem changes. Big-O notation can be defined as follows: a function $f(n)$ is said to be $\mathrm{O}(g(n))$—that is, $g(n)$ is an upper bound of $f(n)$—if for some constants c and k we have $f(n) \leq c \cdot g(n)$ for all $n > k$. If $f(n)$ is $\mathrm{O}(g(n))$ and $g(n)$ is $\mathrm{O}(f(n))$, then $f(n)$ is $\Theta(g(n))$; that is, Θ is used to define functions that grow asymptotically at the same rate.

A function $f(n)$ is $\mathrm{o}(g(n))$ if f is $\mathrm{O}(g(n))$ but not $\Theta(g(n))$. Likewise, Ω and ω are used to describe lower bounds. Other definitions are given by some authors, and the use of the notation is slightly inconsistent, so it is helpful to define what you mean by, for example, $\Omega(g(n))$. For a precise discussion, consult an algorithms text.

Big-O notation is also used in another, less formal sense, to mean *the asymptotic cost* rather than *an upper bound on the asymptotic cost*. An author might write that "comparison-based sorting takes $O(n \log n)$ time" or that "linear insertion sort always takes at least $O(n)$ time"; which, although an abuse, is perfectly clear and has stronger emphasis than "linear insertion sort has asymptotic cost $\Omega(n)$". But beware of loose usage that could be misunderstood. When you describe an algorithm as "quadratic", some readers may assume that quadratic worst case is meant, while others may assume that it is quadratic in all cases. Similarly be careful with "constant", "linear", "logarithmic", and "exponential".

A related cause of occasional confusion is the distinction between the intrinsic asymptotic cost of a problem and the properties of a specific algorithm for the problem. For some problems, a theoretical limit is known, which is usually a lower bound on the asymptotic cost of any algorithm that solves the problem. Specific algorithms may not approach this lower bound. Unhelpfully—and incorrectly—authors have been known to write statements of the form "the problem's complexity is $O(f(n))$", implicitly meaning intrinsic cost, even though $O(\cdot)$ is an upper bound and intrinsic cost is (usually) a lower bound.

For algorithms that operate on static data structures, it may be appropriate to consider the cost of creating that data structure. For example, binary search in a sorted array takes $O(\log n)$ time, but $O(n \log n)$ time could be required to initially sort the array.

Make sure that the domain of the analysis is clear, and be careful to analyze the right component of the data. It would usually be appropriate, for example, to analyze database algorithms as a function of the number of records, not of the length of individual records. However, if record length can substantially vary then possibly it too should be considered. For algorithms that apply arithmetic to integers it may be appropriate to regard each arithmetic operation as having unit cost. On the other hand, if the integers involved can be of arbitrary length (consider for example public-key encryption algorithms that rely for privacy on the expense of prime factorization) it is appropriate to regard the cost of the arithmetic operations as a function of the number of bits in each integer.[2]

Two subtle problems are that the dominant cost may change with scale, and that the cost that is dominant in theory may never dominate in practice. For example, a certain algorithm might require $O(n \log n)$ comparisons and $O(n)$ disk accesses. In principle the asymptotic cost of the algorithm is $O(n \log n)$, but, given that a disk access may require 5 ms and a comparison less than a nanosecond, in practice the cost of the disk accesses might well dominate for any conceivable application.

Some authors misunderstand the logic of asymptotic claims. For example, Amdahl's law states that the lower bound for the time taken for an algorithm to complete is determined by the part of the algorithm that is inherently sequential. The remainder can be executed in parallel and hence time for this part can be reduced by addition of processors, but no increase in the number of processors can affect the lower bound. However, it has been claimed that Amdahl's law was broken, in the context of a certain algorithm, by increasing both the size of the input data and the number of processors. These changes had minimal impact on the sequential part of the algorithm, so that the proportion of total processing time spent in the sequential part was reduced; but this result does not contradict Amdahl's law, and so the claim was false.

Another fallacious claim was that, for a certain indexing technique, the time required to find matches to a pattern in a database was asymptotically sublinear in the database size—a remarkable result, because the probability that a record is a match to a given pattern is fixed, so that in the limit the number of matches must be linear in database size. The error was that the author had assumed that the length of the pattern was a logarithmic function of database size, so that the number of answers

---

[2] The asymptotic cost of search in a binary tree of size n is usually given as $O(\log n)$, but that assumes that the cost of comparing two keys is constant. If the keys are strings, the expected cost of a pairwise comparison is $O(\log n)$, because, as the search narrows, the number of bits to be inspected in each string grows; and thus the overall cost might be $O(\log^2 n)$. On current machines, however, the string comparison cost is unlikely to be significant—presenting the question of what exactly should be considered in the analysis. Perhaps cache fetches rather than machine instructions are the operation of interest.

was constant. The technique gave the appearance of being sublinear because the task was changing.

Sometimes a formal analysis is inappropriate or only a minor consideration. For example, an algorithm for arranging line breaks in paragraphs of text will only rarely have to operate on a large input, so showing that a new algorithm is better than an existing algorithm in the limit may be of less interest than showing it is better on a typical case. More generally, although some results can be conclusively obtained by analysis, others cannot. Analytical results often say nothing about constant factors, for example, or behaviour in practice where CPU, cache, bus, and disk can interact in complex ways. Such properties can only be determined by experiment. Thus, while an asymptotic analysis tells us that a hash table should be faster than a B-tree, in practice the B-tree may be superior for storage of records in a large database system. That is, despite the fact that such analyses in some sense measure costs, in practical terms they do not in general concern resources, but only concern how the resources change with scale; and may have little bearing on performance in practice.

Moreover, an analysis is no more reliable than its assumptions. In an analysis of a data structure, the data must be modelled in some way, perhaps with simplifications to make the analysis tractable; but there is no guarantee that the modelling is realistic. The subjective elements of an analysis are just as significant as they are in an experimental design, due to the need for assumptions about properties such as machine behaviour and data distribution. Even the most fundamental assumptions, such as that analyses concern the number of instructions to be executed, may in some cases be inappropriate for a machine in which memory or disk references require thousands of instruction cycles and are the dominant practical cost.

The assumptions, in most research, flow from properties of the task the algorithm is being considered for. It may well be, for example, that a network analysis method has an utterly intractable worst case, but it also may be that this case cannot arise in the contexts that inspired the research, or is extraordinarily improbable. In general, algorithms are not applied to random input, and thus there is an important distinction between asymptotic costs in principle, asymptotic costs in plausible contexts, and actual costs in practice—each of which can be the subject of an informative analysis.

Analytical results can be powerful indeed—with, in some cases, implications for performance in practice on all machines for all time—but, as also discussed in Chap. 4, they are not necessarily sufficient by themselves.

Chapter 11
Graphs, Figures, and Tables

Graphical excellence is that which gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space.

Edward R. Tufte
The Visual Display of Quantitative Information

"*And what is the use of a book*", *thought Alice,* "*without pictures or conversations?*"

Lewis Carroll
Alice in Wonderland

Well-chosen illustrations breathe life into a paper, giving the reader interesting visual elements to browse and highlighting the central results and ideas. A typical figure consists of visual matter such as a graph or diagram, or of textual matter such as a table, algorithm, or, less commonly, complex mathematics. Some information is best presented in a pictorial form, such as a graph or figure, to show trends and relationships. Other information is best as a table, to show regularities. This chapter concerns style issues related to such material.

Graphs

Graphs are often the best way to present numerical results. Use graphs wherever appropriate, to elegantly summarize numbers and make obvious the behaviour and trends that you wish to demonstrate. If you must list the numbers as well, put a detailed table of results in an appendix, but in many cases the trend is the interesting outcome; the numbers may be of only transient significance and can be omitted.

You should present information because it is supporting evidence for a hypothesis, not because it is an output of some program. Don't flood your paper with statistics, even in graphical form, and avoid repetition; each graph should convey interesting new information. It is easy to generate reams of numbers by running software with different combinations of parameters, but, even though these numbers may contribute to your analysis and understanding of the phenomena being observed, they are unlikely to be of value to a reader.

© Springer-Verlag London 2014

J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9_11

Graphs should be simple, with no more than a few plotted lines and a minimum of clutter. If the graph is being used to demonstrate variation in output values for a range of input values, the horizontal or x-axis should be used for the parameter being varied, or the input; the vertical or y-axis is for the function of the parameter, or the output. Plotted lines of discrete data should always have points marked by distinctive marks such as circles, boxes, or triangles.

Consider using greys, colours, or line thickness rather than dots and dashes to distinguish between lines. If you use shades of grey to distinguish different elements in the graph, ensure that the shades are sufficiently distinct; lines in lighter grey sometimes need to be a little thicker than other lines. Greys are preferable to cross-hatching for filled-in areas in a figure, as the latter can create the optical illusion of shimmering and does not always print or photocopy well.

Colours are more eye-catching than are greys, but, in a graph, do not necessarily communicate better. First, some journals do not print in colour. Second, colours can render inconsistently in different media; for example, a vivid yellow on a screen may be almost invisible on paper. A related issue is that some colours are more conspicuous than others, in particular red—a reader may not even notice information in mild colours if a bright red is present. Third, greys are emotionally neutral, and thus don't carry the subconscious messages that colours can. In a sales pitch, colour is used to make implicit statements about products or methods, and to persuade through visual excitement. In a write-up, where the goal is persuade through logic and evidence, such an approach is inappropriate.

Minimize use of unnecessary elements and remove all decoration. Are the secondary ticks on the axes useful? If not, discard them. Is a legend necessary? If not, remove it, and label the lines directly. Do the captions have to be in a large font? If not, diminish them. Are the fonts and font sizes different to those of the rest of the paper? If so, change them. Axes should be inconspicuous; ink should be used for data, not dressing. Gridlines and boxing are other forms of unnecessary ornamentation. Secondary marks, such as axis ticks, should be a little lighter than the other elements. The lines and other elements should be of similar weight—don't mix a large, bold font with lightly drawn lines, for example.

Many of the commonly used graphing tools provide features that are only rarely of value; worse, some of these features are invoked by default.[1] Poor versions of a graph are shown in Fig. 11.1, with revisions of it in Fig. 11.2. A slightly more complex graph is shown in Fig. 11.3. See also the graphs in Figs. 11.4, 11.5, 11.6, 11.7, 15.2, 15.3, and 15.4.

Note, though, that these examples illustrate just a few aspects of design of graphs. With the breadth of kinds of data and result reported by researchers, and also the breadth of tools available for interpreting and presenting data, it would not be sensible to attempt to present a comprehensive set of examples; instead, these graphs are

[1] Some widely used tools have truly strange default settings, currently including unusual colours for labels and legends, massive fonts on the axis labels, and dense grids across the background. When a paper includes a graph produced in that way, the immediate message is that the author is uninterested in trying to communicate well.

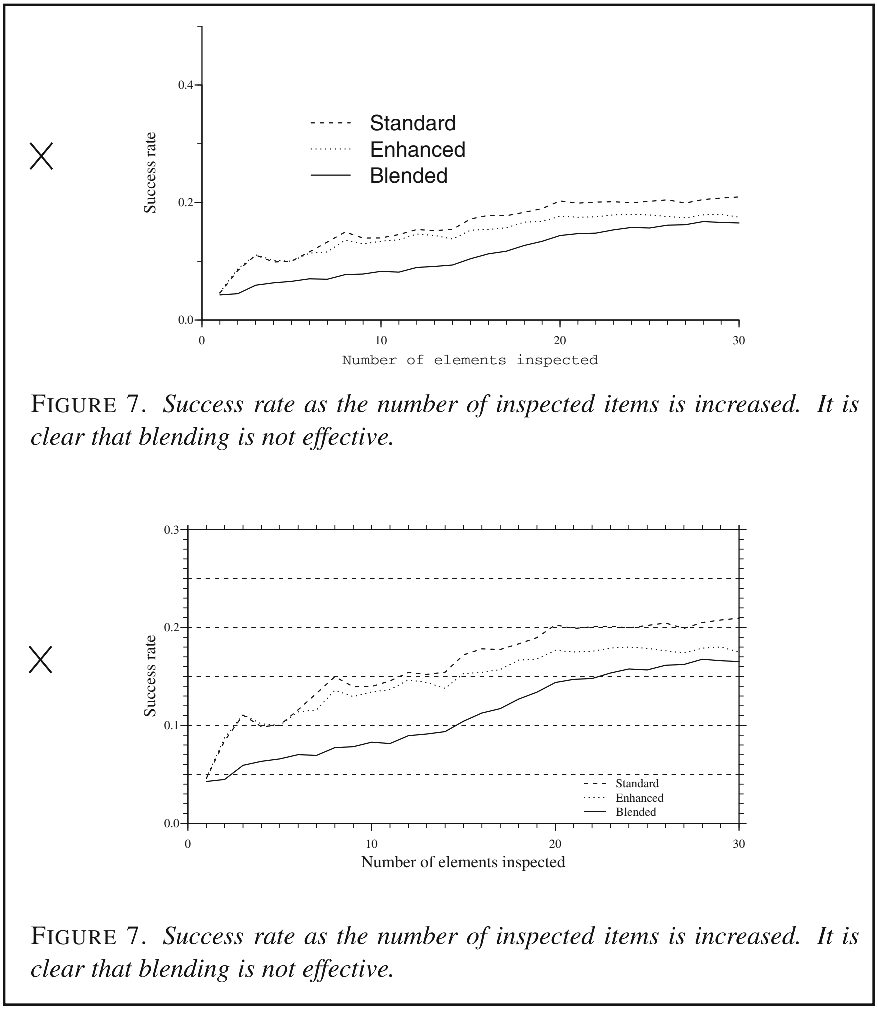


Fig. 11.1 Badly designed graphs. These graphs show the same data. In the *upper* version, poor use has been made of the vertical space available, and the legend is awkwardly placed. Fonts and size are changed unnecessarily, and are inconsistent with the main text. In the *lower* version, the vertical scaling and fonts have been partially corrected, but unnecessary ornamentation has been introduced, and the fonts are still too small. The grid lines and heavy border now greatly outweigh the data being presented

primarily a demonstration of the kinds of improvement that a careful researcher can achieve. The underlying point is that care is needed. A graph can be the central element of a write-up, the place where the results are demonstrated and the case made for support of a hypothesis. Great care is needed to ensure that graphs are effective at communicating results.

In these examples, the graphs are rectangular rather than square, with the legends placed in spare space within the body of the graph. The legend needs to be placed

FIGURE 7. *Success rate as the number of inspected items is increased. It is clear that blending is not effective.*

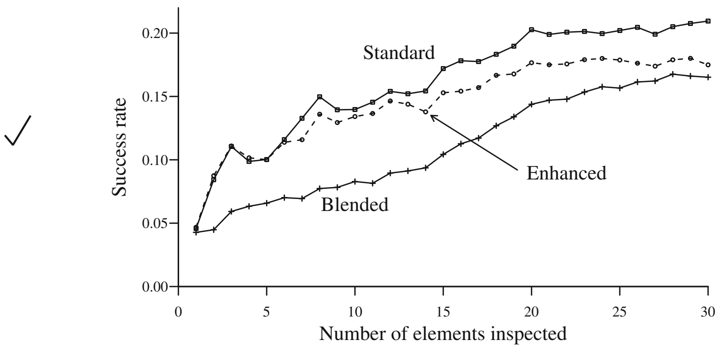


FIGURE 7. *Success rate as the number of inspected items is increased. It is clear that blending is not effective.*

Fig. 11.2 Graphs reconsidered. These graphs show the same data as those on the previous page. Vertical scale is now completely corrected, and unnecessary tick marks have been removed. In the *lower* version, the data lines are stronger and the legend has been replaced with direct labelling. Line ticks have been introduced to reflect the fact that the data is discrete, that is, non-integer values are not meaningful

where it can't be confused with other material; default placement may mean that the legend obscures part of a curve. The emphasis is on creating as much space as possible for presentation of data, while other elements are held to a minimum.

Imagination may be needed to allow the desired picture to emerge. Logarithmic axes are useful because they show behaviour at different orders of magnitude. An example of changing to a logarithmic axis is shown in Fig. 11.4. Graphs with logarithmic axes are also useful when plotting problem size against algorithm running

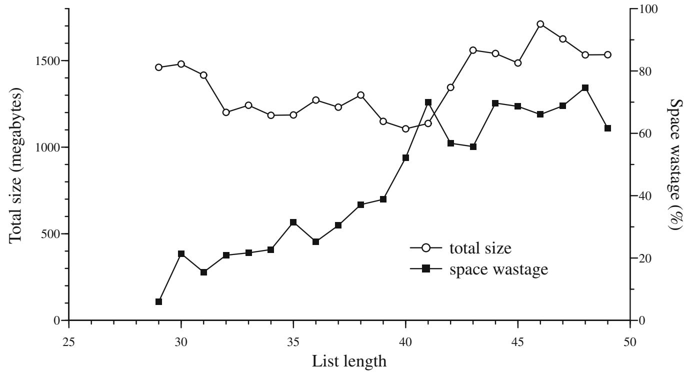


FIGURE 2. *Size and space wastage as a function of average list length.*

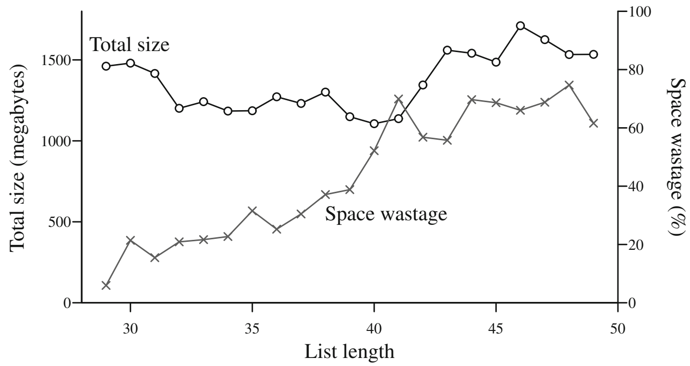


FIGURE 2. *Size and space wastage as a function of average list length.*

Fig. 11.3 Two functions plotted on one graph. It is necessary to label the axes to correspond with the curves; otherwise it would be difficult to identify which curve matched which axis. Axes are labelled, so that a reader can easily identify which one matches which curve. The *lower* version is a revision of the *upper*, with distracting elements removed or de-emphasised and several other minor alterations

time, as different asymptotic growth rates give straight lines of different slope. In particular, if variables x and y are related by $y = ax^c$, then $\log y = \log a + c \log x$, that is, the relationship of the logarithms is linear. If the relationship is more complex, some sort of transformation on the data may yield a straight line or some other simple curve.

Log scaling is not always appropriate. If one algorithm is 30 % faster than another at all scales, then, depending on overall scale, their performance could be almost indistinguishable on a log-log graph, although the constant-sized gap would be

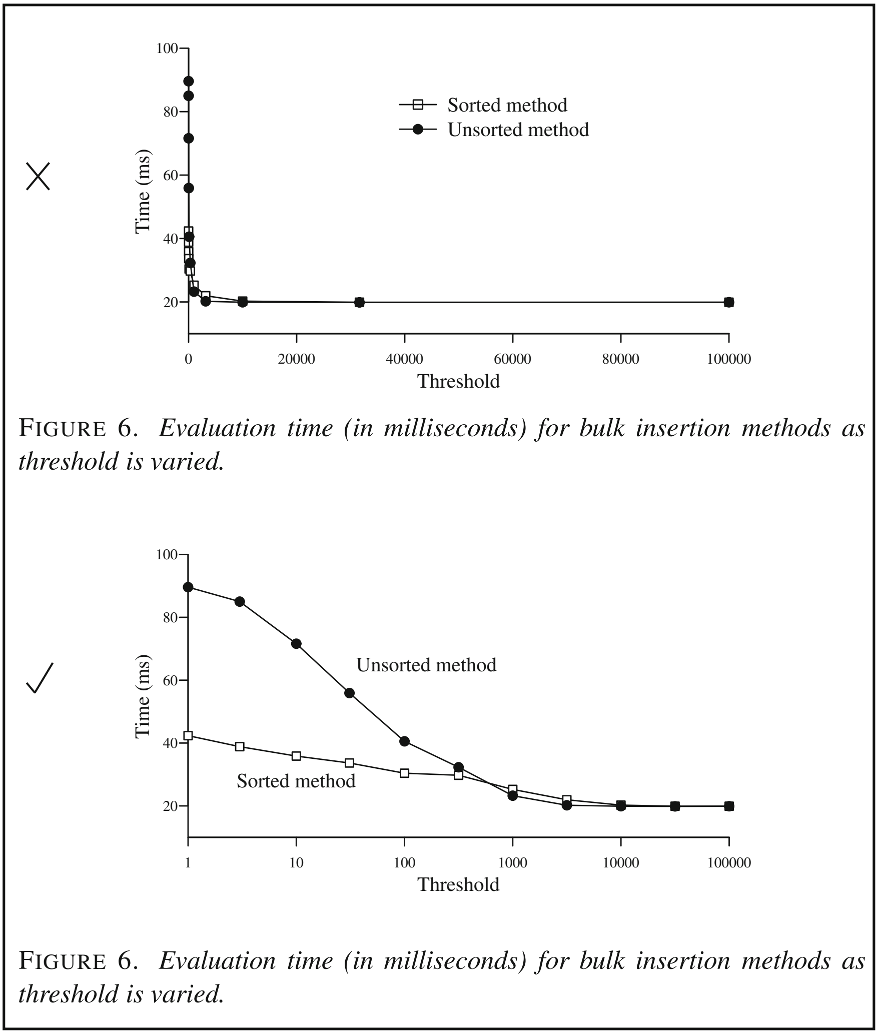


Fig. 11.4 Choice of axis scaling. For these graphs showing the same data, in the lower graph the logarithmic scaling on the x-axis allows the behaviour for small thresholds to be seen

informative. Even if one algorithm is twice as fast as the other, a log-log graph will show one line just a little below the other.

In some cases, data that seems innately tabular can be represented as a graph. Often a bar graph is suitable because the items being compared are not ordered, as shown in the graph in Fig. 11.5. (Such data should not be represented by a line chart, in which the points are connected into a continuous line, which would imply that the axes were related by a function.) A richer example is shown in Fig. 11.6. For the more complex problem of comparing space and time simultaneously, the graph in Fig. 11.7 would serve well.

Data set	Method	
	A	B
Small, random	11.5	11.6
Large, random	27.9	17.1
Small, clustered	9.7	8.2
Large, clustered	24.0	13.5
All documents	49.4	60.1
First 1000	21.1	35.4
Last 1000	1.0	5.5

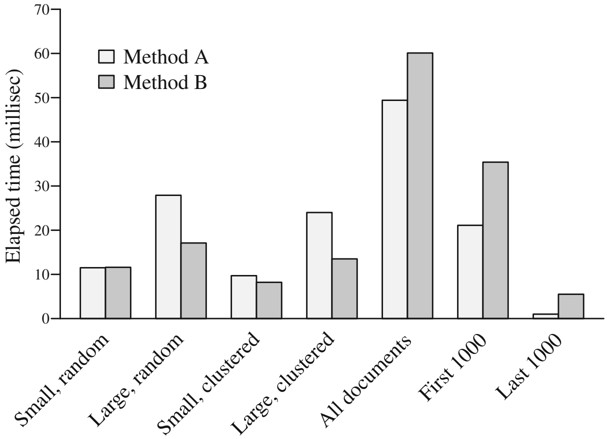


FIGURE 2. *Elapsed time (milliseconds) for methods A and B applied to data sets 1–7.*

Fig. 11.5 A table compared to a graph. The data shows how two methods compare over seven experiments. The graph is a better choice for this data because the pattern is more obvious

Graph-drawing tools allow bar graphs to be three dimensional, but the addition of depth is deceptive; if one bar is twice the height of another, the depth exaggerates the difference.

Graphs are used to illustrate change in one parameter as another is varied. In some cases more than two parameters can interact in complex ways. If two parameters, say B and C, depend on another, A, then a good solution is to plot A on the x-axis and have two y axes, one for each of B and C, as shown in the graph in Fig. 11.3.

If two parameters, say D and E, jointly determine a third, F, in some complex way—thus describing a three-dimensional space—the problem is more difficult. Use of a three-dimensional graph is an option, in which varying both D and E produces

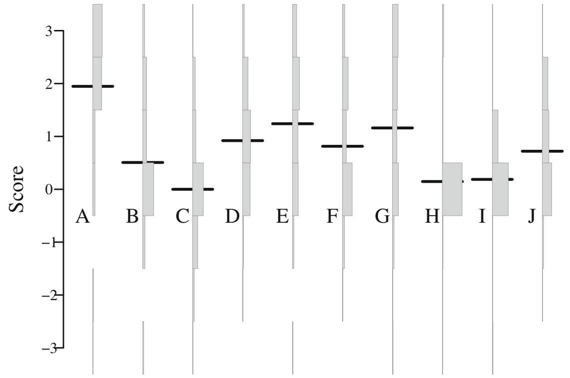


FIGURE 2. *Average score in each category. There were 75 responses overall. The proportion of responses in each category, for the possible scores of* −3, −2, −1, 0, 1, 2, *and* 3, *is shown as a vertical histogram. The solid bar is the mean in each case.*

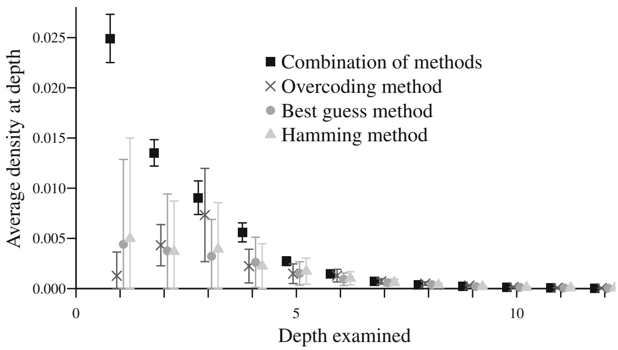


FIGURE 9. *Range of scores with each method, at each depth. The principal mark in each range is the average score. As can be seen, each method returns results within a reasonably narrow band, but they are surprisingly different from each other. Combination is highly effective in this case.*

Fig. 11.6 Further bargraphs. The *upper* graph shows an approach to comparing distributions across a set of related statistics. The *lower* graph has error bars to show range and scale; however, while it is a reasonable initial presentation of this data, it could easily be improved

a landscape of F values. Such graphs can be powerful explanatory tools, but should not be used merely because they are dramatic or eye-catching. Another approach is to experimentally plot D against F for several fixed values of E, and use these results to choose an E value that yields a representative graph; and similarly vary E for several fixed D, to choose a representative D.

TABLE 8.4. *Tradeoff of space against time for methods A to G.*

Method	Space (%)	Time (ms)
A	1.0	7,564.5
B	31.7	895.6
C	44.7	458.4
D	97.8	71.8
E	158.1	18.9
F	173.7	1.4
G	300.0	0.9

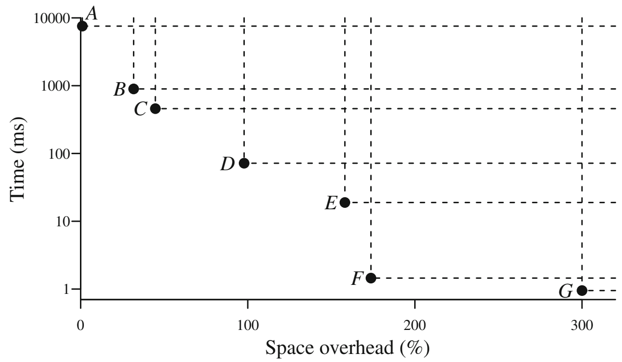


FIGURE 8.4. *Tradeoff of space against time for methods A to G. The boxed area to the right and above each point is of unacceptable performance: any method in that area will be less efficient with respect to both space and time than the point at the box's corner.*

Fig. 11.7 Another table compared to a graph. The data shows how different methods compare with respect to space and time. The table is difficult to interpret. The graph illustrates the frontier of efficiency, that is, it describes the region (towards the *bottom left*) that would represent an improvement on existing methods

Where several methods of achieving the same aim are being illustrated, the axes in each graph should have the same scale. For example, if you are comparing different data structures and a separate graph is used for each one, the axes should be consistent from one graph to the next. That is, if y, say, ranges from 0 to 80 on one graph, it should also range from 0 to 80 on the other, to allow direct comparison between the methods. Comparison is easier with several (but not too many) lines on one graph.

In general, for a quantity that is wholly positive and measured on a ratio scale (where, say, a doubling in the measure corresponds to a doubling in the underlying quality measured), the Y axis should start from 0, as starting from a higher value can exaggerate degrees of change or difference. If the axis is started at a higher value for clarity, the reader should be alerted to this in the caption.

Beware of using graphs to make unsupported claims. For example, consider the line labelled "space wastage" in the graph in Fig. 11.3: it would not be possible to identify the slope of this line with any confidence, nor identify it as a particular kind of curve. The only reasonable inference would be that increasing list length generally increases space wastage.

There are many good software packages for drawing graphs. Valuable features include:

- Placing of several lines on one graph.
- A range of symbols (such as crosses, squares, and triangles) for marking points.
- The ability to create custom symbols of custom size for marking points.
- Optional connection of points with solid, dotted, or dashed lines, and optional omission of the point marks.
- The ability to place text at specified places in the graph.
- Multiple font sizes and line thicknesses.
- The ability to use the same fonts as in the body of the text.
- Availability of greys and colours.
- Optional logarithmic or exponential scaling on both axes.
- Axis editing, to specify where the ticks are placed, how many digits of precision to use, and what range to cover.
- The ability to move and rotate the legend or key, line labels, axis labels, and the graph label.
- The ability to apply simple functions or external programs to (x, y) values.
- The ability to lay out mathematical symbols and basic expressions.
- The generation of images in vector format (postscript, PDF, SVG) rather than raster format (jpg, gif, png), to allow subsequent rescaling.

Most of these features were used in the example graphs in this chapter.

Graphs and diagrams attract the attention of readers, so should be reserved for material that is central to the paper.

Diagrams

Diagrams serve many purposes in computing papers. They illustrate processes or architectures, explain data structures and algorithms, present relationships, visualize data, and show examples of interfaces. There are areas of computer science in which the diagrams are, in some sense, the result being presented in the paper: entity-relationship models are diagrams conforming to a well-defined notation, for example, and automata are often described by diagrams. A novel visualization of a massive dataset can be a potent demonstration of previously unknown properties or behaviour.

Many areas of research have highly developed conventions and standards for diagrams. Browsing relevant papers in the same area as your work should you give a good idea of what elements a diagram should incorporate and of how it should be presented.

Broadly speaking, diagrams are used to show either a structure, a process, a relationship, or a state. Although these are high-level distinctions, they are valuable because a common mistake in design of diagrams is to attempt to combine these purposes inappropriately. For example, a schematic showing data flow in an architecture is likely to be unclear if control flow is also illustrated.

Some forms of diagram and illustration are automatically generated by tools from data. In particular, mechanisms for data visualization can be used to build rich images. Here, however, I am primarily concerned with the line drawings that form a key part of many papers. While automatic tools for generating diagrams can be used to produce a wide variety of representations from the same underlying data, some of which will be dramatically more effective than others, the richness and diversity of these tools—not to mention the rate at which they are developing—means that such diagrams are beyond the scope of this book.

To design a diagram that is to be created with a manual tool, the first step is often to do initial sketches by hand, on paper. This early stage is the appropriate time to balance the diagram, by checking that it is well-proportioned, makes good use of the space, is laid out well and doesn't have the elements bunched to one side, and is arranged so that the relative sizes of the elements look reasonable. However, never submit a paper with a hand-drawn diagram unless it has been prepared by a professional; almost any diagram can be drawn well with the tools available for a typical computer.

A diagram should not be too dark; keep it as sparse as possible. This is best achieved by eliminating all clutter. A diagram does not have to be faithful to every detail of the concept being illustrated; fine details can always be clarified in the supporting text and even the best diagram requires some explanation. Use meaningful labels, which should if possible be displayed horizontally, and make the point size and font of the labels similar to that of the other text. As for text in general, there should be no more than two or three fonts and font sizes.

Lines should not be too heavy, but at most a little thicker than the lines used to draw the text font. Shades of grey can be used to distinguish between solids but are not as effective for distinguishing between lines, and don't use shades that are too light or too similar to each other. Pictorial elements should be used consistently, so that, for example, arrows and lines of the same kind have the same meaning. Use shading rather than cross-hatching. Colour can be highly effective, especially if it is used sparingly; but, as for graphs, do not use colour if the paper will ultimately be printed in black-and-white, or if the sole reason is to make the paper more attractive to look at. If arrows are used to show arcs as well as to point at features, distinguish them by, say, using dashed lines in one case and solid lines in another. Lines should not touch each other unless separating them would create an unnatural break. Thus, for example, there should usually be a gap between an arrowhead and the element the arrow is pointing at.

Diagrams, like graphs, can add greatly to the clarity of a paper. But be aware that the design of good diagrams is not easy. Expect to revise your pictures as often as you would your writing. Some simple diagrams are shown in Figs. 11.8 and 11.17. A weak diagram, shown in Fig. 11.9, is revised in Fig. 11.10.

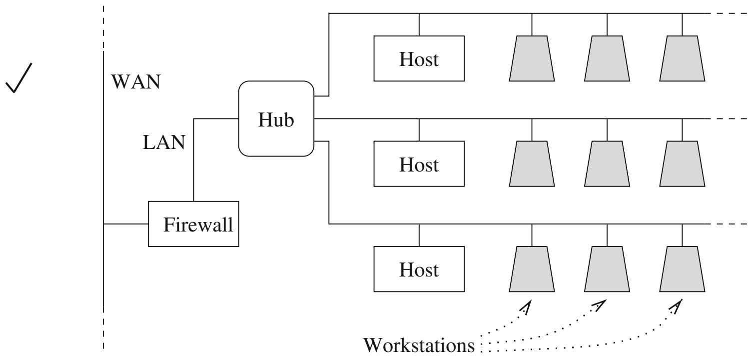


FIGURE C. *Revised network, incorporating firewall and hub with hosts and workstations on separate cables.*

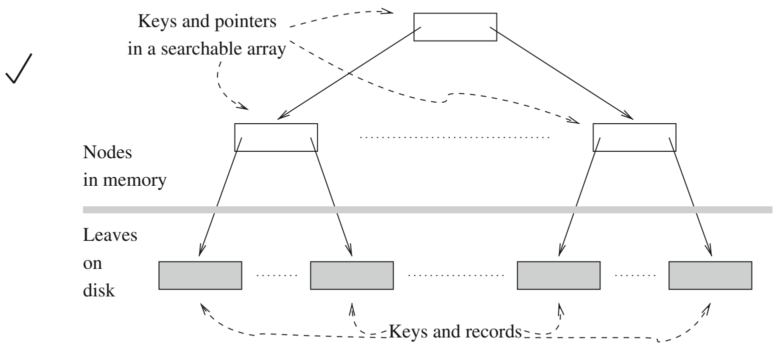


FIGURE 1.3. *Tree data structure, showing internal nodes in memory and external leaves on disk; omitted nodes are indicated by dotted lines. Nodes allow fast search and contain only keys and pointers. Leaves use compact storage and contain the records.*

Fig. 11.8 Shading and dashing in diagrams. In these illustrations, the shading and dashing makes it clear whether entities are of the same kind

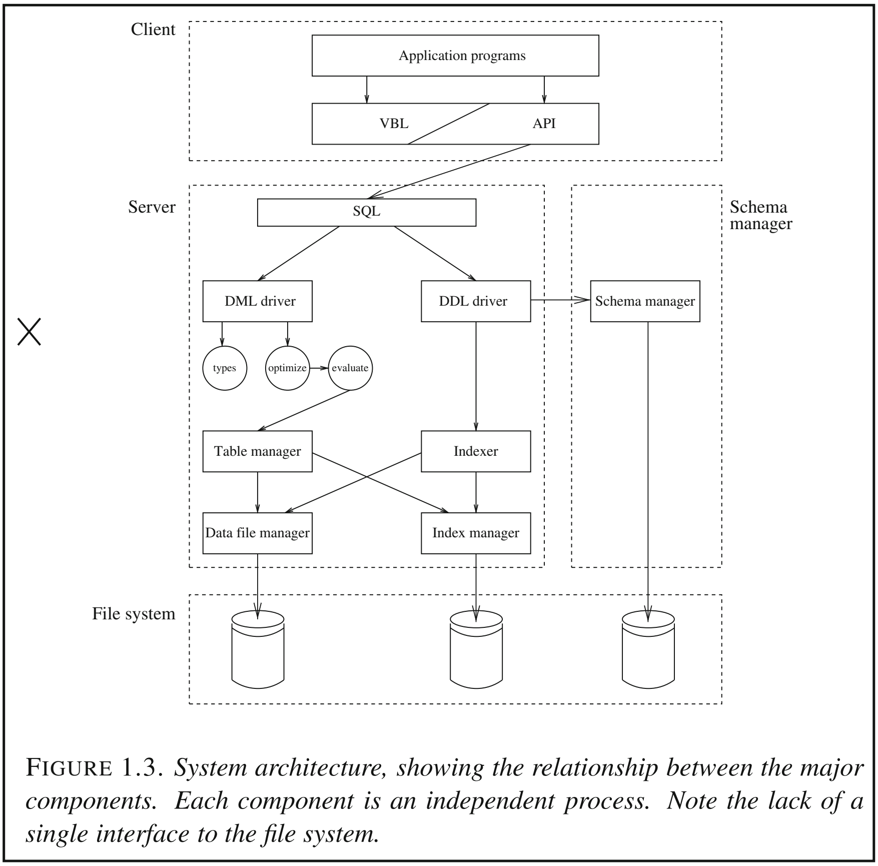


Fig. 11.9 Too much clutter. A carefully constructed figure, but flawed. The font is too small and the lines are too light. The overall structure (the division into four major components) is probably the most interesting feature, but other details are more highly emphasized. Some of the internal detail should be omitted. The arrows add little information, and should point both ways, because information flows out as well as in

Diagrams illustrating system structure often seem to be poor. In too many of these pictures the symbolism is inconsistent: boxes have different meanings in different places, lines represent both control flow and data flow, objects of primary interest are not distinguished from minor components, and so on. Unnecessary elements are included, such as cheesy clip-art or computer components that are irrelevant to the system. A poor structural diagram is shown in Fig. 11.11, with a revision in Fig. 11.12.

Another form of diagram is an image, such as a photograph or screenshot. Photographs are rare in computing papers and, sadly, are often not presented at a high standard. If you do need to include a photograph, ensure that it will render well in

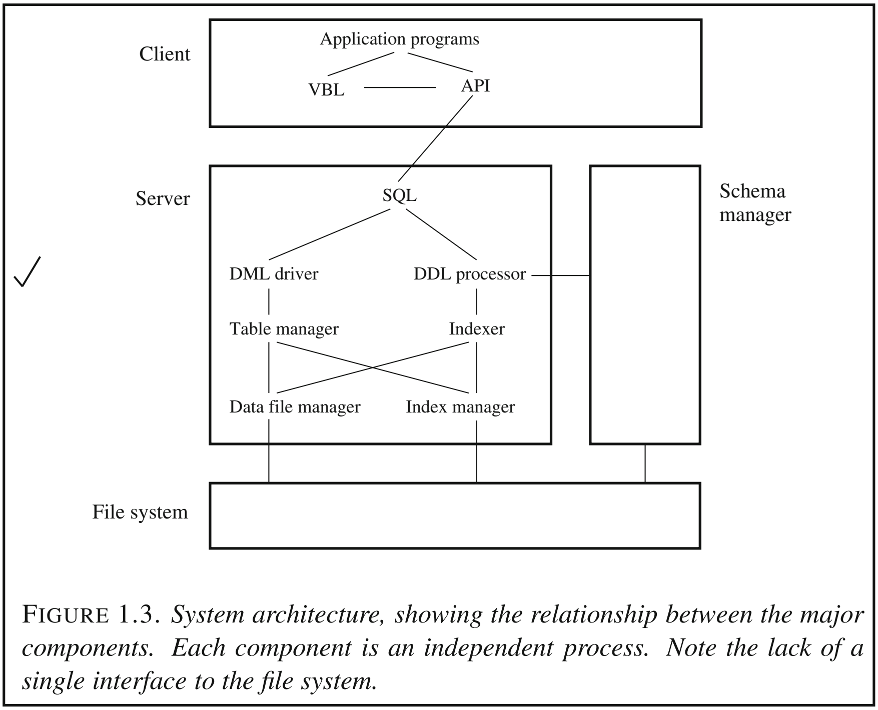


Fig. 11.10 Clutter simplified. A revision of the figure in Fig. 11.9. The overall structure is more prominent, while some minor features have been discarded and the unnecessary inner boxes have been removed. Use of shading would give further improvements

black-and-white (a vibrant colour photograph can be surprisingly dull when transformed to greyscale), and ensure that it is of sufficient resolution. The problem of resolution is even more acute for screenshots, as the pixel density of a screen is much lower than that of print. Screenshots are used unnecessarily in far too many papers, and only occasionally illustrate anything of fundamental interest.[2]

Illustrations are covered by copyright; figures from another source can only be re-used with permission of the author and the publisher of the original. If you wish to re-use a figure, get permission to do so and identify the original author and source, preferably in the caption. You may also need to include the original copyright statement.

[2] Sometimes authors include material in a paper that strongly suggests that the work is not of high quality. An example is when authors use a tool to create a diagram, but, instead of exporting the diagram in a portable, scalable format such as PDF, take a screenshot of the tool. Such screenshots can include irrelevant materials such as the tool's menus, are low resolution, don't print well, and look incompetent.

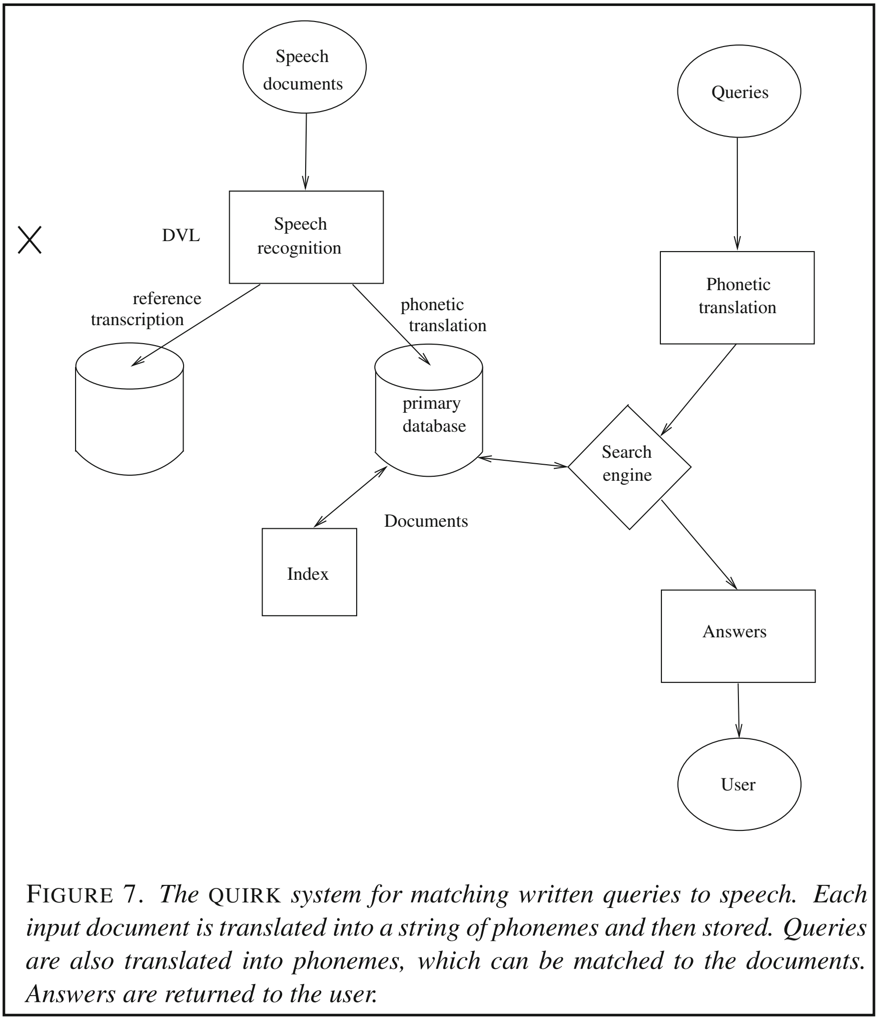


Fig. 11.11 Disorganization. This figure is poorly designed. The elements are inconsistent; data is in both ovals and boxes, and some lines represent data flow while one represents a transformation. The arrowheads touch other lines, creating messy intersections. There is unnecessary material such as the auxiliary databases (write-only, apparently) and the user

Tables

Tables are used for presentation of information that is unsuitable for graphs or figures, such as the properties of each of a series of datasets or data where the exact values are important. A table may also be used instead of a graph when only a few values are to be shown. The tables in Figs. 11.13, 11.14, 11.15, and 11.16 have appropriate

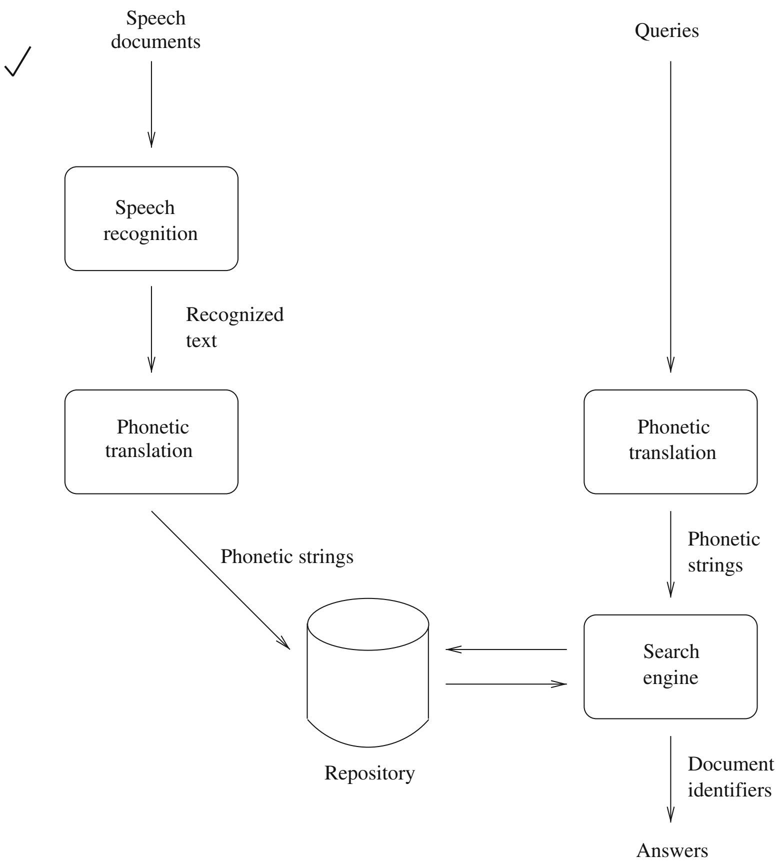


FIGURE 7. *The* QUIRK *system for matching written queries to speech. Each input document is translated into a string of phonemes and then stored. Queries are also translated into phonemes, which can be matched to the documents.*

Fig. 11.12 Clarification. A revision of the figure in Fig. 11.11. The parallels between document processing and query processing are emphasized, and the unnecessary material has been removed. The two-headed arrow is replaced by two arrows, to show that data is exchanged

content, although two are poorly laid out. The table in Fig. 11.5 is of debatable value, as the graph explains the data well and the precise timings may not be interesting. The table in Fig. 11.7 is much less informative than the graph; in this case the table should not be used.

A well-designed table has a logical hierarchical structure. Simple tables are an arrangement of columns and rows, in which each column has a heading at the top and each row has a label or stub at the left. In more complex tables, columns and

TABLE 6. *Statistics of text collections used in experiments.*

STATISTICS	SMALL	LARGE
Characters	18,621	1,231,109
Words	2,060	173,145
After stopping	1,200	98,234
Index size	1.31 Kb	109.0 Kb

TABLE 6. *Statistics of text collections used in experiments.*

	Collection	
	Small	Large
File size (Kb)	18.2	1,202.3
Index size (Kb)	1.3	109.0
Number of words	2,060	173,145
— after stopping	1,200	98,234

Fig. 11.13 Two versions of a table. The *upper* version is poor. Because there is no sense of table hierarchy—all the elements are at the same level—headings and content must be differentiated by case. Different units have been used for file sizes in different lines (assuming characters are one byte each). Units are mentioned explicitly in the last line, and the precision is inconsistent. The heading of the first column is unnecessary and the table has too many horizontal lines. In the *lower* version there are no vertical lines. Rows of the same type are now adjacent so that they can be compared by the reader, and the hierarchy between the total number of words and the number of words excluding stopwords is visually indicated. Note that the values of different units do not need to be vertically aligned on the decimal point or presented with the same precision

rows may be partitioned or have internal structure. The hierarchy can be indicated in several ways: rows or columns can be separated by double lines, single lines, or white space; headings can span several columns; labels can refer to several rows. Deeper structure—which is sometimes necessary but is usually unwise—can be indicated by markup within the table such as embedded headings. (A complex table is shown in Fig. 11.14.) The items below a column head should be of the same kind or about the same thing. Items to the right of a row label should all be properties of that label. The column of labels does not need to have a heading, but this position, the top-left corner of the table, should not be a label for the other column headings. If there is no heading for the column of labels, leave the position blank.

Tables should be open and uncluttered, with ample white space. Don't have too many horizontal or vertical rules. In particular, there is no need to have a rule between

TABLE 2.1. *Impact on performance (processing time and effectiveness) of varying each of the three parameters in turn, for both data sets. Default parameter values are shown in parentheses. Note that $p =$ 100,000 is not meaningful for the data set* SINGLE.

Parameter	Data set			
	SINGLE		MULTIPLE	
	CPU (msec)	Effective (%)	CPU (ms)	Effective (%)
n ($k =$ 10, $p =$ 100)				
2	57.5	55.5	174.2	22.2
3	21.5	50.4	79.4	19.9
4	16.9	47.5	66.1	16.3
k ($n =$ 2, $p =$ 100)				
10	57.5	55.5	174.2	22.2
100	60.0	56.1	163.1	21.3
1000	111.3	55.9	228.8	21.4
p ($n =$ 2, $k =$ 10)				
100	57.5	55.5	174.2	22.2
1000	13.8	12.6	19.8	2.1
10,000	84.5	56.0	126.4	6.3
100,000	—	—	290.7	21.9

Fig. 11.14 Table with a deep hierarchy. There are two columns, one for parameters and one for data sets. The latter is divided into two columns, one for each data set. Each data set has two columns of figures. There are four rows, one of headings and one for each of the parameters n, k, and p. Each of these is subdivided. Note that even this rather complex table does not require vertical rules. This table might benefit from being separated into parts, but it is helpful to have all the data together. There are insufficient data points for each combination of parameters to justify use of a graph

every row or column. (An example of this error is shown in Fig. 11.13.) But do have rules between groups of rows, and, in rare cases, between groups of columns, to act as guides and to separate items that don't belong together. Don't make tables too dense. Rather than cram in a large number of columns, have two tables, or, even better, be selective about the information you present. In most tables no position should be blank; if there is no applicable value, put in a dash, and explain somewhere what it means. Values of the same units in a column should be aligned in a logical way. Numbers should be aligned on the decimal point.

Using tables to show function values at different points is usually not a good idea because graphs serve this purpose well; a possible exception is when a function only has two or three values, in which case a graph would be too simple or sparse to be of interest. If the table or graph shows only a simple relationship, consider stating the relationship and omitting the diagram.

TABLE 11. *Resources used during compression and indexing. Only the vocabulary is constructed in the first pass; the other structures are built in the second pass.*

×

Pass	Output	Size Mb	%	CPU Hr:Min	Mem Mb
Pass 1:					
Compression	Model	4.2	0.2	2:37	25.6
Inversion	Vocabulary	6.4	0.3	3:02	18.7
Overhead				0:19	2.5
Total		10.6	0.5	5:58	46.8
Pass 2:					
Compression	Text	605.1	29.4	3:27	25.6
	Doc. map	2.8	0.1		
Inversion	Index	132.2	6.4	5:25	162.1
	Index map	2.1	0.1		
	Doc. lens	2.8	0.1		
	Approx. lens	0.7	0.0		
Overhead				0:23	2.5
Total		745.8	36.3	9:15	190.2
Overall		756.4	36.8	15:13	190.2

Fig. 11.15 Jumbled table. Columns have been crammed together and are hard to understand. The numbers representing quantities of the same kind don't line up vertically. The percentage column is mysterious, since it doesn't sum to 100. It seems unlikely that all the detail is interesting; consider in particular the "Index map", "Doc. lens", and "Appr. lens" rows, which could presumably be gathered into a single row with a label such as "Other" or discarded altogether

× As illustrated in Table 6, temporary space requirements were 60 % to 65 % of the data size.

✓ In our experiments, temporary space requirements were 60 % to 65 % of the data size.

Small tables can be part of the running text, displayed in the same way as mathematics. Larger tables should be labelled and positioned at the top or bottom of a page.

Tables are used not only for numbers but for analysis of alternatives. For example, a list of approaches to system modelling could be compared in a table, one row per approach, with columns used for positives, negatives, and number of known successful applications of the approach. In such tables, each cell may contain a brief paragraph of text and a single table may occupy a page or more, and thus the overall appearance is quite different to that of a table of numerical data. Nonetheless, the same design guidelines apply.

TABLE 11. *Resources used during compression and indexing. Only the vocabulary is constructed in the first pass; the other structures are built in the second pass.*

Task	Size (Mb)	CPU (Hr:Min)	Memory (Mb)
Pass 1:			
Compression	4.2	2:37	25.6
Inversion	6.4	3:02	18.7
Overhead	—	0:19	2.5
✓ Total	10.6	5:58	46.8
Pass 2:			
Compression	607.9	3:27	25.6
Inversion	137.8	5:25	162.1
Overhead	—	0:23	2.5
Total	745.8	9:15	190.2
Overall	756.4	15:13	190.2

Fig. 11.16 Table simplified. A revision of the table in Fig. 11.15. The confusing percentage column has been deleted. The "Output" column has been deleted, along with the rows corresponding to the output sub-categories; since most of the values in these sub-categories are small, they are relatively unimportant and could if necessary be discussed in the text

Understanding a table of any complexity is hard work. For presentation of results, graphs or explanatory text are preferable; have a table to which the interested reader can refer, but don't rely on a table to convey essential information.

Captions and Labels

Captions and labels should be informative. Though it is common for captions to be only a few words, it is preferable for captions to fully describe the figure's major elements. (A diagram and caption are shown in Fig. 11.17.) Full captions assist the reader who is skimming the paper or referring back to earlier figures and tables. Use either minimum or maximum capitalization, but minimum is better, particularly if the caption is a description rather than a label. Use italics for the caption so that it is distinct from other text. The caption is usually placed below a figure, but above a table.

Since figures and tables should be fairly self-contained, the caption is an appropriate place to explain important details, especially since these would otherwise

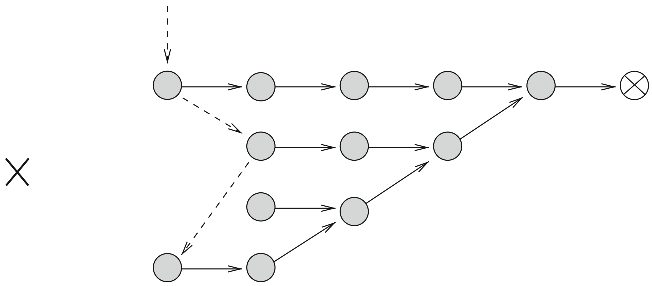


FIGURE 5. *Fan data structure.*

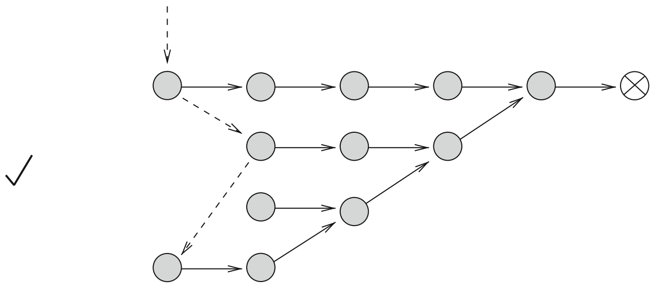


FIGURE 5. *Fan data structure, of lists with a common tail. The crossed node is a sentinel. Solid lines are within-list pointers. Dashed lines are inter-list pointers.*

Fig. 11.17 Styles of caption. For these identical figures, the lower caption is preferable because it allows the figure to be less dependent on the paper's text

interrupt the flow of the main text. For example, a graph might show running time for an algorithm over various data sets; the caption could include parameter values. The caption can also be used to expand abbreviations or notation used in headings.

Each figure and table should be numbered to allow easy reference. If your word processor does not provide automatic numbering, you must number the figures yourself. A figure is usually at the top or the bottom of a page, or on a page by itself, to set it apart from ordinary text. A figure or table should always be introduced and discussed in the main text, preferably just before or on the page on which it occurs. If you don't have anything to say about a figure or table, leave it out.

Axes, Labels, and Headings

The space constraints on axes, labels, and headings may mean that some terms have to be abbreviated; for example, see the table in Fig. 11.15. It is helpful to state these terms in full in the text discussing the illustration, but do so in a natural way.

✗ The abbreviations "comp.", "doc.", and "map." stand for "compression", "document", and "mapping table" respectively.

✓ The effect of compression on the documents and the mapping table is illustrated in the second and third rows.

Where appropriate, units should be stated in labels. Write "Size (bytes)", not just "Size".

Some readers get confused by scaling on axes and labels. Suppose, for example, that an axis is labelled as "CPU time (seconds $\times 10^{-2}$)". The convention is that the reader should multiply axis values by 10^{-2}, so that 50 means 0.5. But some readers may assume that the axis values have already been multiplied by 10^{-2}, so that they read 50 as 5000. In the text where the illustration is referenced, typical values can be discussed to avoid this problem. The problem also arises with graphs; it is helpful to include some representative numbers in the text, because graphs are hard to read with any precision.

✓ Figure 4 shows how time and space trade off as node size is varied; as can be seen, response of under a second is only possible when size exceeds 11 Kb.

Sometimes the terminology of a paper gets changed at a late stage, perhaps with a global substitution. Ensure that graphs and diagrams get updated too.

Chapter 12
Other Professional Writing

An expert is a person who has found out by his own painful experience all the mistakes that one can make in a very narrow field.

Niels Bohr

Many computing graduates find that, once they enter the profession, writing is a surprisingly large component of their daily work. Researchers expect to have to write papers, book chapters, grant applications, and so on; while computing professionals expect to write material such as project and code documentation, manuals, and acceptance records. However, they might well also find themselves having to write other expert material, such as project proposals, technical assessments, tenders, purchase recommendations, reports to managements, descriptive material for the Web, or any of a wide range of kinds of document that are required in large organizations.

Much of this book concerns computer science in an academic context. The challenges of writing tasks in general—job applications, emails, promotional material, and so on—are too diverse, and too different from academic writing, to consider here. However, there are a range of professional writing tasks that are focused on communication, and are written by the computing professional as an *expert*; some of the issues that arise with such tasks are like those of academic writing, and it is these that I now discuss.

Scoping the Task

A first step in most professional writing is to establish the *task*, that is, to figure out what the writing is meant to achieve. It may seem unnecessary to say something so obvious, but some authors appear to jump into writing tasks without any clear idea of what it is that they wish to accomplish, or what their “effort budget” is, or should be.

Yet knowledge of the “effort budget” is critical. Research papers are designed to form a permanent record, or perhaps to be examined, and there is a consistent requirement that they be complete and polished. In contrast, other writing may be

© Springer-Verlag London 2014

J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9_12

intended to be read just once (at a decision-making meeting, say), with opportunities for clarification; or it may be that the task must be completed within some limited time, and under pressure from other competing demands. Before you begin such a task, you need to know how much time can reasonably be committed to it, and then you can use that knowledge to decide where the effort will go—in other words, you need to decide whether compromises are needed, and if so what they will be.

If time is limited, don't begin by doing the writing that is easy; do the writing that the reader needs. And it will help the reader if you acknowledge that material is missing. That is, the effort needs to be spent wisely, with a focus on the document being useful and informative.

It is also critical that you decide what you trying to accomplish. One aspect of this is recognition of the nature of the task itself, which is the topic of the next section. Another aspect is the kind of outcome you are hoping for. You might be aiming to:

Record something (an event? a decision?), perhaps just for yourself or members of your team.

Inform someone, such as a manager or client, of an outcome, decision, failure, obstacle, or result.

Persuade or *convince* someone of the need to take action, or of the need for a decision to made.

Sometimes the author's intention is not apparent. Perhaps the initial impression a reader has of a document is that it appears to be intended to do no more than provide information, but the authors wrote it because they wanted a decision to be made. Or perhaps the author has argued that action is needed, but does not explain what the action might consist of or how urgent it is.[1] Or perhaps the document is fundamentally incoherent; for example, the author's aim was to present an argument for renewed funding of a system development, but the resulting document consists largely of a list of the system's features and shortcomings, and in the last few pages drifted into a discussion of why the user feedback was misguided. In all of these cases, the authors appear to have not thought through what it is that they are trying to accomplish.

Understanding the Task

Any list of typical professional expert writing tasks is likely to be incomplete: each workplace and context has its own demands, which shape the activities that are undertaken. It can help, though, to clearly identify the particular kind of task that you are doing. Examples of professional writing tasks that might be considered include:

[1] Which is not at all the same thing as how excited the author is. Some topics make certain authors highly agitated, so that they press for immediate action, even if there is no discernable urgency at all. The outcome in such cases is often that the author gets told to calm down and the report gets ignored—which is far from what the author wanted.

- Grant applications, funding proposals, and funding requests.
- Requests for or responses to tenders.
- Technical documentation: for example, designs, specifications, test specifications, code documentation, and handover and support documentation.
- Non-technical documentation: for example, requirements, project scope descriptions, project overviews, management documentation, risk documentation, acceptance records, and manuals.
- Requests for comments.
- Technical reports, including reports to management, incident reports, and papers recommending or assessing technical decisions.
- Plans, goals, and strategies.
- Patent applications, expert reports for legal proceedings, and other forms of legal document.
- Popular science articles.

This list consists, more or less, of work that could be regarded as undertaken in an expert capacity. The link between the items above is the concept of *professionalism*. While some documents may be intended for only a tiny audience, there is an expectation that a professional communication be balanced and objective.

Another aspect of the task is the mode of delivery. The document might be brief or long; delivered in person, or in writing only; available to a wide audience, or to just a few key recipients; require reflection and consideration, or require a quick response; be permanently available on the Web, or distributed to a defined group. There are many such variations, and all of them should influence your decisions as to the form of the document and how you will go about writing it.

Documentation

The topic of how to produce code and project documentation is explored in detail in a typical computer science degree. Elements include requirements, specifications, designs, test plans, manuals, and so on, both embedded in the code and as separate documents. Workplaces and projects usually have defined standards for these materials, and there are excellent texts on how to create them.

However, most computing professionals document much more than the software they develop. For example, some documentation tasks don't concern software under development, but instead concern software and hardware used by the development team. Consider overviews and handovers given to new employees. What information should, say, a new employee be given about the working environment? What should a departing employee be asked to record? Documentation might concern:

- Systems, installations, and hardware.
- Coding standards and workplace facilities.
- Algorithms and packages—not just what they are, but why they were chosen and what alternatives were considered.
- Online resources that can be used to guide decision-making.

Documentation for these purposes is likely to be written to the same guidelines as that used for specifications and so on. I've noted it here because it is an aspect of documentation that is sometimes overlooked, in particular the fact that it may need to explain *why* as well as *what*.

Technical Reports

Another common form of professional writing is the broad category of document that can be described as a technical report. In principle, everything from a research paper to code documentation might be in this category, but I'm using the term in a narrower sense: a report concerning some technique or technology. The reader of such a report might in some circumstances be another computer scientist or software engineer, but might in other circumstances be a manager, assistant, funder, lawyer, accountant, consumer, or client—that is, just about anybody. Thus a key difference between a research paper and a technical report, as I am using the term, is that the audience may be inexpert or uninformed about the field.

Perhaps the best way to explain is with an example. For instance, a technical report could concern the outcome or progress of an investigation, such as a report on the reasoning and arguments underlying the engineering decisions in a major software development. In circumstances where large developments can easily cost millions—and sometimes cost billions—management, investors, and other stakeholders expect that fundamental decisions will be given careful attention, and defensible processes may require that they be thoroughly documented.

A technical report of this kind might include:

Examination of choices and options. For example, the author might examine questions such as: What data structure to use? What criteria to apply? Which tools are appropriate? Should software be bought or written? Should it be open source or proprietary? If the former, what is the proposed support model, and what is the evidence that it will work?

Evidence of due consideration. What resources were used as input to these decisions? Or, alternatively, what resources are required to make these decisions, and what will they cost? What are the possible costs of *not* undertaking due diligence for these decisions?

Evidence that claims correspond to facts. What tests were used to validate manufacturer's claims about their products? Or what tests might be used, and why are they appropriate?

A technical report can be used to both record the work that was undertaken to make the decisions, and to summarise them for a non-technical audience.

There are many other reasons why a professional might need write a technical report. They can, for example, be used to examine questions such as: Should a project be undertaken? Why did the project fail? Should the current project be cancelled? Is it on track to success? Is the system reliable? What are the risks of using a certain software development approach? What are the strengths and limitations of the existing software?

Such reports are a key part of how many workplaces function. They provide a permanent record that work was undertaken, that is, they are a form of corporate memory. And they are a mechanism for sharing technical information amongst staff, who may not know each other or even work in the same country.

As another example, consider a report on an investigation that involved evaluation of the quality of a suite of tools. The author would need to:

- Review basic concepts.
- Explain the activity and why it is important to the organization.
- Give a critical history of relevant knowledge.
- Describe the evaluation method, and explain why that method was chosen.
- List activities that were undertaken.
- Show and explain the results.
- Critically analyze the outcomes.
- State limitations of the investigation.

As can be seen from this list, a report of this kind may look a lot like a research paper.

Grant Applications

Other kinds of professional writing also have expert technical content. For academics, one such document is a grant application. How grants are written is highly specific to the funding body; for example, some are focused on innovation, such as the major national funding bodies, while others are focused on societal outcomes, such as some of the large charitable foundations. By far the best starting point will be a previous successful application written for the same agency.

While agencies are diverse, in my experience there are some principles that are more or less universal. First, you need to focus on the kinds of outcome the agency is seeking. Second, honesty is important, and not only for its own sake; grant assessors are likely to be highly experienced, and will be alert to the kinds of ways in which some applicants inflate or distort their track records, or conceal shortcomings. They will similarly be alert to ways in which the goals of applications can be overstated, and thus it is in your interests to be realistic, not grandiose, in your aims. Third, the agency may be swamped with applications, so yours should be written so that it can have impact with a few minutes of superficial reading. Fourth, in contrast to the plain or even understated writing that is appropriate in a research paper, the claims of a grant application should be stated in a direct way that highlights the potential for impact and positive outcomes (but, as noted above, without exaggeration). The forthright tone of a grant application can seem a strong contrast to the more restrained writing style of science, but it is required if your intent to achieve the research goals is to be taken seriously. Last, a piece of personal advice: be confident that you are willing to do the work. A commitment to a project that you find unrewarding may not lead to strong outcomes, and is likely to undermine your commitment to other activities.

Non-technical Writing

Professional computer scientists have specific expert or technical knowledge in their areas of competence. They apply this knowledge to produce materials such as software and technical descriptions of software. However, this material often has to be explained in a non-technical way, that is, technically-informed writing can be for a non-technical purpose or a non-technical audience.

System requirements and user manuals are an obvious example. But there are many other circumstances in which professionals and researchers need to write for non-technical readers. For example, management may be seeking a recommendation as to whether to proceed with a particular project, and need to understand the benefits, options, costs, and risks. Expert evidence to be placed before a court has similar characteristics; it needs to be technically precise, but, often, must be written for an audience with little domain knowledge.

For example, a company might need to decide which workflow tool to purchase. The IT unit is asked to prepare a recommendation. This is likely to consider factors such as cost, reliability, support, and the vendor track record—elements that are no different from any other purchase. But technical factors are also likely be important, and need to explained: choice of server platform; functionality; obstacles to compatibility with existing systems and applications; and the method of adoption. Can it be gradually introduced, or must it be a sudden cut-over from an existing application?

And then there are pragmatic considerations, which for example may flow from the preferences or capabilities of the IT staff rather than definite technical grounds: will the code be developed under Microsoft® Windows or Linux? Will it be browser-based or stand-alone? And so on. A responsibly written recommendation should be open about the influence of such considerations, and about their implications and costs.

I've observed that these kinds of writing task are more challenging than just about any other kind of professional communication. They demand more skill than is required when writing for colleagues, who have the advantage of working in a common framework; they also require imagination, for example to create illustrative descriptions that concisely explain concepts that may be alien to the audience. Even fundamentals can be difficult to communicate. Consider, for instance, how you might describe *software architecture*, *type definitions*, or *open-source development* to an attorney. And then consider how you might explain—in writing, and without an opportunity for face-to-face conversation—that a certain online data collection process is error-prone because of how Web browsers manage state and identity.

A key to taking on such tasks is to approach them much as you would a research paper, as described in Chap. 5 and elsewhere in this book. Write early, and be willing to experiment with analogies and explanations—and to discard them if they aren't working. It may also be that you have the flexibility to use or adapt other people's text, such as (with appropriate acknowledgement!) Web pages or text books; so long as the document is not going to be published, reuse does not usually violate copyright.

As for a research paper, a little planning can make a big difference to the outcome. A common early misstep is for the author to not appreciate what is actually required, or why, or when. This leads to people spending weeks (or longer) writing a document that only needs to be a page long, and will then be discarded; or not undertaking background work that the reader was expecting; or answering entirely the wrong question. Find out what you need to achieve, who the readers will be, and why they need the document at all. Consider the length of time for which the document will be of value: will it be read once, or will all future managers need to read it? Will it be open or confidential? How widely will it be circulated?

Structuring a Report

Having established the purpose of the report, the next stage is to plan the design, just as you would a research paper. For an analysis of, say, steps to take to prevent reoccurrence of a technical failure, an appropriate sequence of sections might be: Summary; Introduction; Technical overview; Definitions; Observations; Results and discussion; and Recommendations. In addition, a report typically has most of: a title; authors; a date; a history of revisions; a list of sources; references or background reading; a list of materials used; and appendices.

As for any kind of writing, choose headings wisely, and use a logical structure at all levels, that is, sections, paragraphs, and sentences. Sections with similar content should be organized in a similar way, just as tables and figures should have consistent structure. If necessary, try to tell a story (to the extent the material allows), and use annexes or appendices for bulky supporting material that interferes with the message you are trying to convey. And use numbering for any elements that might be referred to from elsewhere.

Reports produced in a professional environment are often much richer than a single conventional document. They can include materials such as code, spreadsheets, Web links, repositories, and interactive diagrams, and might be published using any of a wide range of mechanisms. However, the fundamentals are unchanged: readers need structure, regardless of the medium in which the report appears.

The best starting point is likely to be similar reports previously produced in your organization. Innovation in reporting style in unlikely to be successful; you need to learn to comply with the norms of your workplace before you can figure out ways of improving them.

Not all innovations in reporting style are positive. A recent trend is to "report by presentation", where authors create a deck of slides that reads like an overview of an extensive study, but is in fact the entirety of the report. In this approach, there is no conventionally formatted document, and no appendices, spreadsheets, analyses, or background materials. In my view, this is like having a report on the safety of some equipment that has no written justification for the results, no calculations, and no supporting evidence. In many contexts, such a report is unprofessional.

Audience

In academic writing, the audience is reasonably well defined: readers are academics, research students, or, more rarely, coursework students. While the level of expertise of the audience may vary, the kind of writing expected in a research paper is much the same regardless of the topic of the work or the likely publication venue.

In other professional writing, the audience is much broader. The single key guideline that an author needs to follow is: ensure that your text can be read and understood by the likely readers. Some people seem to have a knack for this, and effortlessly shift tone and vocabulary as they switch between producing material for, say, school children, legal proceedings, and funding agencies. In contrast, most people need to consciously consider what the knowledge and abilities of the readers are likely to be, and to adapt their writing appropriately.

It is critical that you know your intended audience; think of specific individuals, and write for them. It is also critical to consider who will be affected by the document and what their reactions to it might be. If you are recommending a shift to an environment in which some skills will no longer be required, some people's working lives may be drastically affected; if they are also readers of the document, they will at the very least expect the recommendations to be argued for with clarity and independence.

Consider the audience's education and prior knowledge, and how much detail they will need to be given to understand the report's implications; you may also need to consider how much time they are likely to have to read your document. And, while it is easy for a technically literate professional to feel superior about their knowledge, even an unsophisticated reader can be written for as an equal. A common mistake is that the authors fail to realise that they do not share values or priorities with the intended audience, that is, there is a cultural division that hasn't been bridged. For example, a company may have no interest in investing in the technical "cutting edge" when their existing infrastructure appears to be working well, or when resources are urgently required elsewhere.

At the beginning of every document, then, the reader and writer should be in agreement. The opening can be used to state common ground, from which the author can begin to build a case for the recommendations or outcomes. That is, the purpose of the document is to bring the reader to a certain point of view, or depth of understanding, starting from a common basis. Continuing the example in the previous paragraph, what is the value of a recommendation that proceeds from the assumption that a client may be willing to dump their investment in existing systems and platforms? Without a persuasive argument from a common basis of shared values—the need to attain a certain level of reliability, say—the recommendation is worthless.

Style

The advice on style in the other chapters of this book applies just as much to professional writing in general as it does to research papers and theses. However, research papers are written within a defined culture, whereas other professional writing can take place in a wide range of contexts, a factor that can change the emphasis: some issues become more important, some less so. Probably the most significant single issue is the importance of, not only *being* professional, but *appearing* to be professional.

The way in which you write sends messages about the kind of person you are. If you are gushing, and for example claim that "NoSQL is probably the most important computational innovation since the invention of the Internet", you have strayed from the realm of the professional and into that of the salesperson or biased advocate.

There are many ways in which your writing can undermine your message, To begin with: do be professional! Take care to stay within your expertise and, when necessary, admit your ignorance. Flag opinions as opinions—don't present them as facts, and don't pretend to be objective. Likewise, don't misrepresent your accuracy or overstate your confidence, and don't falsely imply that tests took place, or that evidence was gathered.

Sometimes writing can betray an underlying attitude that is not objective. Consider the following examples of unintentional self-revelation, where the statement contradicts the impression that the writer would presumably want to convey.

Impartial and trustworthy? "Our competitor uses shoddy development practices whereas we are fully standards-compliant and have a much larger team." ("Shoddy" is a judgemental term, and team size may not even be relevant.)

Knowledgeable? "Linux is the successor to the Unix operating system used widely in universities in the 1980s and is now being adopted commercially." (Unix has been used commercially since the 1980s, and has not been superseded by Linux.)

Risk aware? (*And not a risk-taker.*) "A parallel test period is not necessary due to the simplicity of the design used." (Even a simple design can fail.)

Diligent, responsible, careful, thorough? "We looked at the availalbe [sic] options and decided to use of [sic]Mysql due to its populity [sic] and use for free [sic]."

And similarly: "We will investigate the issue when that stage of the development is reached, as we are confident that we can solve any inter-platform software migration problems." (On what basis does this confidence rest?)

Reflective? Flexible? "It is obvious that reconsidering design decisions some weeks after work has commenced is infeasible." (Bad design decisions should always be reconsidered, surely.)

Expert? "The decision to use XL as the underlying database was taken due to the extensive XL experience in the department." (Do you need much experience to use XL? Is it an appropriate choice technically?)

Independent? "Purchase of these tools will give the IT staff the opportunity to interact with other users of this technology." (And why is this a benefit?)

The following two items, quoted or closely paraphrased from real documents, are failures of good professional writing. The first is from a functional specification, and is intended for a non-technical client.

✗ Crucial user requirements as indicated in the task analysis (detail) phase include:

- A record-level mutex provision allowing long-duration transactions.
- Rollback with user authorization and high-level "diff" display for feedback.
- Blackboard-based shared workspaces with hierarchical access-control protocols.
- Arbitrary nesting of customized document elements.
- Definition of customized elements in UI mode and plain-text mode with full BNF-based error-checking on demand.
- Write-on-demand and auto-write in both generic and customized element creation environments.

It is difficult to see how a client could use this specification to give meaningful feedback to the developers; I would guess that even experienced computing professionals would struggle to understand this specification, and would need to use their imagination to see how the separate statements relate to each other.

The next example is from instructions written by a developer for users of a configurable workflow system.

✗ Modification of the interface to allow direct displaying of and modification to currently-hidden fields can be done by performing changes to the record-display screen module in the interface package. The key routine is the `advancedShowItems` routine which is used by administrators when low-level changes need to be done to patch errors in stored data. Fields displayed in the `advancedShowItems` routine are editable in a browser and changes are performed in the database when "Save Record" is done by the user.

Three changes need to be done for the customization. The fields where changes can be performed need to be fetched which involves changing the SQL statement used to fetch records by adding the required fields. Then displaying the fields need to be done using the same routine as existing fields (the `showEditableField` routine is used for this). This will break the layout table which also needs to be modified. Then the final stage is to modify the test statements so that modified fields are added to the update SQL statement.

These instructions ran to about 50 pages, all at this level of clarity. A developer who used the system described the instructions as being, in most cases, a response to easily fixed bugs and omissions in the browser-based interface, and “a good reason to never buy that product”. I agree—the writing style suggests lazy, poorly managed software development.

Indeed, as a general rule, if a reader does not understand why a system is constructed in a certain way or follows certain principles, it may be impossible to maintain it or use it. Documentation is not a substitute for good design and sensible decision-making. It describes what you *have* done, not what you *should* have done. Aim to generate the documentation that teaches the reader and leads them to share your understanding. And volumes of documentation that are too large may be unusable; only write documents that you expect will be read.

Other Problem Areas

Prominence. Often, a reader will fail to notice the main point of a document, or will miss elements such as actions they need to take or questions that the author wants them to address. A common cause is that the author, who may have written the document with great care, somehow expects the reader to spend the same effort. Instead, though, the reader may just glance through, looking at no more than headings and a few sentences.[2] While not all readers are so slapdash, some will be, and the skill of communication is to work with the behaviour of readers, not to expect them to change.

Thus it is the author’s job to make sure that key messages are conspicuous to even a casual reader. In a research paper, the standard structure guides the reader to the main lessons, but, in other documents, the author must create this prominence explicitly. Careful use of elements such as bold text, bullets, and repetition of the main statements ensures that the reader sees the material that you want the reader to see.

Jargon. Jargon was mentioned earlier in this book, but is more of an issue in general professional writing than in academic contexts because there may be a greater mismatch in expertise between author and audience.

It is all too easy for a technical description to degenerate into a stream of acronyms.

✗ The SRS module uses the MDD and QDD subsystems to manage XDL records via the CRS, with XMOP, DND, and a DNOF interface. On QVID failures the module relies on QTS2.1 and XNS.

Mathematics is a form of jargon. Use mathematical notation for mathematical concepts, but consider the audience. Translating math into text won’t help people who aren’t familiar with the ideas, or who aren’t accustomed to mathematical thinking.

[2] One of my colleagues was notorious for never reading past the first ten lines of an email—and then pleading ignorance of whatever the bulk of the email said. This included reminders to update his password (he lost access to email on several occasions) and once, catastrophically, an email from a travel agent advising of changes to flights.

Get rid of noise words and phrases such as “situation”, “perform”, “it is often the case that”, “done”, or “during the course of”. In general, choose the simplest suitable word or the most precise suitable word; and consider the reader’s vocabulary.

Citation and Acknowledgement. People who contribute to written material can reasonably expect to be acknowledged, and you should fully acknowledge any external sources whose material is used in your work. (This also includes source code.) It is wrong to take credit, implicitly or explicitly, for work done by others. That is, it is unethical to fail to cite technical papers, texts, Web pages, and other resources when their ideas or content are used. This matters even in emails! They are not as ephemeral as you might think (or hope).

The passive voice is just as unhelpful in this context as in any other. Don’t write “it was decided that the rollback module would not be proceeded with”; make clear who did what.

A “Professional Writing” Checklist

- What is the task that the document addresses?
- What is the purpose of the document? To inform or persuade?
- Is the purpose of the document clear to the reader? How prominent are the key messages?
- Is the writing appropriately professional and independent?
- What other tasks must be undertaken to complete the document? For example, does code need to be written, or systems tested?
- How will the document be consumed? Will it be read once, or form part of a permanent record? How polished does it need to be?
- How will the document be published? What form will it take? What form do such documents usually take in your organization?
- Who is the audience? What is their level of expertise? Will it be necessary to explain basics?
- Are you proceeding from shared values?
- Who will be affected by the document, and what will their reaction to it be?
- What is a reasonable amount of time to allocate to the task? Should it be worked on full-time, or amongst a mix of other tasks?
- If time is limited, how will it be spent?
- Have all participants in the activity been recognized or credited appropriately?

Chapter 13
Editing

(1) *The reader should be able to find out what the story is about.* (2) *Some inkling of the general idea should be apparent in the first five hundred words.* (3) *If the writer has decided to change the name of the protagonist from Ketcham to McTavish, Ketcham should not keep bobbing up in the last five pages.*

James Thurber's standing rules for writing of humour
What's So Funny?

If a conscientious reader finds a passage unclear, it has to be rewritten.

Karl Popper
Unended Quest

If you give me an eight-page article and I tell you to cut it to four pages, you'll howl and say it can't be done. Then you'll go home and do it, and it will be much better.

William Zinsser
On Writing Well, Sixth Edition

The writing of a paper begins with a rough draft, perhaps based on records of experiments or sketches of a couple of theorems. It will probably include material produced during the project, such as notes taken in meetings, reviews of literature, emails discussing the research question, and text from sources such as progress reports. The next phase usually consists of filling out the draft to form a contiguous whole: explaining concepts, adding background material, arranging the structure to give a logical flow of ideas. Finally, the paper is polished by correcting mistakes, improving written expression, and taking care of layout. Although it does not change the quality of the research, it is this last phase—the styling of the paper—that has the most impact on a reader. It should not be neglected, however strong the ideas being communicated.

Few writers are good at judging their own writing. Discovery of shortcomings in your text takes time and effort: careful reading, a willingness to admit to mistakes, the regret of discarding text that was hard to create, and the labour of writing it afresh. We know what we meant to say, but what we actually said may only be obvious to others. The difference between a weak writer and a strong writer is, often, not the ability to write fluently, but the effort taken to diligently edit and revise.

© Springer-Verlag London 2014

J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9_13

Consistency

Editing is the process of making a document ready for publication or examination. Much of editing consists of checking the document for errors that fall under the heading of consistency (or lack of it). Use the checklist at the end of this chapter when revising your papers, or when proofreading papers for others. A surprisingly effective editing exercise is to pretend to be a reader, a member of the paper's intended audience. This shift of framework, of consciously adopting the position of critic, often exposes problems that otherwise go unnoticed.

My experience is that early drafts tend to be repetitive and long-winded. Often, not only are concepts awkwardly expressed and sentences unwieldy,[1] but material on one theme might be in separate parts of the paper. It is common to find similar material included several times, in different places, particularly when there are several authors. Another problem is that some material may become irrelevant as the paper develops through a series of revisions.

The ordering too may need to be reconsidered once the paper is complete. When material is moved from one place to another, check that the text in each location is intelligible and appropriate in the new context. Beware, for example, of moving definitions of terms or of breaking the flow of an argument.

For many papers, editing leads to removal of text. Don't be afraid to shorten your papers: cutting will improve the quality. Edit for brevity and balance. Omit or condense any material whose content or relevance to the paper's main themes does not justify its length.

Style

Another kind of editing is for style and clarity, and is perhaps the hardest part of finishing a paper. Chapters 6–10 are concerned with points of style that should be checked during editing; these should be considered during every revision. Keep in mind the fundamental aim, which is to make the paper clear. Lapses will be forgiven so long as you are easy to understand.

When revising the text of other writers, it is often preferable to make minimal changes: correct the presentation but retain the flavour of the original text. Don't expect to impose your style on someone else.

Most journals have a preferred style for elements such as references, figure numbering, spelling, table layout, and capitalization. If you are planning to submit to a particular journal, consider using its style.

[1] Not having an absence of double negatives is a common problem in my writing.

Proofreading

There is no excuse for a report that contains spelling errors. They jump out and glare, displaying not only your inability to spell, but also your casual attitude to your work. Find a spell checker that you like and get into the habit of using it, and use a style checker too. But spell checkers won't find missing words, repeated words, or misused words, or correctly handle names. Nor will they find misspellings that form another correct word; a typical example is the substitution of "or" for "on" or "of".[2] Adopt a convenient set of symbols for hand-correcting printouts of documents; many dictionaries and style guides have good examples of notation for copy-editing.

A common error of mine is, when intending to type a word, to instead type some other word that shares a few initial letters. A related error is that of replacing words by their anagrams; I type "being" for "begin", "form" for "from", "relation" for "relative", "compute" for "complete", and so on. I also replace words by their homonyms, such as "two" for "too". Undoubtedly there are a few of these errors in this book—they are hard to find.[3]

Identify and look for your own common errors. Typical examples include incomplete sentences and sentences that have been run together inappropriately. Check for errors in tense and in number, that is, in the use of plural and singular forms. When you identify an error that you often make, add it to a checklist, and look for it whenever you revise. But put the list aside when writing—it will distract you.

Many papers are completed without ever having been printed, but most people read more carefully away from the computer than they do at a screen—if only because there are fewer distractions. It is vital to read your paper at least once in its entirety, to check flow and consistency. Set the draft aside for a day or two before proofreading it yourself, as doing so increases the likelihood of finding mistakes.[4] (Many people have an emotional attachment to their writing; the delay allows this attachment to fade.) Read each sentence carefully, and ask yourself how easy it is to understand.

It is particularly important to check the bibliography. Readers will use it to track down references, so any garbling of information can lead them astray, and other writers may be offended if you have misreferenced their papers. Format should be consistent and each reference should include enough information to allow readers to locate it.

---

[2] From newspapers: A couple who, in their wedding ceremony, "stood, faced the floral setting, and exchanged cows". Another couple, where the "groom appeared to stumble as he walked in, while the band playing 'Hear Comes The Bridle' ". Miners, who "went ahead with their strike ballet". A poet, who released "a slimy volume" entitled New Poems.

[3] When the final draft of the first edition was being checked, a reader noticed that this sentence said: "Undoubtedly there are few of these errors in this book—they are hard to find" .

[4] Newspapers, with their short deadlines, inevitably overlook some mistakes. The following is the complete text of a newspaper article (as quoted in The New Yorker).

> The Soviet Union has welded a massive naval force "far beyond the needs of defence of the Soviet sea frontiers," and is beefing up its armada with a powerful new nuclear-powered aircraft carrier and two giant battle cruisers, the authorative "Jane's Fighting Ships" reported Thursday.

Always get someone else to read your work before you submit it or distribute it. You may have misunderstood a relevant article, or made a logical error; most authors are poor at detecting ambiguity in their own text; and there may be relevant results of which you are unaware; explanations may be too concise for the uninitiated; a proof that is obvious to you may be obscure to others. And a proofreader's comments should never be ignored. If something has been misunderstood, the paper needs to be changed, although not necessarily in the way the proofreader recommends.

Publication-quality word-processing is so widely used that poorly presented reports look cheap. But word-processors can glitch on the final draft. The last word of a section might be the first word on a page, a line of text might be isolated between two tables, or a formula might be broken across two pages. This is also the last chance to correct bad line breaks. Some editing may be required to fix such errors, to move or change the offending text or to relocate a table. In desperate cases, such as a long piece of displayed mathematics that is broken, consider putting the offending material into a figure.

Fussy people like me clean up widows and orphans. If the last line of a paragraph contains only a single, short word, that line is a cub; use an unbreakable space to join the short word onto the previous one. When the last line prior to a heading is by itself at the top of a page, or a heading or the first line of the following paragraph are alone at the bottom of a page, that line is a widow or orphan; rewrite until it goes away. I also clean up stacks, that is, paragraphs where successive lines begin with the same word.

Choice of Word-Processor

When you start to write a paper you need to choose a word-processor. The choice is dictated by availability, but also by how well the available word-processors cope with the demands of authoring. In addition to text, much research writing involves figures,

---

> (Footnote 4 continued)
> "The Soviet navy at the start of the 1980s is truly a formidable force," said the usually-truly is a unique formidable is too smoothy as the usually are lenience on truly a formidable Thursday's naives is frames analysis of the world's annual reference work, said the first frames of the worlds' navies in its 1980–1981 edition.
> "The Soviet navy at the start usually-repair-led Capt. John Moore, a retired British Royal News Services.
> "The Soviet navy as the navy of the struggle started," she reportable Thursday.
> "The Soviet navy at the start of the 1980s is truly a formidable force," said beef carry on the adults of defence block identical analysis 1980s is truly formidable force, said the usually-reliable of the 1980s is unusually reliable, lake his off the world's reported Thursday.

The following is from a paper in a conference proceedings where the authors provided camera-ready copy.

> Not only is the algorithm fast on the small set, but the results show that it can even be faster for the large set. (This can't be right, run the experiment again?)

tables, mathematics, use of multiple fonts and sizes, and cross-references to figures, tables, equations, sections, and bibliographic entries. Most authors of technical papers find at one stage or another that they must contend with the limitations of word-processing software.

Further problems are presented by the lifecycle of technical papers. For example, a paper might initially be drafted for circulation amongst colleagues, revised for submission to a conference, then accepted after further revision and experiments; but, because the paper is too long, some text must be omitted. Subsequently, after rethinking, new work, and reintroduction of omitted text, the paper is combined with a report on earlier work and submitted to a journal, where, after revision to meet referees' comments, it is accepted, perhaps as long as three years after the initial draft was written. Word-processors need to be able to handle this high level of revision and re-organization.

There are, broadly speaking, two kinds of word-processor, the visual or WYSIWYG style typified by Microsoft® Word, and the compiler style typified by LaTeX, which compile marked-up text into a page description language such as PostScript. The visual word-processors are generally superior at production of documents for immediate use such as letters and Web pages, and for first drafts, but for technical writing the compiler word-processors are preferable. The compiler word-processors have features such as transparent methods for commenting-out text, making omission and re-inclusion straightforward, and macro facilities that make it easy to generate multiple distinct documents (such as a conference version and a more complete technical report) from one source file. Documents produced with visual word-processors can look amateurish, particularly if mathematics is involved.

The LaTeX word-processing system was used for this book, and is today arguably the best word-processor for technical writing. The first edition was written under Unix; the second edition was written under both Unix and Windows; the third was written under Linux, Windows, and Mac OSX. There are many circumstances in which I choose to use a visual word-processor, but technical writing is not among them.

An "Editing" Checklist

- Are all of the components present: title, authors, abstract, and so on?
- Are the acknowledgements complete and accurate?
- Is the ordering of material correct?
- Are the titles and headings consistent with the content?
- Have all terms been defined?
- Is the style of definition consistent? For example, were all new terms introduced in italics, or only some?
- Has terminology been used consistently?
- Are defined objects always described in the same way? For example, if you use the expression "all regular elements E" when introducing a concept, but the

shorter form “all elements E” in later references, is “regular” implicit in the latter expression?

- Are abbreviations and acronyms stated in full when first used? Are any abbreviations or acronyms introduced more than once? Are the full statements subsequently used unnecessarily?
- Are any abbreviations used less than, say, four times? If not, can they be removed?
- Do all headings have maximum or minimum capitalization? Has a term been capitalized in one place and not in another?
- Is the style and wording of headings and captions consistent?
- Are names always used in the same way? Has a consistent convention been used for the formation of new names?
- Is spelling consistent? What about “-ise” versus “-ize”, “dispatch” versus “despatch”, or “disc” versus “disk”?
- Is tense used correctly? Are references discussed in a consistent way?
- Have bold and italic been used logically?
- Are any words hyphenated in some places but not others?
- Have units been used logically? If milliseconds have been used for some measurements and microseconds for others, is there a logical reason for doing so? Is the reason clear to the reader? Has “megabyte” been written as “Mb” in some places and “Mbyte” in others?
- Are all values of the same type presented with the same precision?
- Are the graphs all the same size? Are the axis units always given? If, say, the x-axes on different graphs measure the same units, do the axes have the same label?
- Are all tables in the same format? Does the use of double and single lines follow a logical pattern? Are units given for every value? Are labels and headings named consistently? If, say, columns have been used for properties A to E in one table, have rows been used elsewhere? That is, do all tables have the same orientation?
- Does every figure and table have a caption?
- Has the same style been used for all algorithms and programs? Is there a consistent scheme for naming of variables? Do all pseudocode statements have the same syntax? Is the use of indentation consistent?
- In the references, has each field been formatted consistently? Have italics and quotes been used appropriately for titles? Is capitalization consistent? Are journal and conference names abbreviated in the same way? Is the style of author names consistent? Has the same core set of fields been provided for each reference of the same type?
- Is formatting consistent? Has the same indentation been used for all displays? Are some displays centred and others indented? Do some sections begin with an unindented paragraph and others not?
- Do the parentheses match?

Chapter 14
Experimentation

The senses deceive from time to time, and it is prudent never to trust wholly those who have deceived us even once.
Rene Descartes

A hypothesis is… a mere trial idea, a tentative suggestion concerning the nature of things. Until it has been tested, it should not be confused with a law… Plausibility is not a substitute for evidence, however great may be the emotional wish to believe.
E. Bright Wilson, Jr.
An Introduction to Scientific Research

Even the clearest and most perfect circumstantial evidence is likely to be at fault, after all, and therefore ought to be received with great caution.
Mark Twain
Pudd'nhead Wilson's Calendar

The use of experiments to verify hypotheses is one of the central elements of science. In computing, experiments—most commonly an implementation tried against test data—are used for purposes such as confirming hypotheses about algorithms and systems. An experiment can verify, for example, that a system can complete a specified task, and can do so with reasonable use of resources. A tested hypothesis becomes part of scientific knowledge if it is sufficiently well described and constructed, and if it is convincingly demonstrated.

Some people disagree with the view that rigorous experiments are essential in computer science; or, if they do not explicitly disagree, may hold a low opinion of papers that have no new theory and are "merely" experimental. Yet such views are in stark contrast to the role of experiments in other disciplines. Experiments are an essential part of sound science.

Experiments in computing take diverse forms, from tests of algorithm performance to human factors analysis. However, the principles underlying good experimentation are much the same regardless of what is being investigated. Tests should

© Springer-Verlag London 2014

J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9_14

be fair rather than constructed to support the hypothesis. If the design of tests seems biased towards the intended contribution, readers will not be persuaded by the results.

The topic of this chapter is the design, execution, and description of experiments in computing. As elsewhere in this book, to some extent the material here draws on my experience as a researcher. These examples are for the most part work that led to successful outcomes—which is not to imply that all of my research has succeeded to this extent.

Baselines

A first step in the design of experiments is to identify the benchmarks against which your contribution will be measured. That is, it is essential to identify an appropriate *baseline*. For example, no sensible researcher would advocate that their new sorting algorithm was a breakthrough on the basis that it is faster than bubblesort; instead, the algorithm should be compared to the best previous method.

A benchmark is only compelling if it is implemented to a high standard, and thus it may be that comparison to a baseline is difficult because an implementation for a competing method must be obtained. However, without such a comparison it may be impossible for the reader to know whether the new method offers an improvement. This is a *barrier to entry*: before you can begin to produce competitive work in an area, it is necessary to not only become familiar with the methods and ideas described in a body of literature but also to have access to a collection of appropriate tools and resources. But the fact that there is a barrier to entry does not excuse poor science.

A danger in an ongoing research program is to fail to update the choice of baseline. In the context of text indexing, for example, in work in the 1980s on signature files performance was compared to that of inverted files as reported in papers from the 1970s. (One of these 1970s papers gave a figure for inverted file size of 50 %–300 % of the indexed data, though skeptical considerations strongly suggest that the larger figure is implausible.) Papers on signature files even in the 2000s continued to quote these baselines, despite dramatic improvements in inverted files (and well-known experiments reporting sizes such as 7 %–10 %). New work in signature files was compared to previous work in the same area, but not to relevant work on other pertinent technology.

A similar problem can arise when a well-known, widely available implementation becomes commonly used as a reference point. When the worth of every new contribution is shown by comparison to the same baseline system (or, in some cases, baseline data set), in some respects the field benefits, because the use of the common resource means that readers can have confidence that the baseline is accurate. However, in other respects the field may suffer, because the advances that are being described may not be cumulative.

Some new algorithms solve a novel problem, or solve an existing problem in a novel way that is for some reason not comparable to previous work. There may still be a clear baseline to compare to, however. For example, there may be an obvious algorithm that an intelligent, informed person might use if asked to solve the problem. That is, one potential point of comparison is the first workable option that a reasonable person might suggest.

It is critical that baselines be identified early in the research program. For example, what is the point of developing new methods if existing methods—or, perhaps worse, trivial methods—provide a satisfactory solution? In work with Web data, for example, we found that problems we were experiencing with parsing of the text might in principle be resolved by automatically determining which (European) language each page was written in. To our surprise, a trivial method based on counting occurrences of a small number of representative words (such as "the" for English or "der" for German) gave 100 % accuracy on our test data. Plans to investigate richer techniques had to be abandoned.

Persuasive Data

For work that involves experiments, it is critical that you have access to appropriate data, and that you understand it well.[1] In general terms, you need to consider:

- What data may be available, and whether it is created by you or sourced from elsewhere.
- What specific mechanisms will be used to gather and standardize the data.
- Whether the data will be sufficient in volume or quality to give a robust answer to the question.
- What domain knowledge may be required to properly interpret the data.
- What the limits, biases, flaws, and properties of the data are likely to be, and how these problems will be addressed or managed.
- What the results will be like if the data supports the hypothesis; or, alternatively, what they will be like if the hypothesis is false.

To understand these requirements, put yourself in the position of the reader. You want to be persuaded that an algorithm is the very best option available, for a certain class of problem. A test on inadequate data, or on data that has any of a range of uncertainties, will leave you doubting the claims.

Consider a detailed example. A *microarray* is a technology that can be used to cheaply obtain the genetic profile of a human, by identifying, for each of say a million common genetic variations, whether the variation is present or absent. With a large collection of individuals that have a microarray profile, and other *phenome* data on the individuals, such as health status or physical characteristics, it is possible

[1] In this discussion I generally use *data* in the usual sense in computing, namely as the raw material on which experiments operate. In other contexts, *data* is the result or output of an experiment, such as measurements gathered in a lab or from human subjects. Confusingly, computing experiments on data produce data as output. It is the output sense of the word data that is meant in the truism "we process data to obtain information, analyze information to obtain knowledge, and comprehend knowledge to obtain wisdom".

to analyze the profiles to see if there is a clear link from specific genetic variations to specific aspects of the phenome, for example to try and identify variations that are linked to occurrences of a particular cancer.

In this example, microarray-based linkage analysis is an application for which a researcher is developing computational methods. Because the linking process is uncertain (the data is unreliable and incomplete, genetics is imperfectly understood, and so on), there is scope for a new method that improves on existing approaches. Validation of this method will require data, and simulated or artificial data is unlikely to be persuasive, because the accuracy of simulation would depend on complex assumptions about factors such as laboratory conditions, biases in the sample of humans chosen for profiling, microarray error rates, and distributions of individual genetic variations.

Considering the list of desirable characteristics given above, then, the researcher might give the following responses.

What data: The researcher could profile some individuals directly, but it is much cheaper to obtain a collection of profiles from a public genomic database. Such profiles will often be associated with previous publications, so their characteristics should be well understood.

What mechanisms: To choose specific data sets, a good approach in this case could be to find research papers that draw biomedical conclusions based on particular data of the right kind, and then obtain that data. The data may then need to be normalized or cleaned up in some defensible way: if the data was originally gathered for other purposes, some of it may not be suitable for the current investigation; or it may be in an inconsistent format; or it may contain known outliers that could reasonably be removed by hand; or, if derived from multiple, inconsistent sources, it may need to be unified by separate preprocessing of each of the components.

Sufficiency of data: There are several respects in data volume is relevant. One is algorithmic: methods tend to behave differently at different scales. Performance (in terms of processing time) for a data set that fits in CPU cache is unlikely to be informative for data that requires hard disk or networked access, for example. Some methods simply don't scale, as unanticipated costs become dominant. A slightly more subtle challenge of scale is that larger data sets offer different statistical properties. Finding a matching image from amongst a hundred hand-chosen candidates may be much easier than from amongst a million that were chosen at random; while the smoother distributions of a large data set may, for example, simplify the problem of finding similar documents—or, as in this case, detecting genetic linkages.

Another respect in which volume is relevant is statistic significance; data volumes need to be large enough to ensure that the experiment will be able to detect the effect that is being hypothesised.

Sufficiency also has another dimension—the number of data sets being used. A single data set may not be persuasive, particularly if the reader suspects that the method was tuned to perform well on the data set reported in the paper.

Domain knowledge: A failure in some computer science research is that it is inspired by a real-world problem, such as genetic linkage, but the problem has been abstracted or simplified in some way that makes it unrealistic, and thus the methods that are developed have no practical relevance. For example, there would be little value to a linkage algorithm that assumed that microarray data was free of error.

A similar failure is reporting of unvalidated results, such as a researcher reporting that linkages have been found, but which are biologically implausible. There is a detailed understanding of biological roles of the components of the human genome, and this domain knowledge should be part of the interpretation of the results.

Limits: Microarrays are noisy; values may be incorrect or uncertain, and the amount of noise in biological experiments can vary from one laboratory to another. Likewise, the human processes of gathering phonemic data are also uncertain. There are inherent biases, such as the tendency of microarray data sets to consist of samples from wealthy countries, and from people who are known to have genetically linked disease, with only limited numbers of controls (healthy cases). The incidence of a particular condition, such as a rare cancer, may be very low, and some machine learning techniques are poor with highly imbalanced samples. Knowledge of the extent of such confounds in the data is needed to help assess the significance of the results, and to then, as far as possible, rectify the data. This may involve, for example, careful manual data processing, following explicit guidelines. It is essential that such manipulation doesn't in some way bias the results towards the method that is being explored—thus avoiding situations that can be parodied as "our method is poor in the presence of outliers and inconsistency, so we removed the problematic data".

Results: With a good understanding of the data that is to be used, the researcher should be able to make some predictions about the results, or about their form. On an assumption that the method works, the researcher could perhaps estimate the likelihood that a particular level of statistical significance is observed for a particular strength of linkage; or could estimate the size of the smallest data set in which the linkage would be detected.

Some experiments depend on human annotation of data, to provide a *gold standard* or *ground truth*. In document classification, for example, human annotation may indicate the topic of a document: politics, entertainment, sport, and so on. For microarray data, this annotation is available in a natural way, as the characteristics of each profiled human are part of the data set. In other cases, gathering of annotations can be the dominant cost of an experiment—a factor that is often overlooked, or underestimated, by inexperienced researchers.

In the process of developing new algorithms, researchers typically use as a testbed a data set with which they are familiar. If the algorithm is parameterized in some way—by buffer size, say—this testbed can be used for tuning, that is, to identify the parameter values that give the best performance. What this tuning process almost certainly cannot do is identify the best parameters for all data sets, or even identify whether there are stable best parameters to choose.

It is for this reason that descriptions of the research cycle distinguish between an observation phase (used to learn about the object under study) and a testing or confirmation phase (used to validate hypotheses). If parameters have been derived by tuning, the only way to establish their validity is to see if they give good behaviour on other data. Choosing parameters to suit data, or choosing data to suit parameters, in all likelihood invalidates the research.

The research in some fields is underpinned by the availability and use of reference data sets. Such resources can be dramatically larger and more comprehensive than the materials that could be created by a typical research team, are easy to explain to readers, and, in principle, allow the direct comparison of work between institutions and between papers. In some instances, it can be difficult to publish work unless a reference data set has been used. However, use of such data also carries risks, in particular of overfitting; that is, methods can become so specialized that they do not work on other data.

When considering what experiments to try, identify the data or input for which the hypothesis is least likely to hold. These are the interesting cases: if they are not tested—if only the cases where the hypothesis is most likely to hold are tested—then the experiments won't prove much at all. The experiment should of course be a test of the hypothesis; you need to verify that what you are testing is what you intended to test, and an experiment should only succeed if the hypothesis is correct.

An underlying point, then, is that persuasive research requires appropriate data, and thus you need to be confident that you can obtain good data before committing to a particular research question. (In some fields, it may be that the research goal is to obtain data: telescopes and particle accelerators are built to collect data, for example. But, in computing, such research is extremely rare.) It follows that pursuit of some questions, no matter how interesting they may be, will not feasible for some researchers.

Ask whether a single data set is sufficient, or whether multiple data sets are required: for separate training and testing, or for independent confirmation. A related question is whether multiple data sets are indeed sufficiently independent; subsamples of a single large data set may, for practical purposes, be the same, and not yield the truly independent confirmation that is being sought.

Sometimes appropriate data can be artificial, or simulated; as noted in Chap. 4, such data can allow a thorough exploration of the properties of an algorithm. But such data should not be used without a clear understanding of its limitations. For example, application of a new hash function to random data is unlikely to be a convincing demonstration that the function is uniform, since the data was uniform to begin with. Fundamentally, any scheme for generating artificial data relies on a model, which embodies assumptions and, probably, simplifications. The strongest defence of artificial data is to validate it against real data.

A related question is estimation of the volume of data required. Another way of phrasing this same issue is: to what volumes of data should your claims apply? If you are making claims about terabytes (say), but testing on megabytes, you are asking the reader to believe that your results can be extrapolated a million-fold. Yet, for some problems, merely doubling the volume of data can introduce new challenges.

Such issues arise in testing of techniques such as document search methods. They also arise in challenging ways in the context of algorithms for analysis of DNA, that is, the strings representing genomes. These algorithms have to contend with several different forms of scale. One form is the length of the strings, which for a single organism can vary from a few thousand characters (viruses) or millions (bacteria) to billions (a vertebrate such a human) or over a hundred billion (some plants). Another form is the complexity of the genome; some contain a great deal of internal redundancy or copying. Yet another form is the number of organisms being simultaneously analyzed. A further, more subtle form of scale is the evolutionary distance between the individual organisms—here, the dimension is of timescale or diversity, rather than raw data volume. Each of these forms of scale has significant, non-linear impact on behaviour of commonly used bioinformatics algorithms.

The question of data volume arises in the formal statistical sense of power: whether your data is of sufficient quantity, or quality, to allow observation of the effect that you are seeking. For example, if you are comparing two parsers according to their ability to accurately extract phrases from English text, it may be that statistical principles will tell you that a collection of 100 examples is unlikely to be sufficient for the anticipated improvement to be detected.

Interpretation

When checking experimental design or outcomes, consider whether there are other possible interpretations of the results; and, if so, design further tests to eliminate these possibilities. Consider for instance the problem of finding whether a file stored on disk contains a given string. One algorithm directly scans the file; another algorithm, which has been found to give faster response, scans a compressed form of the file. Further tests would be needed to identify whether the speed gain was because the second algorithm used fewer machine cycles or because the compressed file was fetched more quickly from disk.

Care is particularly needed when checking the outcome of negative or failed experiments. A reader of the statement "we have shown that it is not possible to make further improvement" may wonder whether what has actually been shown is that the author is not competent to make further improvement. Moreover, the failure of an experiment typically leads to it being redesigned—such failure is as likely to expose problems in the tests as in the hypothesis itself. Design of experiments to demonstrate the failure of the hypothesis is particularly challenging.

It is always worth considering whether the results obtained are sensible. For example, are there rules of conservation that should apply to the experiment? This issue was illustrated by one of my students, who was evaluating a classification method in which documents were automatically allocated to one of several predetermined categories. She reported numbers of true and false positives and negatives, as a function of a sensitivity threshold; however, these numbers were clearly wrong, as they had no relationship to the total number of documents. As another example, in some cases, boundary conditions are highly predictable—do the results appear to be right as they approach the boundaries? In the same classification work, as the threshold was raised, the number of positives (true or false) should have fallen to near zero; instead, the number rose.[2]

There are many such lines of questioning that a thorough researcher might consider. For example, in a typical case it should be possible to make a rough guess as to the expected experimental outcome—is this observed? What happens when *seeding* is tried, that is, additional data items with known (and possibly extreme) properties are added to a data set—does the right outcome occur? And so on.

Conclusions should be sufficiently supported by the results. Success in a special case does not prove success in general, so be aware of factors in the test that may make it special. As discussed earlier, a common problem is scale—whether the same result would be observed with a larger data set, for example.

Don't draw undue conclusions or inferences. If, say, one method is faster than another on a large data set, and they are of the same speed on a medium data set, that does not imply that the second is faster on a small data set; it only implies that different costs dominate at different scales. Also, don't overstate your conclusions. For example, if a new algorithm is somewhat worse than an existing one, it is wrong to describe them as equivalent. A reader might infer that they are equivalent if the difference is small, but it is not honest for you to make that claim.

Another aspect of interpretation is that numerical measures allow numerical manipulation, but such manipulation does not always make sense if applied to the qualitative goal we wish to achieve—the goal may not behave in a numerical way. One system may achieve a 20 % higher score than another under some measure of user satisfaction, but it makes little sense to say that the user is 20 % more satisfied. A more formal way of expressing this caution is to say that many measures in computer science are on an ordinal, rather than an interval or ratio, scale: that is, we can say that one score is higher than another, but cannot directly interpret the degree of difference between scores.

In some fields, it is a common, but poor, practice to use not one measure, but several, each of which is intended to capture the same essential quality of a system. For instance, the effectiveness of a search engine might be reported using all of average precision, discounted cumulative gain, reciprocal rank, and precision in the top five, ten, and thirty documents retrieved. Such a practice is obnoxious when the author picks through the measures to find significant ones (average precision for one

---

[2] The existence of these issues was obscured by chaotic reporting practices. For example, in one iteration of the work she reported total numbers of positives and total numbers of errors—both positive and negative—but did not report any of the components, such as true positives, false negatives, and so on. In another iteration, instead of reporting numbers as a function of threshold, she reported the number of positives as a function of the number of false positives, so that the threshold acted as a hidden variable. In some fundamental way she had not grasped how a trend can be used to understand the underlying behaviour of the method being investigated; in this case, it would be interesting to observe change in the number of true positives as the threshold was varied.

collection, perhaps, and normalized discounted cumulative gain for another), and rests on two false assumptions: first, that the measures are independent of each other, that is, are assessing distinct qualities; and second, that they are of equal value.

A key concept here is of *predictivity*. The main reason that we experiment and measure is to provide evidence about the behaviour of a system in general—not just on some specific data set. That is, we use measurements on the data we have to hand to make predictive claims about what will happen in the future, when the same system is applied to new data; the conclusions in our papers are usually about properties of systems, not their behaviour on the data we have already seen.

Some measures are more predictive than others, however. To take a concocted example, we could measure a system for translating text between languages by comparing automatic translations to human translations, and counting how many words the automatic and human translations have in common. Alternatively we could rate the automatic translation by how many characters it has—the closer it is in length to the human translation, the better. The method based on words in common should be reasonably predictive: if system A is 30 % better than system B on 1,000 sample translations, then we could reasonably expect A to have better commonality-based scores than B on the next 1,000 translations. But suppose that B was better than A according to the length-based measure on the samples. In all likelihood, we would not expect this to predict length-based performance on the new translations; indeed, commonality probably predicts length better than length predicts length.

Where two different measures do assess distinct qualities, however, it is good practice to report both, particularly where the measures and their underlying qualities are in tension. To give a simple example, in classification both of false positives and true positives are informative, and methods that reduce the first (good) also tend to reduce the second (not so good).

Robustness

Experiments should as far as possible be independent of the accuracy of measurements or quality of the implementation. Ideally an experiment should be designed to yield a result that is unambiguously either true or false; where this is not possible, another form of confirmation is to demonstrate a trend or pattern of behaviour.

A simple example is the behaviour of query evaluation on a database system with and without indexes. For a small database, the most efficient solution is exhaustive search, because use of an index involves access to auxiliary structures and does not greatly reduce the cost of accessing the data. As database size grows, the cost of data access grows linearly, while index access costs may be more or less fixed. Thus the hypothesis "indexes reduce search costs in large databases" can be confirmed by experiments measuring search costs with and without indexes over a range of database sizes. The trend—that the advantage given by indexes increases with database size—is independent of the machine and data. The exact size at which indexes become beneficial will vary, but this value is not being studied; it is the trend that is being studied.

Success or otherwise should as far as possible be obvious, not subject to interpretation. If one team develops a piece of software more quickly than does another, is that because (as a researcher might have hypothesized) the first team used a new software development methodology? Or is it because the first team is larger; or smaller; or happier; or more competitive; or more experienced in the problem, the language, or coding in general; or lucky with regard to bugs and design decisions; or not doing the work while a major sporting event is on; or some other thing?

Another example of this principle is provided by the various improvements that can be made to the standard quicksort algorithm, such as better choice of pivot values and use of loops that avoid expensive procedure calls. With test data chosen to exercise the various cases—such as initially unsorted, initially sorted, or many repetitions of some values—experiments can show that the improvements do indeed lead to faster sorting. What such experiments cannot show is that quicksort is inherently better than, say, mergesort. While it might, for example, be possible to deduce that the same kinds of improvement do not yield benefits for mergesort, nothing can be deduced about the relative merits of the algorithms because the relative quality of the implementations is unknown, and because the data has not been selected to examine trends such as asymptotic performance.

For speed experiments based on a series of runs, the published results will be either minimum, average, median, or maximum times. Maximum times can include anomalies, such as a run during which a greedy process (a tape dump, for example) shuts out other processes. Minimums can be underestimates, for example when the time slice allocated to a process does not include any clock ticks. But nor are averages always appropriate—outlying points may be the result of system dependencies. Statistical considerations are discussed later.

Results may include some anomalies or peculiarities. These should be explained or at least discussed. Don't discard anomalies unless you are certain they are irrelevant; they may represent problems you haven't considered.

✓ As the graph shows, the algorithm was much slower on two of the data sets. We are still investigating this behaviour.

It is likewise valuable to explore behaviour at limits and to explain trends.

A common failing in experimental work is that complex processes are tested as a whole, but not as components. Many proposed methods are pipelines or composites of one kind or another, in which independent elements are combined to give a result. For example, a search engine might consist of a crawler, for fetching pages; a parser, for extracting content; and a query engine, for assessing similarity. The design of each element has impact on the final results. If a researcher proposed a new engine comprised of entirely new components, but only tested it as a whole, the reader would not learn to what extent each component was valuable; it might be that all of the benefit came from just one of them. If a series of decisions have led to the final form of the contribution, or the contribution is composed of separate elements or stages, each should be assessed independently.

Similarly, an experimental regime should include separate investigation of each relevant variable—the reader needs to know what factors are influencing the outcomes. And if increasing a value has an effect, what happens with a further increase? That is, you should explore factors broadly enough to make trends and patterns clear.

Consider the difficulties in measuring how the performance of a distributed algorithm changes as the number of servers is increased. Suppose the volume of data is fixed. As the number of servers is increased, it may be that at some point the data on each machine can be held wholly in memory (because the per-machine volume of data is falling), so that disk is not required. The experiment might then show a dramatic increase in performance; but this would in fact be due to the reduced problem size, not to the power of distribution. On the other hand, if the data volume is scaled with number of machines, then the characteristics of the data set as a whole may be altered. In some cases there is no straightforward resolution to such issues.

At the start of Chap. 4, I introduced an example of a research program that was designed to investigate the impact of cache on the relative performance of two data structures. This example illustrates some of the difficulties in producing a robust, persuasive experimental design.

First, my proposal was that performance be assessed by counting cache misses (an operation that may be supported in hardware, or may involve use of a simulator). This approach produces evidence that is clearly related to the hypothesis, which concerns cache behaviour, but does not directly demonstrate an improvement in computational efficiency.[3] Second, the relative behaviour will depend on the data volumes. For example, it may be that, as the amount of data is increased, the costs for the tree-based algorithm grow slowly until some thresholds are exceeded, then grow dramatically as just a little more data is introduced, and then grow slowly again. Third, behaviour with growth might be obvious for some data sets, and less conspicious for others, depending on the pattern of accesses.

To produce strong experimental designs for this problem, researchers needs to use their understanding (that is, models, formal or otherwise) of the algorithms to anticipate whether issues of robustness will arise, and then to choose measures and data that have the potential to expose inconsistencies between the models and the behaviour is practice.

Performance of Algorithms

As an example of experimental design, consider potential approaches to assessment of the performance of algorithms. The tools, as discussed in Chap. 4, are formal proof, mathematical modelling, simulation, and experimentation. How these are used flows

[3] Is computational efficiency even well defined? Is it the number of instructions used, say, or the number of seconds each CPU core spends on the process? These are no longer equivalent. If the core is otherwise idle, due to memory access delays, is processing time relevant at all? The answers to these issues will depend on the research question.

from questions about what the assessment of the algorithm is intended to achieve, and what the likely limitations of and contraints on the experiment are, as follows.

Basis of evaluation. The basis of evaluation should be made explicit. Where algorithms are being compared, specify not only the environment but also the criteria used for comparison. For example, are the algorithms being compared for functionality or speed? Is speed to be examined asymptotically or for typical data? Is the data real or synthetic? When describing the performance of a new technique, it is helpful to compare it to a well-known standard.

A comparison should have a realistic basis. In particular, the basis should not appear to favour the algorithm being demonstrated over existing algorithms. If the basis of comparison is questionable, the results are questionable too.

Simplifying assumptions can be used to make mathematical analysis tractable, but can give unrealistic models. Non-trivial simplifications should be carefully justified.

Processing time. Time (or speed) over some given input is one of the principal resources used by algorithms; others are memory, disk space, and disk and network traffic. Time is not always easy to measure, since it depends on factors such as CPU speed, cache sizes, system load, and hardware dependencies such as prefetch strategy. Times based on a mathematical model rather than on experiment should be clearly indicated as such.

Measurements of CPU time can be unreliable. CPU times in most systems are counted as multiples of some fixed fraction of a second, say a sixty-fourth or a thousandth. Each of these fractions of time is allocated to a process, often by heuristics such as simply choosing the process that is active at that moment. Thus the reported CPU time for a process may be no more than a good estimate, particularly if the system is busy.

Memory and disk requirements. It is often possible to trade memory requirements against time, not only by choice of algorithm but also by changing the way disk is used and memory is accessed. You should take care to specify how your algorithms use memory.

Disk and network traffic. Disk costs have two components, the time to fetch the first bit of requested data (seek and latency time) and the time required to transmit the requested data (at a transfer rate). Thus sequential accesses and random accesses have greatly different costs. For current hardware, in which there are several levels of caching and buffering between disk and user process, it may also be appropriate to consider repeat accesses, in which case there is some likelihood that the access cost will be low. The behaviour of network traffic is similar—the cost of transmitting the first byte is greater than the cost for subsequent bytes, for example.

Because of the sophistication of current disk drives and the complexity of their interaction with CPU and operating system, exact mathematical descriptions of algorithm behaviour are unattainable; broad approximations are often the only manageable way of describing disk performance.

Applicability. Algorithms can be compared not only with regard to their resource requirements, but with regard to functionality. The basis of such comparisons will be quite different to those based on, say, asymptotic analysis.

A common error is to compare the resource requirements of two algorithms that perform subtly different tasks. For example, the various approximate string matching algorithms do not yield the same results—strings that are alike according to one algorithm can be dissimilar according to another. Comparing the costs of these algorithms may not be particularly informative.

Human Studies

A variable in many studies is the user. Humans need to be involved to resolve many kinds of research question: whether the compressed image is of satisfactory quality, whether the list of responses from the search engine is useful, whether a programming language feature is of value. Humans can be used to assess outputs—did the robot do a good job of the housework? Is this Web page suitable for children? And humans can be the subject of experiments—what are the main shortcomings of this user interface? Which of these technologies is most helpful for navigating an unfamiliar city?

Appropriate use of humans in experiments allows many forms of rich measurement that provide insights into the value of computational methods. These can be qualitative, such as assessment of ease of use; and subjective, such as self-reported feedback on new technologies. They can also be quantified, through independent observation of behaviours and responses.

Design of human studies is treated in detail in research methods texts written for information systems, psychology, or business methods, and is beyond the scope of this book. However, researchers should consider whether humans are needed for their work, because of the depth and value that a well-designed human intervention can add to measurement of an experiment. Some of the questions that should be considered include:

- Are human assessors or human subjects needed? To what extent will the results be persuasive if humans are *not* used?
- How many humans will be needed, and who will they be? How will their eligibility, typicality, and relevant competence be determined?
- What instructions will they be given, and how will the experimenter avoid communicating to the subjects what the desired outcome is? That is, how will subject bias be avoided?
- Across large, repetitive tasks, such as annotation of a collection of items, how will consistency be ensured?
- How carefully will the experiment have to be planned and prepared for? Should it be run iteratively, with trials used to establish problems that are addressed in later versions?
- Does the experiment need to be blind, or double-blind?

- In experiments in which humans use a simulated system, will they know whether they are interacting with a human or a computer?
- Does it need to be a substantial, controlled experiment, or are case studies sufficient? With the former, quantifiable measures are gathered over a large number of individual tests; with the latter, the goal is to undertake a qualitative analysis of a few individuals.

As an example of this last point, a student of mine developed a tool for identifying and prioritizing changes between versions of legal documents. He tested his system quantitatively, by creating thirty or so pairs of documents (each pair consisting of an original and a revision of it), and observing how many of 15 readers could find the changes in each pair, either with the new tool or with an existing approach. He also tested it qualitatively, by interviewing three pairs of authors who used it for some weeks as they collaboratively authored three documents. This work produced convincing results, with the two approaches independently confirming each other.[4]

Far too many human studies in computer science are amateurish and invalid. Instructions to the experimental subjects should be clear; the sample of human subjects should be representative (a class of computer science students may not be typical of users of mobile devices); the subjects should be unaware of which of the competing methods under review was proposed by the researcher; anonymity should be preserved; and controls—analogous to placebos in medical trials—should be in place. The ethical guidelines for human studies at most universities are far-reaching, and in all likelihood any investigation involving people evaluating a system needs ethics clearance.

There is no doubt, however, that human studies are an essential element of computer science research. Without evaluation of the impact on users, for example, it is difficult to see how to draw strong conclusions concerning a user interface, a search mechanism, a machine translation system, a software engineering methodology, a video compression technique, or any of a vast range of contributions. However, human studies of some questions continue to be a rarity; research in a range of areas is flawed by lack of measurement of the human element. The fact that such studies would be expensive is a poor reason for avoiding them; such reasons would not be acceptable in medicine or psychology.

One of the longest-running experiments in computer science is the TREC evaluation of information retrieval systems at the U.S. National Institute of Standards and Technology, which has a significant human-factors component. Each year, participants—a large number of research groups from around the world—apply their retrieval systems to standard, shared materials with unknown attributes. This side of the experiment is *blind*; the researchers do not know which materials have which attributes. The output of the systems is then manually evaluated by human assessors, who label the materials to assign attributes to individual items. This side of the experiment is also blind: the outputs are merged prior to inspection and the assessors do not know which system has done what. Another aspect of the TREC work is that the

[4] Sadly, the findings were that the system was unhelpful.

use of standardized resources means that there is direct control of the principal variables, and experiments are comparable between research groups; existing published results provide a baseline against which new results can be directly compared.

By the standards of computer science, the TREC experiment is expensive, with, for example, some months of assessor time required every year. However, TREC illustrates that robust experiments can have high impact. When TREC began (in 1992, a year or two before the Web began to be significant), there was a large range of competing theories about the best way to match documents to queries. Weak methods were rapidly culled by TREC, and a great many dramatic improvements in information retrieval were spurred by the opportunity that TREC created. The Web search engines drew substantial inspiration from the TREC work and, in contrast to some other areas of computer science, the links between academia and industry remain strong. This impact could not have been achieved without the large-scale involvement of human assessors, or without the commitment to robust experimentation.

Coding for Experimentation

In computer science research, in principle at least, the sole reason for coding is to build tools and probes for generating, observing, or measuring phenomena. Thus the choice of what to measure guides the process of coding and implementation—or, perhaps, indicates what does not have to be coded.

The basic rule is to keep things simple. If efficiency is not being measured, for example, don't waste time squeezing cycles from code. If a database join algorithm is being measured, it may not necessary to implement indexes, and it is almost certainly unnecessary to write an SQL interpreter. All too often, computer scientists get distracted from the main task of producing research tools, and instead, for example, develop complete systems.

In coding for an experiment, there are several other such rules or guidelines that might seem obvious, but which are often not followed. Examples include:

- One task, one tool: decompose the problem into separate pieces of code. In most cases, trying to create a single piece of code that does everything is just not productive. Do you need to integrate the data classifier into the network generator, and the network generator into the visualizer? Wouldn't it have been easier to develop them independently and combine them with a script?
- Be aware that you may need to trade ease of implementation against realism of the result. Can load balancing across distributed machines on a network be examined without development of significant software infrastructure? Can an algorithm be assessed if all data is held in memory, or is it necessary, for realism, to manage data on disk, perhaps in a custom-built file system? Hard-coding of data structures, input formats, and so on, may allow for rapid implementation; does it lead to unrealistic behaviour or simplifications?
- Cut the right corners. Coding for a day to save an hour's manual work is a waste of time, even if coding is the more principled approach. But coding for an hour

followed by execution for a month is a lot less efficient than coding for a day followed by execution for 20 min—especially if you have to run the experiment again.
- Use the right tool, not the most convenient tool. For example, if you plan to measure algorithmic efficiency, implement in a suitable language.
- Don't re-code unnecessarily. Use libraries; is it really necessary to do a fresh implementation of a B-tree?
- Find an independent way of verifying that the output is correct. If you think you've successfully compressed some data, prove it: write a working decompressor.
- For long-running processes, consider developing mechanisms for periodically saving state, so that weeks (or more) of work isn't lost.

In environments such as the Unix family of operating systems, a program is often tested by being run from the command-line, with output directed to the screen. Parameters may be passed in as arguments, but to simplify coding they may be defined as constants within programs. All too often, though, a researcher discovers that an experiment run in this way cannot be repeated a day or two later.

A more reliable, repeatable approach is to run all experiments from scripts. Parameter settings are captured within the script; the settings used last time can be commented out. Output from the script can be directed to a logfile and kept indefinitely. If the output is well designed, it should include information such as input file names, code versions, parameter values, and date and time.

Using simple Unix tools, it is straightforward to take data directly from a log file and produce a graph or other summary of the results. These steps too can be encoded in a script; the process for completing any stages undertaken by hand may well be forgotten if the work is rested for a few months, such as while a paper is under review. A corollary is that the output of your code should be amenable to scripting, with, for example, consideration given to consistent use of fields in each line of output from experiments.

A practical consideration is whether the experiments are feasible at all. Experiments can require storage of large volumes of data; implementation of production-quality code; execution over months, with repetitions after failure; access to particular machines or configurations; use of humans for evaluation of results; access to restricted data sets; use of particular pieces of software; or most of these things at the same time. Before proceeding too far with a research question, you need to be confident that you will have the resources required to undertake the experiments that are needed for a persuasive outcome.

Describing Experiments

Your interpretation and understanding of the results can be as important as the results themselves. When describing the outcomes of an experiment, don't just compile dry lists of figures or a sequence of graphs. Analyze the results and explain their significance, select typical results and explain why they are typical, theorize about anomalies, show why the results confirm or disprove the hypothesis, and make the results interesting. That is, motivate the work.

Experiments are only valuable if they are carefully described. The description should reflect the care taken—it should be clear to the reader that possible problems were considered and addressed, and that the experiments do indeed provide confirmation (or otherwise) of the hypothesis. A key principle is that the experiment should be verifiable and reproducible. Results are valueless if they are some kind of singleton event: repetition of the experiment should yield the same outcomes. And results are equally valueless if they cannot be repeated by other researchers. The description, of both hypothesis and experiment, should be in sufficient detail to allow some form of replication by others. The alternative is a result that cannot be trusted.

Researchers must decide which results to report. As discussed earlier, researchers should have logs of experiments recording their history, including design decisions and false trails as well as the results, but such logs usually contain much material of no interest to others. And some results are anomalous—the product of experimental error or freak event—and thus not relevant. But reported results should be a fair reflection of the experiment's outcomes.

If a test fails on some data sets and succeeds on others, it is unethical to conceal the failures, and the existence of failure should be stated as prominently as that of success. Likewise, reporting just one success might lead the reader to wonder whether it was no more than a fluke.

Not all experiments are directly relevant to the hypothesis. An experiment might be used, for example, to make a preliminary choice between possible approaches to a problem; other experiments might be inconclusive or lead to a dead end. It may nonetheless be interesting to the reader to know that these experiments were undertaken—to know why a certain approach was chosen, for example. For such experiments, if the detail is unlikely to be interesting it is usually sufficient to briefly sketch the experiment and the outcome.

The experimental outcomes reported in a paper may represent only a fraction of the work that was undertaken in a research program. There will have been exploratory stages and different kinds of failures, and the reported runs may well be carefully chosen to represent a broad range of experiments. Thus the published record of the work is highly selective.

In other disciplines of science, researchers are expected to keep detailed notebooks recording ideas, methods, experiments, data, participants, outcomes, and so on. These notebooks fill a variety of roles, in particular providing a complete history of the research, allowing the experiments to be reproduced, and proving that the published work took place as described—in labs in the biological sciences, for example, it may be required that a senior scientist sign and date each page.

Notebooks have not acquired these roles in computer science. However, as discussed in Chap. 5, they are invaluable. They can be used to record versions and locations of software, parameters used in a particular experiment, data used as input (or the filenames of the data), logs of output (or the filenames of the logs), interpretations of results, minutes of decisions and agreed actions, and so on. They allow easy reconstruction of old research, and simplify the process of write-up. They are particularly helpful if a paper is accepted after a long reviewing process and experiments have to be freshly run; all too often the code no longer produces anything like the original results, because too many details of the experiments have been lost and the code has been modified. Most of all, notebooks keep researchers honest.

While there are obvious reasons to consider maintaining notebooks online, in my experience written notebooks continue to be as effective, and they help provide a physical sense of progress and achievement that is somehow lacking in an online equivalent. In either form, it is good discipline to include dates, never change an entry, and use the notebook as often as possible.

Another strategy that keeps researchers honest, and helps to describe and publicize their work, is to make code and data available online. Doing so shows that you have high confidence in the correctness of your claims. In an informal survey I undertook in the 1990s, several computer scientists commented to me that they would not have made some of the claims in their papers if they had had to publish their code or to run their experiments under external scrutiny. More positively, publishing code reduces the barrier to entry for other researchers, and helps to establish baselines against which new work should be measured.

Part of reporting of experiments is description of the data that was used. Typically, readers need to know how the data was gathered or created; how your version of the data might be obtained, or recreated; what the shortcomings of the data are, that is in what ways it might be uncertain, incomplete, or unreliable; and what aspects of the research question are not tested by the data. Observe that raw data and massive listings of intermediate outcomes are not in this list!

An "Experimentation" Checklist

Regarding *the design of the experiments*,

- Have appropriate baselines been identified? What makes them appropriate? Are they state-of-the-art?
- What data has to be gathered, and where from?
- How will readers gather comparable data for themselves?
- Is the data real? Is it sufficient in volume? What validation is required for artificial data?
- Should the data be seeded with examples to test the validity of the outcomes?
- Is there reference data for the problem, and what are its limitations?
- Will a domain expert be needed to interpret the results?
- What are the likely limitations on the results?
- Should the experimental results correspond to predictions made by a model?
- Will the reported results be comprehensive or a selection? Will the selection be representative?

- What enduring properties might be observed by other people attempting to validate the work with different hardware, data, and implementation?
- Are the experiments feasible? Do you have the resources (time, machines, data, code, humans) required to undertake them to a reasonable standard?

Regarding *any software to be developed*,

- Can baselines be obtained from elsewhere, or do they need to be implemented? Will they be of similar standard to the implementation of your system?
- How much coding is required? What existing resources can be used? That is, is your coding effort being used effectively?
- Can the code (or the proposed contribution) be decomposed into components? How will the individual components be tested for correctness, and evaluated for significance?
- How will you know that the code is correct?
- Is the code going to be made publicly available? If not, why not?

Regarding *the use of human subjects*,

- In what ways might humans be needed? To annotate data? As test subjects? Can your question be meaningfully answered without human subjects?
- Who will the humans be and how will they be selected?
- How many humans will be required, and how does that correspond to your budget? What compromises will be introduced if fewer humans are used, or for less time?
- How will objectivity and independence be maintained? What steps need to be taken to avoid introduction of bias?
- Is ethics clearance required?

Chapter 15
Statistical Principles

Statistical thinking will one day be as necessary for efficient citizenship as the ability to read and write.
Samuel S. Wilks

Statistics are no substitute for judgement.
Henry Clay

We use experiments and take observations to study the behaviour of a system, to test hypotheses, to investigate the effect of manipulations and optimizations, and, overall, to produce evidence for our arguments. The elementary material of evidence is measurement: the reduction of complex phenonema to numerical scores that can be recorded, compared, and analyzed.

Raw numbers, however, are dangerously deceptive in their apparent certainty. If we find in an experiment that our system has a higher score than that of a competitor, we are easily convinced that our system is superior. But, first, as discussed in Chap. 4, the measure itself may be inaccurate or misleading; it may be only an approximation to the real-world quality that we are attempting to measure. Second, even if the measure is appropriate, the value it provides in a single test may be subject to variability and randomness—in the choice of experimental inputs, in the conditions in which the experiment is run, or in human assessment of the outcomes.

To gain trust in experiments, we need to repeat them, giving sets of results to which we can apply statistical methods. Having multiple experiment results not only provides aggregated, stable measurements, but lets us use the tools of statistical inference to determine how confident we should be in our conclusions. Repeated experiments also provide insights into system behaviour through allowing us to observe variability, success, and failure in a systematic way. An introduction to statistical methods for experimentation is beyond the scope of this book, but all researchers should be aware of relevant statistical principles, and be able to judge when use of statistics is necessary for their work.

© Springer-Verlag London 2014

J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9_15

Variables

The ideal experiment examines the effect of one variable on the behaviour of an object being studied. How does increasing the volume of data affect execution time? Can the vision system track rapidly moving objects? How much compression can be achieved without visibly degrading the image? If no other variables are present, it is easy to be confident that the variable does indeed affect the behaviour in the way observed. The test environment should be designed to minimize the effect of extraneous factors—that is, to unambiguously relate variations in one property to variations in another.[1]

In practice, elimination of variables is remarkably difficult. Even elementary properties can be surprisingly difficult to measure: for example, access time to material stored on disk is not just a property of disk hardware, but is affected by access pattern, presence and size of disk cache, and file system design. Tests should be designed to yield results that are independent of properties such as system characteristics or constant-factor overheads that are not part of the hypothesis.

Consider the measurement of performance of two compression techniques. If tested on different data, the results will be incomparable: we have no way of knowing whether the better performance is due to use of a better method, or due to choice of data that is inherently more compressible. Thus one particular component of a test environment is choice of test data. For some experiments standard data is available, such as benchmark problems in machine learning or the corpora used to test compression methods. The use of such standard resources is essential to experimentation on these problems (although, as noted in Chap. 14, such resources also have potential limitations). Where standard data is not available, care should be taken to ensure that the chosen test data is representative.

A fundamental issue is that you should have a clear understanding of the relevant parameters. In hashing, table size is a tuneable parameter that directly affects aspects of performance (collisions and cache behaviour, for example); expected key length and distribution of key values is another. Some parameters are dependent on others,

---

[1] In careful research published in 1648, Jan-Baptista van Helmont concluded that plants consist of water:

> That all plants immediately and substantially stem from the element water alone I have learnt from the following experiment. I took an earthen vessel in which I placed two hundred pounds of earth dried in an oven, and watered with rain water. I planted in it the stem of a willow tree weighing five pounds. Five years later it had developed into a tree weighing one hundred and sixty-nine pounds and about three ounces. Nothing but rain (and distilled water) had been added. The large vessel was placed in earth and covered by an iron lid with a tin-surface that was pierced with many holes (to allow the soil to breathe while preventing dust from adding to it –JZ). I have not weighed the leaves that came off in the four autumn seasons. Finally I dried the earth in the vessel again and found the same two hundred pounds of it diminished by about two ounces. Hence one hundred and sixty-four pounds of wood, bark and roots had come up from water alone.

.

but not always in obvious ways. In a well-designed experiment each parameter will be explored independently.

Another component of many test environments is the hardware, particularly for efficiency research. A good option is to describe performance in terms of the characteristics of some commonly available hardware, as for example specified by clock speed, disk access time, and so on. This allows readers to relate published results to observed performance on another system.

Samples and Populations

Taking a measure of an event—the running time of a program, the proportion of entities correctly identified in a text, the effectiveness of a classifier after a certain number of training examples—gives us a numerical score quantifying that event. The numerical score has a sense of exactness to it; but this exactness is often misleading. Were we to run the experiment again, a different score might be produced, due to factors such as change in the task, change in the input to the same task, variability in the experimental setup that is beyond our control, or variability in the measurement apparatus itself.

The variability may arise naturally in the system or environment; or it may be due to variation in the input, which could be controlled (by varying a parameter, say) or unpredictable (due to data being derived from different sources). In either case, the existing of variability demonstrates that a single test is not enough, and that the variability should be reported and analyzed along with the overall outcomes.

In computer science research, many people view statistics as no more than reporting of averages and deviations. However, the role of statistics in experimental research is a rich one, and seeking to answer elementary statistical questions can illuminate experimental design. Your research may well benefit from a statistical approach.

An approach to understanding these issues is to explore the meaning of simple statements about the behaviour of a system. When we state, for example, that "algorithm NEW is typically faster than algorithm OLD", it is reasonable to suppose that the intention is to claim that NEW is faster on average. (Such a statement could as easily be made on the basis of a theoretical analysis as on the basis of experiments.) But an average of what? If the intended meaning is only that NEW is faster than OLD on average for the runs undertaken in the experiments, what is it about these runs that makes them representative or predictive?

A key concept here is of *population*: the set of all possible runs. If NEW is indeed faster than OLD on average across the whole population, the claim is a reasonable one, but in all likelihood the population is infinite, as it must contain all possible combinations of input data—if the volume of input can be arbitrarily large, even so simple a property as the typical size of an input is ill-defined.

As taking the average across such a population is likely to be impossible, it is necessary to resort to taking a *sample*; to do so, it is necessary to understand what the population consists of. In medicine, to evaluate the benefit of a new antibiotic, the population could be all people, or perhaps all sick people, or possibly all people for whom other medications have failed. Designing an experiment includes deciding what the possible, or reasonable, inputs are, and how they might be varied.

A straightforward case of varying inputs is where this is due to randomization. Consider a classifier that achieves a certain effectiveness score on a collection of 100,000 items, with 10,000 of the items randomly selected for training and 90,000 for the test; obviously (I hope it is obvious), the effectiveness will differ for different random partitionings of the items into training and test sets. The correct approach is to perform multiple randomizations, to yield a distribution of outcomes that can be analyzed statistically. In other cases, explicit randomization is not available. For example, we may be constrained to certain real-world data sets. Any activity involving human judgement will often give different results for different humans—or even for the same human at different times.

Another form of variability in input is where the task itself is qualitatively changed. For instance, a search engine will achieve the same effectiveness on a given query and collection of Web pages each time it is run, but will achieve different scores for different queries or different collections—which may be due to the new data changing the nature of the task. In cases where the tasks change qualitatively, it may not be meaningful to talk about a population, sample, or average, so other strategies for interpretation of results may be appropriate.

Aggregation and Variability

A simple way to aggregate the multiple scores achieved by running an experiment multiple times is to report an average of the outcomes. Typically, the average is reported as the (arithmetic) mean, though in some circumstances—where the results are highly skewed, for instance, or there appear to be outliers—the median, or some sort of trimmed mean, may be preferable. The average score is more stable and representative of the experiment than any single score: it is a better predictor of what score would be achieved if the experiment were run again, and it provides a more stable basis for comparison of results between different experimental setups.

Consider how to identify the likely worst case of a particular class of string-hashing functions—that is, to find out how many strings might conceivably hash to the same slot in practice. We faced this problem in work on string hashing, where analysis of the functions would be suspect. (The distribution of characters in words is not uniform, and the distribution varies according to character position within words. Thus an analysis would involve assumptions that could easily be confounded in practice.) Theory tells us that the worst case of all strings hashing to the same slot is ridiculously improbable for an ideal function; the specific question is to identify how close to ideal a randomly chosen member of a class of hash functions is.

In evaluating the properties of string-hashing functions, there are several variables in the population of inputs: the hash-table size, the number of input strings, the strings themselves, and a hash function chosen from the class. In the particular class we were considering, hash functions are determined by seed value, so the class is finite for an efficient implementation in a language such as C, with say

2^{32} or 2^{64} members. A constraint was that there were theoretical predictions only for some table sizes and load factors. It would be possible to explore other table sizes and input sizes, to seek a wide picture of behaviour, but intuitively it seemed unlikely that different observations would be made. The hash functions were chosen by randomly generating 32-bit seeds. The strings were chosen randomly from about a dozen sources: text, programs, DNA fragments, binary files, and exhaustive sets of strings of some given fixed length.

These strings are clearly not "typical", even assuming we know what "typical" means in such cases; there are many possible sources of strings, and who can say which is most likely to be hashed. Had the behaviour of the functions varied significantly between the different sources of string, it would have been difficult to draw any meaningful conclusion; however, the behaviour for every set was virtually identical, and moreover was indistinguishable from ideal. (Note, too, that this is an example of a multi-variable experiment. The behaviour under each variable can be evaluated by holding the others constant.)

As another example of the pitfalls of sampling and populations, consider natural-language processing, with say the goal of evaluating the accuracy with which a parser can identify nouns. The result depends on the input: perhaps tweets, or text derived by optical-character recognition from printed material, or randomly chosen Web pages, or articles from newspapers. Evaluation of typical accuracy depends on assumptions about the population, and on the sampling process. A truly random sample, if sufficiently large, should have in miniature all of the characteristics of the population it represents.

Given that experiments should be based on explicit assumptions about underlying populations and samples, some interesting consequences follow. Consider a thought experiment: the simple task of measuring the average height of the students at a particular university. Choose a sample of students at random, measure their height, and take the average. It might be found that all the students in the sample, excepting one or two outliers, are between 150 and 200 cm, with an average of 172. Now choose one student at random. It should be obvious that the likelihood that this student's height is average, say 172 ± 1 cm, is low.

We thus conclude that a randomly chosen individual is likely to be atypical! By the same reasoning, conclusions based on a single input, outcome, or event may well be meaningless.

The importance of randomness in the selection or generation of inputs to tests is something that is often missed by novice researchers. As counter-intuitive as it may appear, it is better to pick examples randomly than to try systematically to select "representative" examples, or—worse still—to let some other arbitrary factor (such as date of creation, order in a file, or alphabetical sorting of identifier) make the selection. The extent to which a collection of random samples is likely to be typical of the population from which they are drawn can be estimated statistically from the samples themselves; in contrast, the possible effects of a deterministic process are unknown.

Whether an average is a reasonable estimate of typical behaviour depends on the kind of event being measured. It makes no sense to report average running time when input size varies, for example. For evaluating the accuracy of noun detection of a sample of documents, average may well be appropriate; but for evaluating typical network delay for a round-trip of a packet, average may well be meaningless. First, some delays are effectively infinite (the packet is lost). Second, the distribution of such delays often consists of a large number of fast responses and a small number of extremely slow responses; the average is therefore somewhat slower than the fast times, but in a range where no values were observed at all. An analogy is averaging the duration of a plane flight and of a car journey from Paris to Moscow, and stating that this middle value is a typical travel time—although it would never be observed in practice.

A consequence of this reasoning is that there are cases where the maximum or the minimum may be the best value to report. For example, the time taken for a distributed system to process a problem may vary a little depending on a range of variables, all of which have the effect of interfering with the system. The minimum time represents the most pure run, in which the system has had the least additional work. Thus it may be appropriate to report the fastest time observed, while noting the variance.

Reporting of Variability

Averaging provides valuable insight into typical behaviour, but it is often also appropriate to report variability. It is particularly important for determining statistical significance, as described later; but even a simple analysis of variability can deepen analysis of results.

In a paper where the authors examined how efficiently a particular distributed system (of 64 remote servers connected with a novel virtual topology) could process relational operations such as joins and sorts, they followed an appropriate methodology in which the size of the tables being used as input was varied,[2] and reported on how the elapsed time varied with problem size.

However, they dutifully reported "surprising" results such as that the method was faster for 4,000,000 records than for 3,000,000 records, and elaborately speculated as to whether the topology somehow led to less contention for certain ordinal sizes of problem. It turned out that they had reported averages over a few runs for each size—and that in some cases the average included wild outliers, perhaps due to other traffic on the network, where one run had been 10 or even 100 times slower than the others. Their incomplete reporting had meant that they had significantly misinterpreted the results.

[2] The pattern of size variation was not well chosen, though. As is a common practice, they increased the size of the tables linearly, in this case from 1,000,000 records to 10,000,000 records, in increments of 1,000,000. However, they used this result to make claims about scaling—although only one (decimal) order of magnitude was present. The result would have been more impressive if they had increased the size geometrically, from say 10,000 records to 100,000,000 records, by a factor of 10 or $\sqrt{10}$ at each step. A logarithmic graph of size versus of time would have clearly demonstrated a trend.

A commonly reported descriptive statistic is standard deviation. The benefits of the standard deviation are that it quantifies variability in a single value, which are in the same units as the mean, and that it is a key input to statistical inference (as discussed later). It also has a special meaning for certain distributions, in particular the normal distribution, where mean and standard deviation fully specify the distribution's shape. The variance is sometimes reported instead of the standard deviation, the latter being the square root of the former, but variance is generally less easy to interpret.

An alternative is to report quantiles, such as the 25th to 75th percentiles, or (by analogy with a 95 % confidence interval) the 2.5th to 97.5th percentiles. Quantile ranges are most naturally combined with the median, itself the 50th percentile, rather than with the mean. If one is intending to report quantiles, note that an odd number of experimental runs is preferable to an even number, since the middle run will be the median; similarly, for a set of 21 experimental runs, the 6th is the 25th percentile, and the 16th is the 75th percentile. The more extreme the percentile, the greater the number of runs necessary to attain stable results. In some circumstances, a fuller description of the distribution of scores may need to be reported, in a form such as a graph, histogram, or box-and-whisker plot.

Average scores should only be reported with a precision that corresponds to the accuracy of the average. If only a few instances of a highly variable phenomenon are observed, then reporting many decimal places gives a false impression of exactness. For instance, if five runs of an experiment give timings of 1.143, 0.918, 1.398, 1.535, and 1.049 s, then to say that "the average running time of our algorithm is 1.161 s" makes the result seem much more precise than it really is. In this example, the standard error (the standard deviation divided by the square root of the number of observations) is 0.116, so the average should be stated as 1.2 s. If you want to provide greater precision, you need to run more experiments.

As noted above, variability in inputs or environment is distinct from variability in tasks. An example of the former is to randomly select training examples for a classifier, or randomly generated relational tables for a distributed system; an example of the latter is the existence of different sources of English-language sentences for a parser. In both cases, an average score can be calculated across experimental instances, though with different meanings. In the former case, of variability of inputs or environment, the concept of an average score is meaningful. We somehow believe that there is such a thing (on a fixed dataset and hardware) as an average running time for our program, even though each individual running time varies. It is much less clear that there is such an entity as an average English sentence, not only because there are so many ways in which text is created, but also because the universe of possible sentences is poorly defined. Averaging of scores across tasks can still be useful, for instance for comparing the performance of two algorithms; but we should be reluctant to claim that the averaged score represents anything beyond the confines of the particular experiment.

Statistical Tools

Statistical tools that have wide application in computer science research include correlation, regression, and hypothesis testing. Measures of correlation are used to determine whether two variables depend on each other. Regression is used to identify the relationship between two variables. These can be used, for example, to determine whether input size affects speed or whether light intensity affects object recognition.

Given the variability inherit in the experimental output, how do we know that the results we observe are due to some real effect, and not just to chance? Understanding this core question is fundamental to understanding not only which statistical tests to use, but also how to design experiments, and what conclusions can be drawn from them.

The principle concepts of statistical inference can be seen through a simple example (which I present here in some detail, because these concepts are often misunderstood). Consider the experiment of trying to determine whether a coin is biased; that is, whether the coin has a probability of coming up heads that is other than 50 %. Suppose the coin is flipped 12 times, and on 9 times heads are observed. Taken naïvely, the results of our experiment might suggest that coin is biased: three-quarters of the flips have turned up heads. But even if the coin is unbiased, on any given sequence of flips the proportion of heads may diverge from 50 %; any sequence of coin flips is possible.

The question we have to ask instead is, if a coin is unbiased, how likely are we to observe 9 heads or more from 12 flips? If this likelihood is sufficiently small, then we can with confidence—though not with certainty—conclude that the coin is biased.

There are $2^{12} = 4096$ distinct sequences of toins cosses. The number that have 12 heads is 1; that have 11 heads is 12; that have 10 heads is $12 \times 11/2 = 66$; and that have 9 heads is $(12 \times 11 \times 10)/(3 \times 2) = 220$. So there is a total of $220 + 66 + 12 + 1 = 299$ ways of getting at least 9 heads. If the coin is unbiased, then any given sequence of flips, such as HHTHTHHTTHTH, is as likely as any other sequence, even TTTTTTTTTTTT. Therefore the probability of flipping 9 or more heads with 12 flips of an unbiased coin is $299/4096 = 7.3\,\%$. A common experimental protocol is to set a threshold of 5 % probability or less before we are confident in a conclusion;[3] the probability here is slightly too high to confidently reject the possibility that the coin is unbiased towards heads.

This example illustrates most of the important concepts behind statistical *hypothesis testing*. The supposition that "the result was by chance", is represented by our null hypothesis—that the coin is truly unbiased. The result we are testing is stated as the alternative hypothesis—that the coin has a positive bias. We then calculate the likelihood of the observed or a more extreme result, of 9 or more heads, on the assumption that the null hypothesis is true. This is known as a one-tailed test. For

[3] The question of whether and when this protocol is correct or appropriate is beyond the scope of this book. The use of thresholds and particular statistical tests is a continuing topic of scientific debate, and methodologies continue to develop. What is clear is that some use of hypothesis testing is clearly preferable to simple reporting of averages and claimed "improvements".

the generally preferred two-tailed test, we would have to calculate the likelihood of 9 or more tosses of any one side, heads or tails; a more demanding test, because the probability is twice that of heads alone.

The calculated probability is known as the p-value of our test. If this p-value is below some specified significance level, often denoted α, then we achieve statistical significance, reject the null hypothesis, and accept the alternative: that the coin is biased. (Specifying that $\alpha = 0.05$ is a common, but not particularly strict, threshold.[4]) If the α threshold is not achieved, we cannot reject the null hypothesis, the result is not significant, and our experiment is inconclusive.

The coin-flipping example also illustrates another important principle, which is that the larger the sample size, the easier it is to find significance. Observing that three-quarters or more of the tosses are heads has a p-value of 7.3 % for 12 tosses, 3.8 % for 16 tosses, and 1.1 % for 24 tosses. This relates to the statistical concept of *power*: to reliably observe a slight, but real, effect (say, that our coin comes up heads 55 % of the time) requires far more trials than is required to observe a more pronounced effect, such as coming up heads 95 % of the time. If we have an estimate of the size of the effect we are seeking, power calculations allow us to estimate the number of trials required to observe it.[5]

Hypothesis tests are used, then, to investigate whether improvements are significant. It is often the case that, in a series of comparisons of two techniques for the same task, one is better than the other some but not all of the time. In statistical terms, in such a case the researcher needs to determine whether the two sets of results—two samples—are drawn from the same population.

We may have experimentally determined, for example, that NEW is faster than OLD "on average". That is, perhaps NEW was faster than OLD on balance when measured over a variety of inputs, or was faster in the majority of runs on the same input. In many experiments, execution times can vary substantially from one run to the next, for all the reasons discussed earlier—the layout of a file on disk, for example, could be different each time it is constructed, due to operating system variables.

Whatever the cause of the variability, this experiment is based on two sets of times, one for NEW and one for OLD. But suppose that we have a large population of running times for NEW alone, and we draw two samples from this population. It is unlikely that the two samples will have identical averages. Either we conclude that NEW is faster than itself, or that NEW and OLD might in fact not be meaningfully different. This problem is particularly acute when in some cases OLD is faster (by, say, only a small margin) and in other cases NEW is faster (by a large margin).

The issue can be resolved with a hypothesis test that compares the distribution of observations. Consider the figures in Fig. 15.1. Both of the graphs show a pair

[4] There are many instances when much smaller α is appropriate. Determining which of (say) 1,000,000 genetic variations is significantly linked to a particular property (such as susceptibility to a certain disease) might require $\alpha < 10^{-10}$, or smaller. There is an extensive literature on estimation of α in different contexts.

[5] It is astonishing how many papers report work in which a slight effect is investigated with a small number of trials. Given that such investigations would generally fail even if the hypothesis was correct, it seems likely that many interesting research questions are unnecessarily discarded.

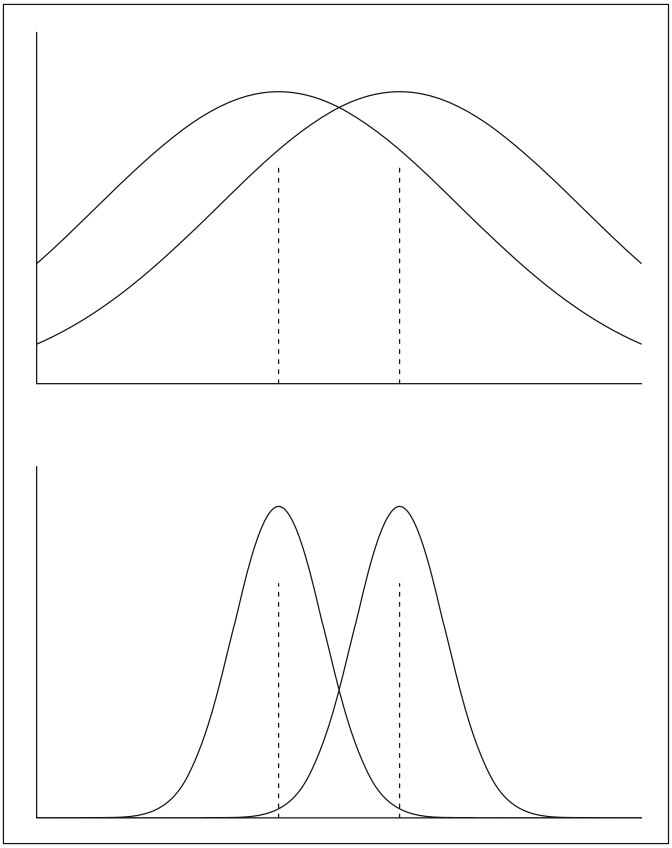


Fig. 15.1 Hypothesis tests. In the *upper* figure, the means are different, but there is a reasonable likelihood that the samples are drawn from the same population, as the distributions have high overlap. In the *lower* figure, the means are different, and the distributions are well-separated; a hypothesis test should identify that these samples are drawn from different populations

of normal distributions in which the means are different. In the upper graph, the distributions cover much of the same area; most of the points under one are under the other. Intuitively, it seems quite possible that a single underlying population is involved, and that the differences are due to the randomness of sampling choosing slightly larger instances in one case than in the other. In the lower graph, the distributions barely overlap at all. For the same underlying population to be involved, the sampling process would have had to be highly biased, choosing first a series of small values and then a series of high values. It seems improbable that this could happen by chance, so we conclude that the samples are in all likelihood drawn from separate populations.

There is a variety of hypothesis or significance tests. Choice of test depends on factors such as whether experimental outcomes are binary, or if there are scores are available; whether multiple tests are run on the same data, or if different tests were run on different sets of data; or whether the aim is evaluate the behaviour of one system, or compare two.

Some researchers are deterred from considering statistics because of the mathematical analysis that may be involved. However, first, there are packages that do much of the hard work. Second, many statistical problems can be couched in terms of elementary probability and then resolved computationally. For example, consider the problem of identifying the likelihood that a particular tennis player will win a match, given that this player has an independent probability of 60 % of winning each point. The rules are: a game is won when either player has at least four points and a two-point advantage; a set is won when either player has at least six games and a two-game advantage; a match is won when either player has a two-set advantage, or by the winner of the fifth set. Determining the probability of a win is non-trivial for the statistically innocent.

However, it is easy to write a program that runs a series of matches with a simple random-number generator for determining who wins each point; over a few thousand matches the average converges on a reliable value. A more computationally sensible option is to run a large series of trials to determine the likelihood of winning a game, then use this value as input to determination of likelihood of winning a set, and so on. It is not difficult to check that such code provides reasonable answers and that the values do indeed converge.[6] Many statistics can be estimated in this way; it was used, for example, to confirm the outcomes of the string-hashing experiments and to confirm the theoretical predictions for an ideal hash function, where random numbers and 32-bit pieces of pi were used in place of hash values.

Randomness and Error

In some circumstances it is possible for an experiment to succeed, or at least appear to succeed, by luck; there might be an atypical pattern to the data, or variations in system response might favour one run over another. Where such variations are possible, many runs should be made, to reduce the probability of accidental success and (in the statistical sense) to give confidence in the results. This is particularly true for timings, which can be affected by other users, system overheads, inability of most operating systems to accurately allocate clock cycles to processes, and caching effects.

For example, consider the apparently simple experiment of measuring how fast a block can be accessed from a file stored on disk. Under a typical operating system, the

---

[6] A typical guess of the likelihood of the better player winning the match is 90 or 95 %; in fact, the likelihood is close to certainty.

When I once suggested to students that they test the code by running it with a probability of 50 % of winning each point, several argued strongly that the program wouldn't terminate—which is more or less equivalent to arguing that, when tossing coins, you can't get some given number of heads in a row. They had confused the short-term variability (any number of consecutive throws of a head will come up eventually) with long-term averages. Such are the pitfalls of intuition.

first access is slow, because locating the first block of a file requires that header blocks be fetched first; but subsequent accesses to the same block are fast, because in all likelihood it will be cached in memory. Some deviousness is required to ensure that averages over a series of runs are realistic. Now consider a more typical experiment: real time taken to evaluate queries to a database system. If the queries are poorly chosen, the times will vary because of the caching and the complex ways in which the system components interact, and multiple runs will not give realistic figures.

Now consider another elementary experiment—comparing the speed of two algorithms for the same task. The implementations (NEW and OLD) to be used for the experiments take the same input and produce the same output, and thus are externally indistinguishable. The algorithms are run in turn on the same data, and it is observed that NEW is faster than OLD by several percent.

It would be easy to conclude that NEW is the better algorithm. However, on the evidence so far, it would also be possible to conclude that:

- NEW is better implemented than was OLD. After all, NEW is your invention, and it is reasonable to take the care that is necessary to ensure that it runs well. Perhaps the same care was not taken with OLD.
- OLD uses more buffer space than NEW, leading to poor behaviour on this particular computer. The same results might not be observed on another machine.
- OLD uses floating-point operations that are not supported in hardware.
- At compile-time, OLD was accidentally built with debug options enabled, slowing it down.
- Inaccuracies in the timing mechanism randomly favoured NEW. Although, for example, Unix-style timing mechanisms can return values in nanoseconds, their accuracy below tenths of a second is often questionable.
- OLD was run first, and was delayed while the input was copied to memory; NEW accessed the input directly from cache.
- The particular input chosen happened to favour NEW.

Further experiments are required to distinguish which of these conclusions is most likely to be correct.

Some such effects are random, some are systematic—the same wrong measurement is recorded every time the experiment is run. In work on text indexes some years ago, we had some deeply puzzling results. The first stage went exactly as predicted: we built an index for a small text collection (250 megabytes, in an era when the capacity of a typical disk drive was a gigabyte or so) and tested a heuristic improvement to the query evaluation algorithm. The test showed that the new method was about twice as fast as the old. But how would it scale? My research assistant then built an index for a gigabyte of data, and ran the same queries. The same result was observed, with the new method about twice as fast as the old; but the queries on a gigabyte used only 65 % of the time needed for a quarter gigabyte.

Considering the detail of the experiment, this result made no sense at all. Two quantities were independent of scale—the number of documents fetched and the total number of disk accesses—but the index size scaled linearly with data volume, and other measurements showed that four times as much index was being processed; yet only two-thirds of the time was required. The explanation, it developed, was that the smaller collection was kept with its index on a single disk drive while, for the larger collection, the index had been placed on a separate drive due to space constraints. In the case of the smaller collection, the accesses to data and index had been interfering with each other, causing disk access delays that were largely absent when two drives were used.[7] Another aspect of research that this incident illustrates is the need for inventive experiments. Identifying a range of ways to alter the behaviour of a system, then measuring their effect, can lead to unexpected revelations.

Compilers are a substantial cause of systematic error. Versions of the same compiler can vary dramatically for particular programs, as can system software such as file managers. After an upgrade from one version of Linux to the next release, the time to run an external sort routine we were testing rose from 3,500 s to 7,500 s. However, a code profiler showed that some individual routines were running more quickly.

In another experiment, we were troubled by a random error. Sometimes a run completed in around an hour, but often took an hour and a half. Intermediate times were not observed. The experiment involved a complex interaction between stored data, a memory buffer, and temporary files, so some variability was reasonable, but we expected a spread of results—not two widely separated clusters. The explanation was the screen lock; while earlier experiments had been run on a server, we had recently moved to a high-performance desktop PC. The slower runs had been overnight, when the PC was not in use.

Test your intuition on the following example. Suppose you write a program for searching a large file of randomly selected strings. The first stage of the program reads the set of strings into memory, creates an array of pointers (one to the start of each string), then sorts the pointers so that the strings are in lexicographic order. The program then reads a query string and uses binary search in the array to find whether the query is present in the original string set.

1. If two searches in a row are for the same string, do both searches take about the same length of time?
2. Suppose the number of strings in the file is increased by a factor of sixteen. Do searches then take about four times as long?
3. Suppose linear search is used instead of binary search. If the original file is already sorted, are searches the same speed as for an unsorted file? (Recall that the pointers to the strings are sorted after the strings are read in.)

---

[7] Problems of this kind, and their solutions, can be highly illuminating. In this case, we discovered that disk seek times were a major component of total costs, accounting for around half of all elapsed time. Had we been explicitly investigating the significance of seek costs, we might not have thought of this experiment.

In an experiment undertaken by colleagues of mine in the mid 2000s, they found that the time required for disk-based algorithms could vary by as much as 15 %, depending on whether the data was stored near the inside or the outside of the disk platter. As they noted in their paper, to do such experiments “you may need to become intimately aquainted with the behaviour of your disk drive”.

Many programmers answer yes to these questions, but in each case the correct answer is no, largely because of the impact of cache on running time. (1) The first search loads the strings into cache. Memory access costs are a large component of total time, so the second search is much faster. (2) Two factors affect search time as data set size increases. One is that adjacent strings share longer prefixes, so the cost of a string comparison grows, as well as the total number of comparisons. The other is that, at some point, cache becomes ineffective because the volume of data means that there are too many collisions, and memory access costs rise. (3) If the file is sorted, the strings are likely to be sorted in memory, and will be prefetched during the linear search. If the file is not sorted, each string comparison requires a random memory access.

Intuition

Intuition is often unreliable in the context of statistics. Perceptual fallacies are well known, such as the elementary mistake that, if the last few coin tosses were heads, then the next is likely to be tails. Coincidences are more memorable than non-coincidences; thus they seem more common than is in fact the case. A long random sequence will have short subsequences that appear non-random. If a selected subsequence has pattern, it is easy to jump to an incorrect conclusion.

A well-known example is the three-box problem. A contestant in a game is told that one of X, Y, or Z contains a prize. The contestant chooses X but does not open it. The host then opens Y and shows that it is empty. Should the contestant change to Z or stay with X? Intuition says that it doesn't matter, as the probability of X containing the prize is 1/2. Careful analysis of the alternatives shows that Z contains the prize with probability 2/3, but when this problem was presented many mathematicians publicly argued that 1/2 was correct.

In a variation of this error, suppose that person P has tossed two coins. Person Q asks if one of them is heads, and P says yes. Then the intuitive estimate of the probability that the other coin is heads is 1/2, on the basis that the status of one coin is independent of the other, but again this is wrong. The correct response is 1/3. The reason is that there are four possible configurations: heads-heads, heads-tails, tails-heads, and tails-tails. Only tails-tails is eliminated by Q's question.

Intuition often seems to fail in the context of randomness and hashing. For example, given a uniform random hash function, the likelihood of a given key hashing to any particular one of 1,000 slots in a hash table is 1/1000. However, this does not mean that 1,000 random keys will be allocated evenly amongst the slots; the likelihood of this event is an infinitesimally low $\prod_{i=1}^{1000} i/1000$. Nor is it feasible for all values to hash to the same slot; the probability of even twenty values hashing to any one slot is absurdly small. On the other hand, statistical estimators such as the Poisson distribution can make accurate predictions of values such as the number of empty slots (around 368). Or an estimate can be formed with a simple simulation program.

Observers tend to make unsupported extrapolations from small numbers of events. A sequence of observations can be thought of as a tiny sample drawn from a vast population, and in statistical terms we would not expect a small sample to be representative. However, if a robot successfully traverses a room once, a researcher may well jump to the conclusion that the robot can always do so. The researcher has reasoned that the robot was designed to avoid obstacles; it successfully did so; and therefore the robot was working as intended. But whether this conclusion is reasonable depends on other context. For example, consider a robot that moves entirely at random. It may nonetheless traverse the room without encountering obstacles—sometimes, but not always. If we observed such a robot traversing a room, we could draw the inference that it was doing so by design. The general lesson from such cases is that a cautious researcher should consider whether any assumptions are statistically reasonable.

A related issue is of confirmation bias. A researcher runs an experiment, it fails, a problem is found; runs it again, it fails again, another problem is found; and again; but eventually the experiment appears to succeed. At this point the researcher claims success and regards the work as done, and may not even feel any need to mention the failures. But is the claim justified? A colleague told me of an instance in which he tweaked his motion-detection software again and again, until it finally worked—but only later discovered that he had been tweaking one version, but running another, static version. The "successful" run was pure luck. Claiming a positive result, detached from the context of failures, tuning, and exploration in which it was achieved, is not sound science.

Visualization of Results

We use computers to produce results, and can also use computers to help to digest them. One approach is to apply statistics. Another approach is to use visualization. Visualization of data is a substantial field in its own right, with a wide range of established techniques and principles. These are beyond the scope of this book, but should be explored by any researcher who has data sets that need rich interpretation.

However, even elementary approaches to reinterpretation of data via graphs can yield valuable insights. For example, curve fitting can be used to summarize data; and a graph showing the fitted curve can give a strong sense of whether the fitting was accurate. Graphs can also be used to interpret data from a variety of perspectives. The upper graph in Fig. 15.2 shows the number of events observed as a parameter, "depth", is increased. (This is real data from an experiment in information retrieval.) The crosses, joined by a jagged line, show the actual number of events. This graph illustrates that the number of events declines with increasing depth, but inconsistently; the long-term trend is unclear. A line has been used to connect the crosses to indicate overall behaviour. However, including the jagged line in such a graph is a mistake, especially if the number of points is small, as it wrongly suggests that there is a trend from point to point. A line is an interpolation between two points; if no data can be validly said to lie in that space, omit the line.

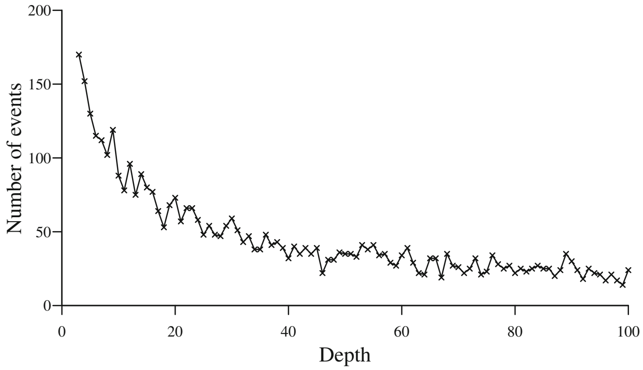


FIGURE 7. *The number of events observed at each depth; depths 1 and 2 have been omitted for reasons of scale.*

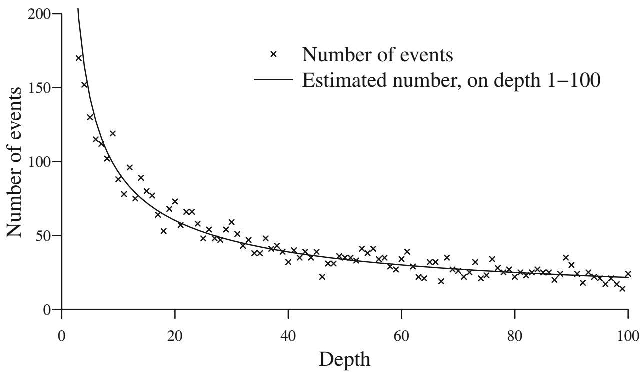


FIGURE 7. *The number of events observed at each depth; depths 1 and 2 have been omitted for reasons of scale. The solid line shows a best-fit to the points.*

Fig. 15.2 Curve fitting. In the *upper* graph, it is an error to show a line that implicitly interpolates the points, as such interpolation is not meaningful. The *lower* graph shows the quality of the fitted line, visualizing information that would not otherwise be intuitive

The lower graph in Fig. 15.2 shows the same events, without the jagged line. It instead shows, as a solid line, a linear regression of $\log(events)$ on $\log(depth)$ that has been used to find parameters C and p for the equation

$$events = C \cdot depth^{p} - 1.$$

The computation of linear regression returns a measure of error, but what this value means in practice is difficult to interpret. However, the visualization of the fit is striking: the line rides neatly between the points.

The graphs in Fig. 15.3 compare the ability of two systems to respond to 50 events (the score is a human-assigned value for quality of response; again, this is data from a real experiment). In the upper graph, System 1, with the crosses, often appears to be better than System 2, with the triangles; but in a reasonable number of cases the reverse is observed.

Which is better? Wilcoxon's signed rank hypothesis test reports that, for a specified level of 99 % confidence, System 1 is superior. This can be confirmed through visualization. One possible visualization is shown in the lower graph in Fig. 15.3, where the events have been sorted by the performance on System 1. The crosses now form a clear line; while a few of the triangles are above, the majority are below. It is a simple transformation, but highly informative. The data is simply a set of matched pairs: there is no innate sequence, and so the pairs can be reordered.

Another example of visualization is shown in Fig. 15.4. In the upper figure, a dot plot has been used to capture the relationship between the effectiveness of a baseline query evaluation technique and the improvement available through an alternative method. The hypothesis was that queries that were originally successful would be less amenable to further improvement than queries that were originally poor. Original effectiveness and new effectiveness are strongly correlated: a query that can be resolved with one method can also be resolved with the other. However, as the figure illustrates, there is no clear indication that poor queries can be improved more than others.

An alternative view is presented in the lower figure, where the effectiveness values on the horizontal axis have been averaged across subranges of width 0.05. This graph shows that the improvements are more or less the same, independent of the original effectiveness of each query, and thus suggests that there is no correlation.

Tools for visualization of data continue to develop, with rich mechanisms that allow dynamic interrogation and reinterpretation of the underlying behaviour. However, even elementary visualization can be extraordinarily revealing. Such analysis is often the best way to explore and explain data.

A "Statistical Principles" Checklist

- What variables might influence your results? Will analysis of these variables mean that you need to make use of statistics?
- Can you predict the effect of altering each variable? How do they interact? Are they independent?
- How do the experiments distinguish between the effects of the variables?
- Are effects random or systematic? How are they to be controlled?
- What method will be used to investigate outliers?

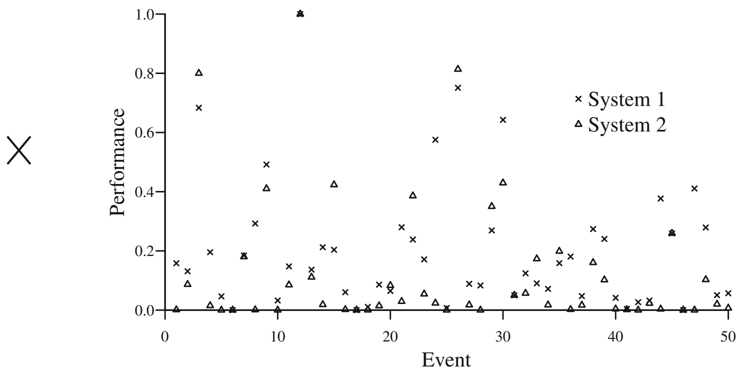


FIGURE 3.3. *The ability of each system to respond to an event, for each of 50 events.*

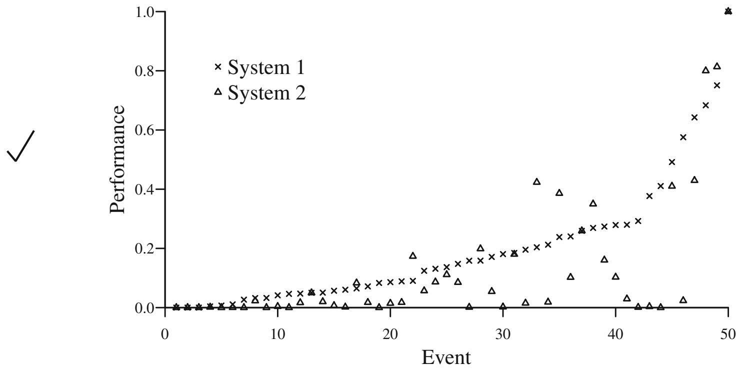


FIGURE 3.3. *The ability of each system to respond to an event, for each of 50 events. The events have been sorted on the score allocated to System 1, demonstrating that in most cases it has outperformed System 2.*

Fig. 15.3 Revisualization. The simple act of sorting the points according to score achieved by one of the systems shows how the performance of the systems compares. Even though System 2 is occasionally superior, the lower graph clearly shows that System 1 is better in most cases

- What is the population? How is a sample to be taken? What is the argument that demonstrates that a sample will be representative?
- How precise will the individual measurements be? How important is it to achieve a particular level of precision?
- What is the right way to summarize your results—an average? A median? A minimum?

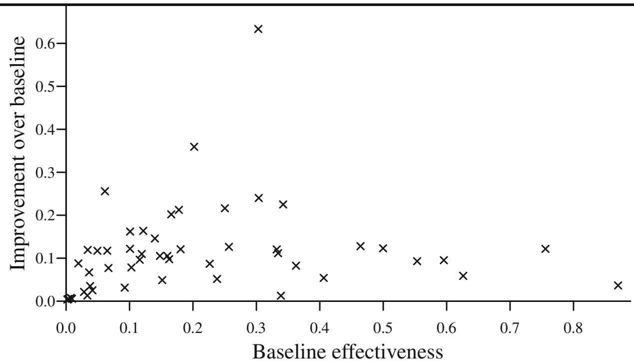


FIGURE 3. *For each query on the* FINNEGAN *data, original effectiveness versus improvement.*

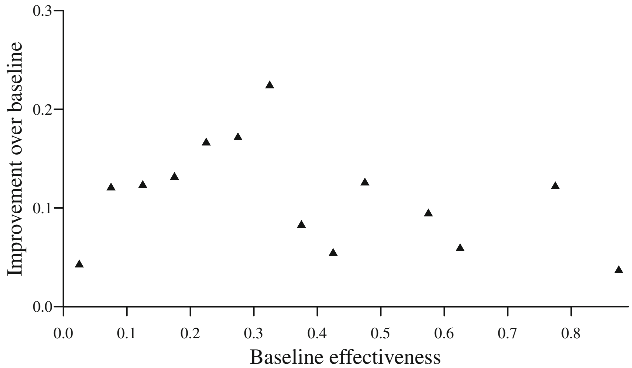


FIGURE 3. *Average improvement against original effectiveness, for queries on the* FINNEGAN *data. Each triangle is the average over a range of 0.05. Thus, for example, the average improvement for queries with effectiveness in the range 0.20 to 0.25 is 0.166.*

Fig. 15.4 Correlation. In the *upper* figure, no clear correlation can be seen between the two variables. In the *lower* figure, the same data is revisualized by plotting the average improvement in each of 20 sub-ranges. In this figure, averaging has removed the extreme data points and the trend is clear—there is no correlation

- What form of variance will be present, and how will it be captured in your results?
- How large is the effect you are hoping to observe, and how many measurements will be required in order to reliably observe it?
- Is a hypothesis test appropriate, and if so, which one?
- Do the results make sense? Are they consistent with any obvious points of comparison?
- What visualizations might help provide insight into the pattern or behaviour of the results?

Chapter 16
Presentations

Members [*use*] *a close, naked, natural way of speaking; natural expressions; positive expressions; clear senses; a native easiness: bringing all things as near the Mathematical plainness, as they can.*

Bishop Thomas Sprat
History of the Royal Society

You, having a large and fruitful mind, should not so much labour what to speak as to find what to leave unspoken. Rich soils are often to be weeded.

Francis Bacon
Letter to Coke

Scientists often have to talk about their work in front of an audience. Even experienced speakers can feel intimidated when they have to give a presentation, but the main challenges can be addressed with a straightforward approach based on good preparation, careful development of materials, and familiarity with the possible pitfalls. A nervous researcher needs practice to become an accomplished public speaker, but with the right approach even a first talk can be successful.

The purpose of a presentation is to introduce a research program and persuade the audience that the work is significant and interesting. There can be inaccuracies or generalizations that would be unacceptable in a paper, while obvious mistakes—or even correct statements that have not yet been justified—may be criticized immediately. Detail that is essential to a paper is often of little value in a presentation, so the principles of organization and presentation are quite different to those of a write-up.

Research presentations—primarily talks or seminars, but also posters—are the topic of this chapter. Some issues, such as speaking skills and good design of slides, are applicable to any form of presentation; while other issues are more specific. Experience with any kind of audience is of value, but may be only a partial preparation for learning to talk about research.

© Springer-Verlag London 2014

J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9_16

Research Talks

A research *talk*, or presentation, is typically a brief lecture about a particular piece of research, intended for an audience of other scientists. Another common form of presentation is a *poster*, where work is presented as a poster that is pinned to a wall or noticeboard and explained to interested passers-by.

A research presentation is used as an opportunity to discuss work with your peers. For example, it is used to examine your research directions, the outcomes of your research, issues and uncertainties you have encountered, or other work that you think is of interest. It may be used to explain to the audience why a certain paper is worth reading, or to present an overview of an area, or perhaps to get feedback on alternative research directions that you are contemplating.

Research presentations are different, in both audience and content, to other kinds of lecture, seminar, or talk.

- They are used to convey ideas, observations, and discoveries. The intent is to openly educate and inform, and also for the speaker to learn from the audience.
- The duration is usually fixed, say of 10, 30, or 60 min.
- Unlike lectures delivered to undergraduates in college or university, they are conversations between equals rather than lessons by professors; and there is no expectation that the audience is going to become expert in the material.
- Sales pitches or hyperbole are inappropriate—research talks are a contrast to business or information seminars, where the goal is typically to convey an impression, philosophy, or strategy rather than to educate.

Thus, for example, in contrast to the task of giving lectures, in research talks the timing is critical, while detailed explanations are in some cases unimportant; and the skills needed for management of the audience may be very different.

Typically, the audience is mixed. Some listeners may be actively interested in your precise area, but, usually, many will not be scholars of your topic. Instead, they are simply interested: they know you personally, are in your research group, attend seminars out of curiousity, or, maybe, had nothing better to do. A well-designed talk will speak to all of these kinds of listener.

To some extent, the style of the presentation is determined by its length. An hour or 45 min can have the space to motivate the work with a thorough example, explore side-topics, or run a discussion with the audience. For such a talk to work well, however, the structure (discussed below) must be carefully designed. In contrast, while a 25-min conference-style talk is long enough, in most cases, to convey the key details of a piece of research, the time needs to be used well—no one component can be allowed to run over 5 min. However, it is long enough for structure to remain important.

For a 10-min talk, though, it is rarely necessary to have an explicit structure. There is usually only enough time to introduce the topic and give a brief introduction to the method or results. There may be 6–12 slides, but certainly no more. And beware: high-speed delivery is not a good solution to lack of time. A brief talk of this kind is an opportunity to explain what the work is about, but not to explain the work itself.

Content

The fundamental challenging of developing a talk is to design a spoken tale. This tale should inform the audience, and have a structure that keeps their attention and answers likely questions. Such tales are often based on recent research or on a particular paper, and can be thought of as an introduction to the work that is the talk's topic.

Many researchers seem to think of the task of developing a talk as consisting of writing a deck of slides.[1] But this is only one element in the real task: creation of an engaging, flowing presentation. Slides are a significant component, but there are others; as well as artefacts such as webpages or videos, for example, there is the need for structure, narrative, and balance in the delivery. While a slide-authoring tool might be used to assist in assembly of the talk, it isn't essential in the early stages of drafting, and the focus should always be on the flow of information, regardless of how the information is to be communicated.

The first step in preparation of a research talk is deciding what to cover. Such talks are usually based on a paper or thesis, or on work in preparation, but most papers have far more detail than can be conveyed in a talk. Problems of this kind are highlighted in the experiences of research students. The initial reaction of a typical student preparing a talk is concern that there isn't enough to say, but the initial reaction of an advisor is, often, that the student's draft talk is far too long. Thus the content must be selected carefully.

What and how much to select depends not only on the time available but also on the expertise of the audience. A workshop attended by specialists in a narrow topic would suggest a different talk to one to be given to researchers in your department. Papers are usually specialized, but a diverse audience may be unfamiliar with even the area of your research, so it may be necessary to introduce fundamentals before proceeding to the results. For any audience, there is no need for a talk to just be an overview of the paper; it is an introduction to the ideas and research results described in the paper, and many paths can lead to that same outcome of teaching the audience about your work.

When constructing a talk, begin by choosing the single main goal, that is, the particular idea or result the audience should learn. Then work out what information is required before the result can be understood. Often this information is in effect a tree whose branches are chains of concepts leading to the result at the root. Much of the hard work of assembling the talk is pruning the tree, both to suit the audience and to strip the talk down to essential points that listeners should remember.

An approach to gathering material for a talk is "uncritical brainstorming, critical selection" (which should also be used for writing). In the first phase, jot down every idea or point that might be of value to the audience, that is, list the topics you might conceivably have to cover. Imagine yourself chatting with someone about your work, and note down the things you might say. During this first phase it is helpful to not

[1] Or set of slides. Or pack. Or, sometimes, of a set of overheads. To me, *deck* suggests the action of displaying the material in turn, following some sort of order or structure, and that is the terminology I use here.

judge each point, because questioning as you write tends to stall the brainstorming process. It can be helpful to set a time limit on this phase of no more than 20 min.

In the second phase, assemble the talk by critically selecting the important points and ordering them into sequence. During the second phase you should judge harshly because otherwise the talk will contain too much material; be lean and leisurely, not crowded and hasty.

A talk should be straightforward, although it can be used to convey complex ideas. Rather than asking yourself what you want to tell the audience—the interesting little issues explored, the technical problems confronted, the failings of the previous research—consider what the audience needs to know to understand the main result. Remember that a talk is a discussion with peers, not a news bulletin or political speech. There should be a logical reason for the inclusion of each part of the talk. Provide the minimum of detail that allows the audience to understand the result, while being inclusive. If the audience believes that they have learnt enough to confidently discuss the work with someone else, they will feel that the talk was of value. Think of the talk as a demonstration that the work is of value and, in particular, that your papers are worth reading.

Context can be as important as the ideas themselves. Take the time to explain why a problem is important, where it arises, or why previous approaches are unsatisfactory. Motivate the listeners so that they want to hear how a problem was solved.

Complex issues should be presented slowly and in stages; avoid detail that the audience is unlikely to follow. Once listeners do not understand the flow of the discussion, they are lost and will remain that way. Material that speakers shouldn't present, although some choose to do so, includes messy details such as the internals of a system, a proof of a theorem (attempting to walk the audience through a long series of logical steps is a particularly bad idea), the elements of a complex architecture, or technical information that is only of interest to a few specialists. There are of course cases in which such material is necessary—the proof might be the main idea to be conveyed, for example, or the theorem so unlikely that the proof, or its outline, is required to convince likely skeptics—but as a rule the audience will not enjoy being asked to understand intricate material that is unnecessary to appreciation of the overall result.

Some material, particularly abstract theory, is dry and difficult to present in an interesting way. However, there are alternatives to using the presentation to work through the details of a theory. Rather than just discuss the research, explain the relationship of the results to the broader research area. Explain why the project was worth investigating or consider the effect of the results on related work. Listeners who are interested in the theory itself can speak to you privately or read the paper afterwards.

A talk is an opportunity to discuss problems. A speaker who is not frank about shortcomings or difficulties, but is then exposed during questioning, can look foolish. Obstacles are part of doing research and, not only can they add interest to a talk, but just possibly the audience may offer solutions.

Never have too much material for the alloted time. Either you hurry through your talk, not explaining the ideas well and getting flustered, or you run over time, the audience is irritated, and the time for subsequent speakers is cut—not something for which they will thank you.

Organization

A crucial difference between a talk and a paper is that talks are inherently linear. A reader can move back and forth in a paper and has the leisure of putting the paper aside for a time; but in a talk the audience must learn at the speaker's pace and cannot refer to material that was presented earlier on. Talks must be designed within this constraint. A standard structure is of a sequence of steps leading the audience to the single main point.

Broadly, a typical talk is structured around the following components:

Motivation. Reviews the topic or problem, sets the context, and creates interest in the work.

Overview or goals. Explains where the talk is going, perhaps by presenting the structure, or, perhaps, by indicating what an attentive listener will learn.

Background. Reviews the state of the art, and achievements, highlights, and limitations of current work.

Contribution. Discussion of what is proposed, how to understand it, and how to appreciate its value.

Evaluation. The observations, experiments, demonstrations, and proofs—and their limitations—that support the claims made about the work.

Conclusions. What the audience should have learnt, and what the results imply for future work.

In other words, the organization of a talk is much like that of a paper. However, the structure of a talk is more fluid, and in some cases may be very different to the template given above. For example, a talk given early in a research project might focus on problems or difficulties rather than evaluation, with no contribution to discuss.

The typical structure does have potential pitfalls. In particular, take care to ensure that the relevance of the background is obvious. You will lose the audience's attention if they are wondering why you are discussing an apparently unrelated topic. Whatever the structure, ensure that all topics are relevant and follow an obvious sequence.

For the audience to follow the flow of argument in a talk, they need to understand its logical structure. The preview-do-review strategy is highly effective. That is, use backward and forward references ("I previously showed you that…", "I will shortly demonstrate that…but first I must explain…") to show how the current topic relates to rest of the talk. At changes of topic, summarize what should have been learnt by the audience and explain the role of the new topic in the talk overall. Distinguish between material that the audience must know to understand the main point and material that is minor or incidental. If you skip important detail, say so.

Getting the timing right, particularly for a short talk, can be difficult. Somehow the pace is never quite as you expect. It helps if your talk is designed so that there is material towards the end that can be skipped without breaking continuity, or included seamlessly if time permits. An effective strategy is to use repetition to emphasize major points: present a second example, or explain the impact of the work in several contexts. These repetitions can strengthen the talk, but at the same can be passed over if time is tight.

The Introduction

Begin well. The audience's opinion of you and of the topic will form quickly and a bad first impression is hard to erase. The first few sentences should show that the talk will be interesting—you can make a surprising claim, argue that some familiar or intuitive solution is incorrect, or show why the problem to be solved is of practical consequence.

Many speakers begin with an outline that lists the topics to be covered. At the beginning of the talk, however, the audience may not even be familiar with the terminology, and such outlines are quickly forgotten because they have no context. Outline the talk's structure if you want to, but not on the first slide. Before you reach the outline, make sure that the goal of the talk is clear. That is, explain where you are going before explaining how you will get there.

✗ "This talk is about new graph data structures. I'll begin by explaining graph theory and show some data structures for representing graphs. Then I'll talk about existing algorithms for graphs, then I'll show my new algorithms. I'll show experimental results on our cluster machine and then show why the algorithms are useful for some practical graph traversal problems."

Not only is this a poor introduction, but the outlined structure is poor too. (But note that the speaking style in this example is fine; it is a representation of a typical fluent speaker, punctuated for readability.) A better introduction is as follows, of a talk in which interesting material is discussed much earlier on.

✓ "My talk today is about new graph data structures. There are many practical problems that can be solved by graph methods, such as the travelling salesman problem, where good solutions can be found with reasonable resources so long as an optimal solution isn't needed. But even these solutions are slow if the wrong data structures are used. I'll begin by explaining approximate solutions to the salesman problem and showing why existing data structures aren't ideal, then I'll explain my new data structures and show how to use them to speed up the travelling salesman algorithms. I conclude with examples of where the new method makes a real difference."

The speaker should then continue on the topic of why the algorithm is useful, and, once the main concepts have been introduced, present the outline of the main part of the talk.

Some talks can be introduced with a tale or anecdote, to motivate the need for a solution to the problem or to illustrate what would happen if the problem were not solved. For example, a talk on automatic generation of acceptable timetables began with an account of the timetabling problems at a certain large university; the speaker made a good story of the estimate that, without new algorithms, the timetabling of a new degree utilizing existing subjects from several faculties would require 200 years. But in no circumstances should you try to tell a funny story unless you are an experienced speaker and are *certain* it will be funny.

All talks need a few words of preamble and warm-up. A surprisingly frequent omission is that speakers forget to say who they are! Begin with the topic of the talk, your name, the names of any co-authors, and your affiliation. If there are several authors, make sure the audience knows which one is you.

The Conclusion

End the talk cleanly; don't let it just fade away.

✗ "So the output of the algorithm is always positive. Yes, that's about all I wanted to say, except that there is an implementation but it's not currently working. That's all."

Clearly signal the end. Use the last few moments to revise the main points and ideas you want the audience to remember, and you may also want to outline future work or work in progress. Consider saying something emphatic—predict something, or recommend a change of practice, or make a judgement. Such statements should of course be a logical consequence of the talk.

Preparation

As a research student I was advised that the best way to prepare for a talk was to write it out in full so that (supposedly) if I froze I could just start reading from my notes. This was terrible advice. The writing of text that is fluid when spoken aloud is an art few people master. Written English sounds stilted, most speakers cannot scan far enough ahead to predict the right intonation and emphasis, and the act of reading prevents you from looking at the audience. Even the vocabulary of written and spoken English differ; for example, written English has "do not", "will", and "that" where spoken English has "don't", "shall", and "which".

Supporting notes can be helpful, if they are treated as prompts for issues to discuss rather than a script. Write notes as points of a few words each, in a large print that is easy to read while you are standing at a podium and doing things such as operating a computer.

Rehearse the talk often enough and the right words will come at the right time. You want to appear spontaneous, but this takes practice. A casual style is *not* the product of casual preparation. You will only be relaxed and deliver well if you have prepared thoroughly and are confident that you have prepared thoroughly. However, don't memorize your talk as a speech; decide what you want to say but not every word of how you will say it. Recitation sounds as stilted as reading and you are likely to freeze when trying to remember an exact phrasing.

Time the talk and note what stage you expect to reach at 5 min, 10 min, and so on, to help you finish on time. An effective exercise is to rehearse in front of a mirror or onto tape. Rehearse while standing, because that is how you will deliver. Think about possible questions. Familiarize yourself with equipment: for example, find out how to start up the computer, connect it to the projector, and run the presentation software. Last, get someone to give you feedback, and make use of it. If one person dislikes something it is likely that others will too.

Delivery

Assembly of the content is one aspect of a successful talk. Another aspect is creation of a cohesive sequence of slides, as discussed later. The third main aspect is presentation: speaking well, making good use of slides, and relating to the audience.

An obvious point is that you must speak clearly: develop sufficient volume and project your voice without shouting. Use a natural tone of voice. Breathe deeply, by inhaling slowly to the bottom of your chest. Speak a little slower than you would in normal conversation; around 300–400 words per 3 min is right for most people. Slightly overemphasize consonants, a habit that is particularly helpful to the 10 % or so of your audience who are at least a little deaf. Avoid chilled drinks, which can tighten the throat. Keep your head up, to help maintain volume. And face the audience.

Consider your style of speech. Avoid monotony, both in pace and tone. Pause occasionally, particularly when you have given the audience something to think about, and pause in preference to filling gaps with noise such as “um” or “I mean”. Pause to collect your thoughts before speaking rather than pausing mid-sentence.

Never read your slides to the audience—they can read faster than you can speak. This is perhaps the commonest mistake made by inexperienced speakers, and it is certainly one of the most irritating. The text on your slides should, at most, be a reminder to you of the concepts you wanted to mention.[2] As you prepare your talk, you should be developing a set of messages you want to say to accompany each slide (or is the slide accompanying you?); these messages, not the written text, are the core of your talk.

[2] I’ve noticed that different disciplines have their own conventions for slides, and in particular that in biomedicine slide decks often consist almost entirely of images and tables. In such cases, the talk proceeds by explaining each slide in turn. It helps that biomedicine has research topics that are so visual! Common images include devices, places, activities, genomes, cells, organisms—from bacteria and viruses to trees and whales—and representations of statistical data. Also, intriguingly, biomedical slide decks often include photographs of members of the research team, and of historical figures, a practice that is rare in computing.

At the other extreme, some disciplines still have the convention of the speaker reading from a written script, with no slides at all. While it is rare that such talks are engaging, in what I remember as the best talk I have ever attended the speaker used a single slide, which consisted of a complex Venn diagram showing the relationship between government, politics, industry, and media. The theme of the talk was the impact of these bodies on researchers and funding at different stages of their careers. Admittedly, this was opinion rather than science.

Also consider the personality you present. As a speaker you want to be taken seriously, but this does not mean that you cannot be relaxed, vivid, even amusing. Show your enthusiasm. Avoid sudden movements or distracting mannerisms such as pacing, bouncing, or gesticulating, but don't freeze; gestures should be natural. Vary what you are doing: move away from the computer to talk to the audience directly, for example, spend a couple of minutes with a non-technical slide after working through complex material. Make frequent eye contact with the audience; find some friendly faces to check with every now and again. Above all, be yourself—don't adopt a false persona and don't show off. The right note to hit is of a conversation with friends.

Showing off, swagger, or vanity of any kind, is if anything worse in a talk than in a paper. Be modest. Don't talk down to the audience or make aggrieved statements such as "people all said it couldn't work, but my work proves them wrong". Maybe the work is indeed remarkable, but that doesn't mean that the speaker is too; keep the distinction between presentation and presenter clear.

At the same time, you shouldn't diminish your achievements. Avoid excess humility, don't suggest that the outcomes are unimportant or uninteresting, and don't begin by saying that the talk will be dull or that you are nervous. Too many talks begin with a disclaimer such as "the talk was only written last night" or "I haven't had time to prepare". The intention is to lower the audience's expectations and thus mute any possible criticism, but the effect is to diminish their interest; and, if the talk turns out to be excellent, the disclaimer is then an unfortunate boast.

Beware of irritating habits. "Umming", pacing, and gesticulating were mentioned above. Consider taking off your watch; if it is on your wrist you cannot check the time inconspicuously. Only drink if you absolutely have to; if you have to drink, don't gulp. Don't read directly from slides or written notes, or stand behind the projector so that your face can't be seen and you cast a shadow on the screen. When referring to the screen, use a stick or laser pointer rather than the computer's mouse. Don't overact, use slang, or laugh at your own jokes. Don't act nervous, mumble, look at your feet, face the wrong way, scratch, fiddle, or fidget. If you think you might be tempted to rattle the coins in your pocket, put them somewhere else. And don't change slides before the audience has had a chance to read them.

Handle distractions tactfully. If someone persistently interrupts, or excludes the rest of the audience by asking too many questions, offer to talk to them afterwards.

Expect to be nervous—adrenaline helps you to talk well. Even experienced speakers can be nervous, despite their appearance of cool calm on the podium. The best cure for serious attacks of fright is to give a preparatory talk or two, so if possible practice before a friendly (but critical) audience. A constructive attitude is to view each talk you give as training for the next one. Don't be too ambitious; master the basics of getting a clear message across before, for example, attempting to tell jokes or make advanced use of presentation tools.

Standing in front of an audience of your peers or superiors can be intimidating, particularly if the audience is silent. But silence is a good sign; it means people are paying attention. Even yawning isn't necessarily a disaster; lecture halls are often stuffy, and nobody stays focused indefinitely. A typical listener's attention drifts away momentarily now and again, no matter how good the speaker is.

Most importantly, remember that the audience wants to enjoy your talk—their attitude is positive. People don't attend talks with the intention of being bored, and welcome any sign that the talk is interesting. The need to build on this initial goodwill is why opening well is so important.

Question Time

Question time at the end of a talk is used to clarify misunderstandings and to amplify any points that listeners want discussed in more detail. Five or ten minutes is too brief for serious discussion: you need to keep answers short and avoid debating with an audience member, because it is annoying for everyone else. Some questions can't be answered on the spot: they are too complex, or the questioner has misunderstood a fundamental issue, or you simply don't know the answer.

Involve the audience in question time. Repeat the question in your own words and talk to the whole audience, not just the questioner, in your reply. Respond positively and honestly to all questions. Never try to bluff when you don't know—doing so can only look stupid. It is far better to be frank and admit ignorance. It is equally important to never be rude to audience members or dismissive of their questions. Questions can be misguided, irrelevant, or amazingly inane, but more than one audience member may think such a question to be reasonable, and the only appropriate response is to answer as politely and accurately as the question permits.

Slides

Slides are a point of focus for the attention of the audience. Text on slides is a visual guide to what the speaker is saying. Figures—graphs, images, diagrams, or tables—show results or illustrate a point.

However, keep in mind that the focus of the talk is you, not the slides. What you are saying, rather than the sketchy content of a slide, should be the centre of attention. That is, you shouldn't use slides as a way of avoiding contact with the audience.

Some principles to keep in mind when developing a deck of slides are discussed below. This is not an exhaustive list, so use it to guide your own sense of what is appropriate.[3] Further issues are considered later, in the context of particular kinds of slides.

[3] Earlier editions of this book included examples that illustrated positives and failings in slide design. As technology for presentations has developed and diversified, and a wide range of templates has emerged, it cannot be argued that there is a single "good" style. Moreover, elements such as dynamic images and animation are not easily captured in a printed book, and thus illustrations printed here would inevitably seem (no pun intended) unilluminating.

Curiously, some of the slide decks on the topic of "how to present well" seem to me to be chaotic, crude, or unpolished. It is clear that there is little consistency as to what is regarded as good taste and good style.

Individual Slides

Each slide should have a heading and be fairly self-contained; don't rely on the audience remembering complex details or notation introduced elsewhere. Before including elements such as equation numbers, consider whether they will be of value to listeners. A slide is unlikely to be entirely self-contained; the audience expects it to be explained by the speaker and to be based on material introduced earlier. But take the effort to make the slide reasonably complete.

Aim for about one slide per minute or so—too few is dull and too many is bewildering. It is a mistake to design a talk so that rapid back-and-forth switching between slides is required. Consider instead repeating crucial information. For instance, show a whole algorithm, then on successive slides show each step with an example.

Slide Tools

The tools for making and presenting slides continue to develop. Those in wide use—such as Microsoft® PowerPoint and LaTeX—provide excellent environments for writing slides, and include a range of elaborate features. Each tool has different strengths; for example, Microsoft® PowerPoint allows rapid drafting and easy layout of images, while LaTeX supports talk structure well and is preferable for maths and tables. Even an inexperienced speaker can use these tools to produce a professional-looking talk.

Slide tools are an effective way of drafting talks, once the content has been identified. My approach is to create a series of empty slides with indicative headings, bring in the images and tables that I want to use, start to add text (which might at this early stage be editorial remarks, such as "discuss initial method here"), and use many-slides-on-a-page output to get a sense of how the structure is developing.

The fact that an authoring tool provides features does not mean that the features have to be used—ease of use and necessity of use are not the same thing. The principles of a good presentation have not changed since the era of handwritten overheads: legibility, simplicity, and relevance. In too many talks, the speaker has decided to use some element of the software that neither amuses the audience nor helps them to learn. The goal of a well-written talk is for the audience to listen to the speaker; distractions, no matter how nifty, should be eliminated.

The need to avoid distractions is particularly acute with presentation tools that have mechanisms for animating the flow of the talk. They can be entertaining, but do they educate? Do they help create an understanding, or are they just flashy? A straightforward, elegant slide design may be less dramatic than the alternatives, but does not annoy; and experienced listeners are unlikely to be excited by your ability to use the latest software.

Layout

Choose an effective slide design. Simplicity works well, while complex designs and bright colours are a distraction. Consider including running headings and structure guides, so that the audience knows what part of the talk they are listening to.

Dark backgrounds do not always work. In a dimly lit auditorium, if the projected image is dark the atmosphere can be unpleasant. Light fonts on dark background do not display as well as dark fonts on white. Backgrounds that vary in brightness are even worse—text that is legible in some areas may be obscured in others. The use of logos and images should be limited to borders. If you have something to communicate to the audience, screen area is precious; don’t waste it on meaningless graphics.

Animation

Animation that is used for stylistic reasons, such as transition between slides, or transition between bullet points, quickly becomes tiresome. Animated diagrams can be an effective way of illustrating the working of dynamic systems and algorithms, but otherwise animation rarely adds value.

A specific form of animation is point-at-a-time display. There are several reasons why such display works against the success of a typical presentation. One is that it is a constraint on the speaker, who must keep to a rigid script and remember several times a minute to click a button to get the next point displayed. (All too often the speaker does not remember, and then has to click-click-click to catch up.) In contrast, if a whole slide is displayed, the speaker can focus on talking to the audience and can improvise more easily. Another reason is that audiences want to know where the speaker is going; typically a listener reads a slide, then waits to hear the speaker explain it. Point-at-a-time display makes it harder for the listener to focus.

Other Elements

Some talks include materials such as web pages, audio recordings, and videos. These materials can be valuable, but they do bring risk: some auditoriums and projection systems do not support them well. Also, talking to such material takes practice, particularly a video that seems a lot longer (and more boring) when played to an audience than it did when you were writing the talk.

If you add music or noises to individual slides, do so for a serious purpose. Don’t expect the audience to be familiar with cultural choices—they are unlikely to share your taste in music, for example, or be able to guess why you think a particular unfamiliar song is relevant to your talk.

Copyright

The issue of plagiarism and use of other people's material is discussed in Chap. 17. You might be tempted to think that, in a talk, which may leave no permanent record, it is acceptable to use images, figures, and text created by other people. However, as a general principle, you should only include materials that you have the right to use.

Consider the text or figures on your slides. Was any of it authored by someone else? If so, will that person be in the audience? What about people who are familiar with that person's work? If people notice that you have used someone else's materials, they could well react immediately, and ask difficult questions in front of a broad audience.

Even if reuse of material is not noticed, it still can have inappropriate consequences. One is that people make judgements about you, and your capabilities, based on your words and images—and in this case their inferences will be false. It is unethical to deceive people about either your work or your competencies as a researcher.

Sometimes the borrowing is deliberate and obvious, and thus may not seem deceptive. An example is use of cartoons from newspapers. But are they licenced for you to use? And the lifetime of your talk may be longer than you expect. It may be recorded, and possibly even made generally available via a conference or university website. You may re-use the slides later on, or your advisor might ask to borrow them. Inappropriate inclusion of material that is not yours might never be noticed, but, if it does come to light, could cause serious embarrassment.

Text on Slides

The text included on slides provides structure and context. It is usually written in point form; the points should be brief summaries in short sentences of the information you want to convey. The audience will expect you to discuss every point listed on each slide, or, rather, expect that by the time you switch to the next slide the content of every point will have been covered. Each point should be a topic to discuss, not necessarily a complete statement in itself. A slide may be a series of points, but that doesn't mean that the points need to be numbered or even bulleted. Some people argue that bullets add interest; slide after slide of slabs of text can be dull, but bullets (which are greatly overused) do not make much difference.

Some speakers use a kind of pidgin-English for their slides.

✗ Coding technique log-based, integer codes.

Be brief, but not meaningless.

✓ The coding technique is logarithmic but yields integer codes.

Explain all variables and where possible simplify formulas. In papers it is helpful to state types of variables when they are used; in talks it is crucial. Minimize the volume of information, especially detail of any kind, that the audience must remember from previous slides.

Use a font of reasonable size and have plenty of white space. Huge or small fonts look ridiculous. In particular, slides should not be crowded with text. If there is too much text, it is likely to be unreadable and will be a distraction from what you are trying to say. Never display a page from a paper: even a well-designed page is almost certainly unreadable in the context of a presentation. Don't break words between lines; instead, have an uneven right margin. Keep the layout simple—minimize clutter such as frames, shading, cross-hatching, shadows, and artwork.

Explore the available fonts, but don't worry too much: while to some people sans serif may look cleaner than the alternatives, for example, any reasonable font is fine. Be consistent, however; one talk needs no more than three fonts and a couple of font sizes.

Strings of exclamation marks and text in uppercase do not add a sense of excitement. They add a sense of ineptitude.

Figures

Good figures and graphs can make ideas much easier to understand. As a general rule, if a picture adds value to a slide, then use it. Even a simple picture can make a big difference to both the appearance of a slide and the way in which the material might be delivered. Figures should be simple, illustrating a concept or result with minimum fuss; messy or crowded figures have no impact. Tables should only be included when necessary—they can be hard to digest.

An illustration from a paper may not be appropriate for a talk. In a paper, the reader can consider the figure at leisure, but in a talk it is only shown for a limited time, and the freedom of the presenter to point to the parts of a figure and to add to it incrementally means that it may be appropriate to organize the figure rather differently. Perhaps most significantly, in a talk a figure can be dynamically coloured in a variety of ways. For example, text can be in different colours to show an ordering of events; different kinds of entities can have different colours; or colouring can be used to show how routes through a process relate to outcomes.

Figures in slides, as in papers, should focus on the technical content. Distracting elements should be removed. Present the bars of a histograms in three dimensions only if the third dimension carries some information. Keep all objects to a reasonable size—why include a gigantic block-coloured arrow when a simple line will do? Include an image or movie only if there is a need to do so. Animate only if necessary, such as when explaining a data structure.

Clip art, especially of stylised people, can look silly and is often ugly. It does not add class. Use it only when necessary, and select the simplest picture that suits the need.

Label everything, or at least every kind of thing. The labels should be meaningful to the audience—if you have omitted material from the talk, omit corresponding material from the figure.

Posters

A poster session can be one of the most vibrant parts of a conference. Whole halls can be filled with lines of posters, with a presenter in front of each one ready to explain it to interested listeners. Conference attendees, often in a dense crowd, walk through the posters. They glance at some, read others, or stop to talk to a presenter—for a moment, a minute, or, sometimes, for hours. Poster sessions are perhaps the single best opportunity for a researcher to meet new colleagues.

A well-designed poster will meet the needs of all of the people who pass by. As is true for a research paper, readers should be able to quickly tell if it is not of interest to them; no-one wants to spend their time reading something that turns out to be irrelevant. For those who are mildly interested, the main points should be presented so that they can be digested in a few moments. For attendees who are genuinely interested in the work, a strong poster has content that supports the presenter, who should have prepared a detailed story about the research that the poster represents.

A good poster, then, is a balance of several separate aims. It serves as a way of attracting the interest of people passing by, a summary of the work, a support for both brief and detailed conversations about the research, and a demonstration that the work has been undertaken in a robust way.

Content

The first steps in design of a poster are much the same as for design of a talk: assembly of the content. My advice on getting started is as for talks, given earlier in this chapter. Before the poster can be assembled, you first need to know what story you want to tell; this will be an overview of the research, motivation, and outcomes and not, in most cases, the detail of the work itself.

The narrative that accompanies a poster should derive from what you plan to say. Don't set out by asking how your work can be crammed onto a single A0 or A1 sheet of paper. Instead, ask yourself how the work can be explained in a minute or two, and what illustrations and text will help the explanation to proceed smoothly.

For a talk, there is just one version of the narrative. For a poster, the narrative can be different for everyone you speak to, depending on their background and level of interest. You should consider what the 1-min explanation of your work will be; and what might be needed in the 10-min version.

Organization

The structural elements of a poster are, in general terms, the same as in a talk, but are laid out hierarchically rather than shown in linear order.

Posters can be vertical (portrait orientation) or horizontal (landscape). Landscape is preferable—it communicates better, and is easier for the listener—but in a crowded poster hall may not be an option. The first step in poster design is to establish whether landscape is available, and then you need to establish what size you can print (A0 or A1), which will determine how detailed the poster can be.

An authoring tool will be used for most of the work of creating your poster, but in my experience the early stages are most easily done by hand, on either paper or whiteboard. Typically, my first step is to choose a couple of figures and tables that I think will be of value, and then do a rough sketch of the poster on an A3 sheet of paper, with boxes showing where the main elements will be placed. I then annotate this sketch with bullet points, giving a first draft of the text that might be included. These rough sketches can save a lot of work later on.

I also use paper to sketch potential diagrams. Compared to the style that is typical of a paper, these can be more dramatic or artistic (though you should only create such images if you can do so at a high standard). A poster should be attractive to potential listeners, and vivid illustrations help work to be noticed, while an A1 block of solid text in small font is extremely uninviting. But stay within reasonable bounds—you are attending a conference, not an exhibition, and you should avoid implying that your work is not serious science.

A useful way to think of a poster is as an arrangement of discrete regions, each used to communicate one element of the work. One box might be a figure, another some bullet points motivating the problem, another a table giving results. The boxes can be arranged in a simple grid, but other layouts may make the poster seem more dynamic—you should experiment with alternatives. A particular factor to consider is height. It is easier to read and comprehend material that is at eye level; in a landscape-orientation poster, roughly speaking, it is easiest to read the top half, while in portrait orientation it is easiest to read the top third. The lower part of the poster can be used for the detail that only a truly interested listener will want to explore, and thus can be in smaller fonts and have a more technical appearance. The upper part is where the succinct, lightweight version of the narrative is delivered.

Your choice of authoring tool may be limited by the printing facilities to which you have access; for example, some specialised image tools do not produce PDF output. Inkscape is widely used for posters, as is Microsoft® PowerPoint. You can use these tools to create poster elements separately, to see how big they need to be, before deciding how they are going to be integrated into the whole. But whatever tool you use, plan to spend significant time in polishing and revising—good layout is a highly manual task.

Occasionally a presenter will have a poster that is a series of ten or twelve slides, each printed on an A4 piece of paper, pinned up in rows. This design does not communicate well, and, often, such posters fail to attract any interest.

Presentation

A poster presentation consists of a series of conversations with attendees, often with one conversation running into the next as the mix of listeners changes. The questions from the listeners may drive the conversation, but nonetheless you should prepare points to discuss; some listeners will ask you to tell them about your work, and wait for you to tell the story.

That is, part of your preparation should be development of a few mini-speeches, of a minute or two each, concerning elements of the work that you expect to have to explain.

Perhaps the most difficult part of presenting a poster is the quiet moments, when people are walking by without showing interest. In that situation, presenters seem to experience one of two contradictory impulses. One impulse, more or less, is to run away. To a limited extent, this is fine; there is nothing wrong with taking a few minutes off to see other posters, or to get a drink or a snack. However, you should not underestimate the value of the conversations you do have, so even if there are periods when interest is low you should take every opportunity to discuss your work with people.

The other impulse is to try too hard to get people's attention—to attempt to draw them in, and keep them in front of you, even if they are not particularly interested in the work. My experience is that a little seeking-out of attention is acceptable, but if you do so you must remain alert to the possibility that your audience is merely listening out of politeness, and thus you should give them opportunities to move on. When there are dozens or hundreds of posters for people to visit, it is unreasonable to try to delay people who may have other work they wish to see. Overall, in my view the best strategy is to wait until people have made eye contact with you before asking if they would like to hear about your work.

Some people may wish to contact you afterwards. Have a business card or email address handy, in a form that you can pass to people without unnecessary fiddling with phones or contact lists.

A "Presentations and Posters" Checklist

Regarding *the content of your talk or poster*,

- What is the key thing the audience should remember?
- Is there enough background material for the intended audience?
- Is any material unnecessary? Could some of the material be left for people to read about later?
- Is the talk self-contained? Is it appropriate to an audience of mixed background?
- Is the length appropriate? Is the structure right for the length?
- Does the talk have a motivating preamble?
- Is the talk balanced, without too much time given to any one element?

- Are complex issues explained in gentle stages?
- Are the results explained? Is the impact of the results made clear?
- What were the limitations of the research? Where are they discussed?

Regarding *your slides*,

- Have you found good tools, or methods, for drafting a talk?
- Are figures uncluttered, with legible, horizontal text?
- Is there any unnecessary animation? Is the style appropriate, or flashy?
- Are the font sizes reasonable?
- Are the numbers necessary? Are more diagrams needed?
- Are the slides simple? Do they have unnecessary ornamentation or distracting use of colour?
- Does each figure illustrate a major point? Does it illustrate the point unambiguously?
- Are there enough examples?
- Do you have the right to use the figures and illustrations?

Regarding *the presentation of your talk*,

- Have you prepared something to say about each slide?
- Do you explain why the research is interesting or important?
- Is there a clear conclusion?
- Have you rehearsed the talk? What mechanisms are you using to keep yourself to time?
- Have you memorized the talk?
- If you are asked a question you can't answer, how will you respond?
- Have you rehearsed your manner? Will your enthusiasm show?
- Do you know how to use the equipment?

Regarding *your poster*,

- Do you know how large it can be, and in what orientation?
- Does it have figures and summaries that support a brief explanation? Are the key messages conspicuous?
- Have you prepared a brief explanation of the work? Does it include the *why* of the research, as well as the *what*?
- Is there detail that supports a detailed discussion?
- Is it arranged in an accessible, sensible way?
- Is it vivid and attractive?
- Does the form overwhelm the content?

Chapter 17
Ethics

People will work every bit as hard to fool themselves as they will to fool others.

Robert Park
Voodoo Science

The Piltdown hoax … seriously delayed and distorted the urgent work of science … Young scientists and old alike wasted untold thousands of hours on the Piltdown phenomenon … [*It*] *was nothing short of despicable, an ugly trick played by a warped and unscrupulous mind on unsuspecting scholars.*

John Evangelist Walsh
Unravelling Piltdown

These words hereafter thy tormentors be!

William Shakespeare
Richard II

Science is built on trust. Researchers are expected to be honest, and research is assumed to have been undertaken ethically. For example, referees assess whether results are significant but are not expected to investigate whether the reported experiments actually took place, because it is assumed that the authors have not lied about their work.

The major societies of science have codes of conduct that scientists are expected to adhere to. Breaches of these codes are regarded as extremely serious; even senior, respected academics have lost their positions after having been found to commit misconduct. Familiarity with these codes and their implications for day-to-day work is essential for a practicing scientist. In brief, the scientific community expects published research to be new, objective, and fair; researchers should not present opinion as fact, distort truths, or imply that previously published results are original; research should be undertaken within relevant ethical frameworks, which protect privacy and minimize the risk of harm to individuals; and researchers should not plagiarize others or misrepresent their contributions to the work. However, there is no international academic police force, rule of law, or investigative agency. Ethical issues are largely managed by individual adherence to the expectations of academic community,

© Springer-Verlag London 2014

J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9_17

and through the actions of self-organizing groups that respond to ethical breaches when they do occur.

The most conspicuous form of unethical behaviour in computing is plagiarism, because it steals work from other scientists and the hurt to others is obvious. (Also, it is relatively easy to detect.) However, other forms of misconduct are arguably as pervasive. One is abuse of power, such as when senior academics insist on being listed as authors of papers they have not contributed to. Another is fraud, in which claimed results were not in fact observed, or are much less substantial than was claimed. In medicine, fraud is viewed as serious because of the potential consequences—deaths and vast financial liabilities—and because of high-profile cases in which fraud has been detected. In computer science, there is also the potential for such issues. The safety, reliability, and security of computer systems is increasingly a central element of our social and physical infrastructure. Researchers who make grandiose claims based on poor evidence and whose work is subsequently used in practice are creating risks, and may be held responsible for the consequences.

Issues of ethical concern for science in general include misrepresentation, plagiarism and self-plagiarism, authorship, confidentiality and conflict of interest, harrassment and abuse of power, and use of human subjects. The ethics of studies of human subjects are complex, and are beyond the scope of this book. The other issues are discussed in this chapter.

It would be satisfying to be able to give a formula for handling ethical issues. However, the two principal pieces of advice on this topic contradict each other. One is that problems that at first sight seem to be intractable ethical conflicts often turn out to be more superficial; it is sensible to wait and reflect before pursuing action. The other piece of advice is that unresolved tensions can fester, with the potential to permanently damage a relationship; it is sensible to take steps before too much harm is done.

A difficulty in resolving such issues is the imbalance between advisor and student, or between senior and junior academics. If you believe that there is an issue that cannot be resolved fairly by a direct approach, you need to seek confidential advice and support, preferably from a senior academic who knows the individuals involved and has an understanding of the norms of research in that field. It can be difficult to take this step, but it is essential. Keep in mind that public accusation, justified or otherwise, can end a career. Mishandled, a genuine grievance can become a scandal in which any of the participants is a potential victim. Moreover, while issues such as whistleblowing and breaches of research ethics can be highly politicized, and it can be intimidating to approach a senior figure with accusations about a colleague, it is far better to discuss problems with a trusted figure than to let rumour, or silence, destroy relationships or reputations.

Intellectual Creations

Many of the ethical issues that arise in the context of research are related to the ownership of both ideas and descriptions of them. Loosely, these might be described as *intellectual creations*,[1] that is, the stuff that knowledge workers make, including concepts, inventions, discoveries, designs, documents (text, images, or video), or code. People might own their intellectual creations, or the creations might belong to their employer, or a publisher.

Some forms of intellectual creation have an established legal framework. Informally, *copyright* concerns the expression of ideas; that is, the words and pictures used to say something are protected, but not the ideas themselves. *Patents* and *prior invention* protect processes, inventions, and concepts; that is, the ideas, but not how they are described.[2] When a researcher writes a paper, the content represents new intellectual property, which may be patentable. The copyright in the form or presentation is held by the author; typically, when the paper is published the author assigns copyright to the publisher.

Thus the publisher owns the text, figures, and diagrams, and the author—or anyone else—cannot legally use these again without the publisher's written permission. Specifically, copyright prohibits use without permission of pictures and diagrams, or of substantial volumes of text. Brief pieces of text can be used so long as they are attributed, that is, quoted.

It follows that legal frameworks and ethical frameworks are very different from each other. In law, plagiarism is usually infringement of copyright. In academia, plagiarism is meant more broadly, to mean theft of any intellectual creation. Perhaps confusingly, the ownership of intellectual creations in research papers is in many circumstances not covered by law, while academic plagiarism does not always involve infringement of copyright. Laws vary from country to country, while academic expectations are reasonably consistent internationally and across disciplines. The academic community's standards—which are sometimes unstated—reach well beyond what laws tend to require, and it is within these standards that academic plagiarism should be understood.

Plagiarism

A central element of the process of science is that each paper is an original contribution of new work. Scientists' reputations are built primarily on their papers: both the work and how it is reported.

---

[1] Or, more formally, intellectual property—a term with a great deal of legal baggage that means different things in different contexts.

[2] These are separate from the legal frameworks that concerns *fraud* and *misrepresentation*; these protect the identity of individuals (among other things), and how the individual's identity is used.

Plagiarism is re-use in one paper of material that has appeared in another, without appropriate acknowledgement. The theft may involve ideas, illustrations, results, text, or even whole papers; and includes, not just copying from published papers, but from material in electronic form, such as Web pages, news articles, or email. By plagiarizing, a researcher hopes to obtain credit for work that has already been published, and not necessarily by someone else (the issue of self-plagiarism is discussed in the next section). However, while some people do make a deliberate decision to steal and there is a complex range of factors that lead people to plagiarize, one cause seems particularly common: misjudgement by an inexperienced researcher.

Such misjudgements can arise when a research student is unaware of appropriate academic style. For example, a researcher investigating B-trees may find an elegant illustration in a textbook and decide that it is perfect for a forthcoming paper; but copying this illustration (either by reproducing it or by imitating it) is plagiarism. Similarly, a researcher describing B-trees may feel that a paragraph in a reference cannot be improved on; but copying it verbatim is plagiarism. Even a close paraphrase of it is likely to be plagiarism.

Another form of misjudgement is inappropriate or inadequate citation. Suppose that Barlman and Trey (2001) wrote the following:

> The impact of viruses has become a major issue in many large organizations, but most still rely on individual users maintaining virus definitions, with no internal firewalls to protect one user from another. However, any structure is only as strong as its weakest link; these organizations are highly vulnerable.

It would then be considered plagiarism to write the following:

✗ Viruses have become a major issue in many large organizations, but most organizations still rely on users maintaining virus definitions on their individual computers, with no internal firewalls to protect one computer from another. However, any structure is only as strong as its weakest link; these organizations are highly vulnerable to infection (Barlman and Trey 2001).

In this example, a citation is given, but it isn't made clear that the citation refers to the whole block of text. Also, there is nothing to indicate that the wording is unoriginal—despite a few small changes, the text is copied. If the wording or the sense of the original text is required, it would instead be appropriate to write something like the following:

✓ As discussed by Barlman and Trey (2001), who investigated the impact of viruses in large organizations, "most still rely on individual users maintaining virus definitions, with no internal firewalls to protect one user from another. However, any structure is only as strong as its weakest link; these organizations are highly vulnerable."

Alternatively, the essence of the original can be concisely summarized, with clear attribution:

✓ Barlman and Trey (2001) investigated the impact of viruses in large organizations. They found that organizations are vulnerable if individuals fail to keep virus definitions up to date, as internal firewalls are rare.

The lesson of this example is that citation by itself is not sufficient. It is necessary to indicate exactly what material is taken from the reference, and to identify that material as a quote.

The following is adapted (to protect the guilty) from a real example:

✗ This distribution of costs follows a power law [2] in which only a few tasks have high impact. The form of the law is [13] for fixed cost C given by $P(x > C) \sim 2^{-\alpha}k$ where $\alpha > 1$ and $k > 1$. The parameter α describes user behaviour. Determination of k for a specific application can be achieved through modelling as a Poisson distribution.

In this example, everything but the citations are copied from the reference "[2]", including the erroneous misplacement of k, which should be a superscript.

Paraphrase of the structure of a paper is also plagiarism. If one paper follows another to the extent that they use the same headings, have tables of the same layout, cite much the same background literature, describe the literature with respect to the same criteria, and have similarly designed experiments with similar data exploring the same properties, then the second paper is arguably plagiarized.

These kinds of plagiarism can happen when trying to draft a paper (or impress an advisor). An author might, for example, copy the background of a paper with the intention of replacing it later on; or an advisor might give a student an existing paper to use as a model, and the student might then keep some of the text; or any of a range of such scenarios. Without adequate guidance about plagiarism, it is understandable that inexperienced scientists make mistakes, especially when other similar mistakes are in published papers.

It is easy to avoid plagiarism. When writing fresh text, avoid using other text as a guide, even if you are discussing outcomes reported by someone else. Cite other text, and be explicit about which material in your work is derived from elsewhere: mark where the cited material begins and where it ends. Use quotation marks for borrowed text. Construct reference lists by enumerating the papers you have read, not by copying the lists in other papers. And design all your own pictures.

For advisors, a lesson is that naïve students may copy, unintentionally or otherwise. Advisors need to ensure that their students understand what plagiarism is and that their material is original. All of the authors are responsible if published material turns out to be plagiarized.

Self-plagiarism

Authors who re-use their own text may well be plagiarizing. Using the same text in two papers is a step in the direction of publishing the same work twice.

Some scientists feel that it is acceptable to re-use their own background material from paper to paper. A series of papers may be based on the same ideas or previous work, and—it might be argued—rewriting the background each time is pointless. However, there are both principled and pragmatic arguments against this practice. First, if an author is in the habit of copying the background in each paper, the material is likely to rapidly become stale, and authors who adopt this practice often seem unwilling to adapt the material even for papers on a different topic; in contrast, the discipline of writing new text each time helps to keep the material fresh. Second, a high-quality discussion of background material or of competing proposals adds weight to a paper, and increases the chance of it being accepted; by copying, the author is obtaining credit for old work. Third, some scientists view any significant re-use as improper, and authors presumably do not wish even a minority of their colleagues to view them as lazy or unethical. Fourth, most researchers work in teams of shifting membership. The authors of a paper collectively own its text; for some of the authors to take text and re-use it is inappropriate. The safe approach is to write fresh text for each new paper.

Publication of more than one paper based on the same results is prohibited under the standard scientific codes of conduct. An exception is when there is explicit cross-referencing, such as by reference to a preliminary publication from a more complete article that is a later outcome of the same research. (This is the one instance in which significant re-use of text can be acceptable.) Simultaneous submission to more than one journal or conference of papers based on the same results should be disclosed at the time of submission; the usual response to such a disclosure is to reject the paper. In this context, "the same results" does not necessarily mean a particular experimental run; if an experiment has been tried on some data, running the same experiment on other data is not new work unless it leads to new conclusions.

In the context of plagiarism and self-plagiarism, remember that publications are a permanent record. It may well be that a researcher successfully publishes the same results twice, or publishes a series of papers with figures and text in common, and in so doing rapidly develops an impressive publication list. But as time passes it is increasingly likely that such abuses of the system will be noticed, and there is no statute of limitations on plagiarism. The zeal of young researchers to publish should not blind them to the possibility of disciplinary action years or decades in the future.

Self-plagiarism can also be considered from the point of view of copyright. As noted earlier, when you publish a research paper the copyright is usually assigned to the publisher, who thus owns, not the ideas, but your expression of them. An author who re-uses more than a couple of paragraphs or a figure requires the publisher's permission to do so.

Once a paper has been assigned to a publisher, permission may also be required, in principle at least, to self-publish your work on the Web. However, such publication is widely practiced, and is consistent with the expection in many countries that the outcomes of publicly funded research be made freely available.

Misrepresentation

Misrepresentation occurs when a paper does not accurately reflect the outcomes that were observed or the contributions of previous research. When presenting results, researchers are expected to ensure that they are accurate, describe any experimental issues or limitations that could have affected the outcome, provide enough detail to enable reproduction or verification, be fair in description of other work, report negative as well as positive results, not state falsehoods, and take the effort to ensure that statements are complete and accurate. However, an honest mistake is not misconduct.

In its clearest form, misrepresentation is fraud: the making of claims that are outright false. Another form of misrepresentation is when authors imply that they have high confidence in their results when in fact the experiments were preliminary or were limited in some way. For example, reported running times may be based on a small number of runs with high variance, or there may be uncertainties about the quality of the implementation. Even more dubious are cases where the efficiency of a method being tested is based on some parameters, and the reported times are those achieved by tuning the parameters to the input data. Failure to report relevant unsuccessful experiments is explicitly condemned in the academic codes of conduct.

That is, there is a grey area between work that is fraudulent and work that is sloppy. Choosing a poor system as a baseline might just be lazy. Badly implementing a baseline—and thus exaggerating the benefits of a new approach—without verifying that the baseline works as well as was previously reported, begins to look more like deception.

Other forms of misrepresentation concern interpretation of past work. A behaviour that is far too common—and, arguably, is fraud—is to understate other people's work. It can be tempting for authors to exaggerate the significance and originality of their results, and to diminish the status of previous results in the field, to increase the likelihood of their work being published. If you would be uncomfortable defending what you have written about other people's work, then your text should probably be changed.

The issue of misrepresentation arises with online publication. When an author discovers an error in an online paper it is all to easy to correct it silently, with no explicit indication that the paper has changed. Modifications to online papers should always be made explicit, by use of a version number and date of publication; and the original version should continue to be available, as others may have referred to it. Online archives provide versioning functionality; but simply placing an updated version of a published paper on an archive does not adequately advertise that the paper has been changed and what the changes are. Retrospective alteration of a document is not something that should be done lightly.

It is because of the possibility of misrepresentation that codes of conduct require that scientists and departments retain their research data. A typical requirement is that the data must be held for 5 years from the date of publication and must be accessible to other researchers. In computer science, a reasonable interpretation of this guideline is that it is necessary to keep notebooks, software, results, and descriptions of inputs—the material that establishes that the research took place with the claimed outcomes. Implementation of such guidelines is (at best) inconsistent, but a central lesson is that it is reasonable for other scientists to seek to view your experimental setup as reported in a paper.

Authorship

Deciding who has merited authorship of a paper can be a difficult and emotional issue. A broadly accepted view is that each author must have made some significant contribution to the intellectual content of the paper. Thus directed activities such as programming do not usually merit authorship, nor does proofreading. An author should have participated in the conception, execution, or interpretation of the results, and usually an author should have participated to some degree in all of these activities. The point at which a contribution becomes "significant" is impossible to define, and every case is different, but neither code-cutting under the direction of a researcher nor management roles such as obtaining funding justify authorship. Nor is it appropriate to give authorship as a reward or favour.

A researcher who has contributed to the research must be given an opportunity to be included as an author, but authors should not be listed without their permission. On the other hand, involvement in an extended project does not guarantee authorship on every paper that results from the project. Contributors who are not authors should be acknowledged in some way.

Papers that are generated during the course of a student's research program are often jointly attributed to both student and advisor. Usually the student has undertaken the bulk of the task: capturing some idea in writing, running experiments, and locating background literature, for example. However, often the work would not have reached a reportable outcome without the involvement of the advisor. When students work independently, the research is theirs alone, but a student who has put in the majority of the effort while working under supervision should remember that it is intellectual input that determines authorship. An advantage to inclusion of the advisor as an author is that the advisor is committing to responsibility for the quality and originality of the work.

It is not appropriate for an advisor to publish the work of a student without the student's permission; if the student has completed a thesis reporting some research results, then the student has earned authorship on papers derived from these results. Nor is it appropriate for the student to publish without the permission of the advisor.

A related issue is that of author order, since readers may assume that the first author is the main contributor. A researcher who is clearly the main contributor should be listed first—don't believe Alfred Aaby when he tells you that alphabetic ordering is the norm.

Confidentiality and Conflict of Interest

Researchers need to respect each other's privacy. Sharing of a computer system with other people does not mean that one has the right to use their data without permission, for example, or to disclose their results to other people. Code or executables may be made available under terms such as commercial-in-confidence, and the fact that many people use commercial software they haven't paid for does not mean that it is appropriate for researchers to do so. (Use of commercial software presents challenges for reproducibility of the work by other people, who may lack the required licenses or the resources to acquire them, which is a reason to prefer open-source software in experimental work.)

Commercial relationships may need to be disclosed to editors or in the text of a submitted paper. Researchers who are publishing work on products or technologies should not conceal their involvement with the companies that own these products.

Another area where there is potential for conflict of interest is in refereeing of papers and grant proposals, and examination of theses. Researchers should not referee a paper where there is a possible conflict of interest, or where there is some reasonable likelihood that it will be difficult for the referee to maintain objectivity; or even where others might reasonably suspect that the referee would be unable to maintain objectivity. Examples include papers by a recent advisor, student, or co-author of the referee, or an author with whom the referee recently had close interaction, including not only personal or employment relationships but also situations such as competition for an appointment. In such cases, the referee should return the paper to the editor (and explain why).

It can be difficult to maintain objectivity if the author's opinions strongly conflict with your own. Make every effort to be fair, or seek an alternative referee. Also, your evaluation should be based on the paper alone; don't be swayed by the stature of the author or institution. Perhaps the trickiest case is that of a paper replicating your current work, or worse, is a faulty version of work you are currently doing but illustrates that you have made mistakes too. Probably the only solution is to contact the editor, state the case, and seek guidance. Whatever you do, act quickly; delay hurts the author.

A related issue is of confidentiality: papers are submitted in confidence and are not in the public domain. Papers you are reviewing should not be shown to colleagues, except as part of the refereeing process; nor should they be used as a basis for your own research. In practice there is something of a grey area—it is impossible not to learn from papers you are refereeing, or to ignore the impact of their contents on your own work. Nonetheless, the confidentiality of papers should be respected.

An "Ethics" Checklist

- Are you familiar with your institution's code of conduct?
- Have any authors been listed without their knowledge?
- Have other potential authors been omitted? Do they know that publication is proceeding without them?
- Was an ethics clearance obtained for any human studies? Can readers access the protocol that was approved in the clearance?
- Do you have any conflict of interest in publication of the work?
- Is the scope of citation and attribution clear? Is there a clear distinction between new work and previous knowledge?
- Are other papers accurately described?
- Has other work with similar results been appropriately cited and discussed?
- Is all the text in the paper yours?
- Are you the copyright holder for all figures and illustrations?
- If any material is shared with another paper, has the sharing been explained to the reader? Has it been explained to the editor?
- Does the paper include material recycled from your earlier work?
- Do you know which version of the code was used to run the experiments? Could you run the experiments again and get the same outcome?
- How will data and code be retained?
- Are there any weaknesses or limitations in the experiments that need to be described? Would you be prepared to show other researchers the raw experimental materials?
- Has any confidential or proprietary code or data been used? Do you have appropriate permissions?
- Is any of the content confidential?
- Are any claims overstated?

Afterword

Not all those who wander are lost.
J.R.R. Tolkien
The Fellowship of the Ring

Ready, set, go.
Schoolyard expression

The only way to produce a well-written paper is to start early and revise often. Write about what you plan to do, or what the project will be, or related literature, or anything of relevance. A researcher who argues that it is not yet time to start writing is mistaken: once you have a topic, you are ready to go.

Every stage of research benefits from writing. Once you have described your project, it is easier to ask skeptical questions about the direction and aims. Describing a project forces you to analyze it, and fruitful research directions may suggest themselves. Sketching an algorithm can highlight the fact that you do not yet understand some of its properties. A description of experiments allows examination of whether they are consistent and complete.

Procrastination is the enemy of good writing. There are always plenty of things you *might* do first—whether they are sufficiently important is another question. To do good science, it is necessary to write. Start now.

© Springer-Verlag London 2014

J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9

Exercises

The skill of good writing is acquired through practice. Pushing yourself, deliberately testing your ability to write new kinds of material and to write faster and better, can make a remarkable difference to the ease with which you can create polished text. Below is a series of exercises, intended not just for novice writers but also to help more experienced writers test and maintain their skills.

Some of these exercises are self-contained; others will be most helpful if adapted to your area of research, in particular by involving papers or passages that are relevant to your work. Educators may wish to choose standard papers and passages to be used by their students.

These exercises require substantial effort to complete—don't expect to run through one or two in a few spare minutes. Set aside a block of time that will be free of interruptions, say two hours, and in that time aim to do one exercise well. The exercises are loosely ordered by the kind of activity they involve, so if you only do a few, choose them carefully.

1. Choose or invent a research topic; this could be related to your research project, or could be an area that is of interest to you. Preferably, the topic should be reasonably easy to explain, and you should already have some technical knowledge of it. Illustrations of possible topics include "query suggestions in Web search", "vision processing for obstacle detection in domestic robots", "compression of high-resolution MRI images", or "dictionary structures for efficient spelling correction".

   Considering this topic, develop a precise hypothesis and research question.

2. Considering the research topic from question 1, use standard search mechanisms to find, say, twenty papers that from their titles appear highly relevant.

   - Label them by type and publication venue, for example as: journal paper, conference paper, survey paper, extended abstract, open-access, unreferred; some papers may have multiple labels. Quantify them by number of citations (and remember that search tools tend to prefer highly-cited papers).

© Springer-Verlag London 2014

J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9

- Use the papers to *map* the field. Are particular institutions or authors prominent? Which venues or publication types are most significant? Is the field "hot" (that is, the papers are all recent), or spread out over time, or some other pattern? What other features are valuable in mapping?
- Consider how these papers might be organized into a literature review. How should they be grouped?
- Compare the abstracts. Which ones are informative? Which ones can be read quickly? Are there particular venues that seem to have good abstracts?
- Using the abstracts from these papers, reconsider the research question you developed. Is it overly ambitious? Is it clear? By comparison to the existing work, where might a paper on your research question be submitted?

3. Considering your research topic from question 1, write a five-sentence abstract using the guide to abstracts in Chap. 5.

4. Choose a paper from your research area that you have not previously seen. Read the abstract and the conclusion. How are they related? Are there details in the conclusion that would be appropriate for the abstract?

5. Choose a paper from your research area and write a brief answer to each of the following questions.

   (a) What are the researchers trying to find out?
   (b) Why is the research important?
   (c) What things were measured?
   (d) What were the results?
   (e) What do the authors conclude and to what factors do they attribute their findings?
   (f) Can you accept the findings as true? Discuss any failings or shortcomings of the method used to support the findings.

   (These questions are not just an exercise: to some degree you should ask them for every paper you read.)

   Justify your opinions as carefully as you can. As part of the answers to these questions you should summarize the proposed method and the results achieved. The answers should be substantially your own writing, not quotes, paraphrases, or illustrations from the paper.

   Alternatively, use the questions in Chap. 3 to assess the paper.

6. Choose a paper, perhaps the same paper as for Exercise 5, and criticize the structure and presentation.

   (a) Is the ordering (of sections and within sections) reasonable?
   (b) Are sections linked together?
   (c) Does the paper flow? Are important elements appropriately motivated and introduced?
   (d) Where is the survey?

(e) Is there a non-technical introduction?
(f) How carefully has the paper been edited?
(g) Are there aspects of the presentation that could be improved?

Based on your criticism, write a referee's report for the paper, including a recommendation as to whether to accept or reject. Take care to discuss all of the paper's major problems.

Now read your review as if you were the paper's author. Is the review fair or harsh?

7. Some journals have special issues of a series of papers on a related topic. Choose two (or more) papers presenting a similar kind of result and compare them. Have the authors designed and organized the papers in the same way? Where the design choices differ, is one of the alternatives preferable?

8. Some of the papers in the *Communications of the ACM* argue for a point of view rather than present technical results; for example, there are often papers on legal or ethical issues or about computing practice. Choose such a paper and answer the questions in Exercise 5. Carefully analyze the argument used to defend the author's opinion, identifying the major steps in the reasoning. Are the conclusions sufficiently justified?

9. Choose a paper with substantial technical content from *ACM Computing Surveys*.

   - Browse the paper to get a sense of the content and how it is arranged. Then answer some questions about it; What is the main taxonomy or other structure that the authors use to organize the papers that are discussed? How are figures and tables used in the paper to support this structure?
   - Identify instances where a claim is supported by a citation; and see if there are instances where a counter-claim is supported by a citation.
   - Is the paper informative? How helpful is the structure?
   - In many survey papers, the authors are placing their own work in the context of other research results in the area. Do you regard the survey as fair? That is, is the survey an unbiased reflection of the relative strengths of the work in the area?

10. Choose a journal paper presenting new technical results. Based on the content of the introduction—you should not read the rest of the paper—do the following tasks.

    (a) Identify the hypothesis.
    (b) Suggest a suitable methodology for testing the hypothesis.
    (c) Suggest an organization for the paper, with headings and specific suggestions for the content of each section.

    Now compare your proposals to the body of the paper. Where there are differences, decide which alternative is better. The authors had much more time to think about the paper than you did, but are there any problems with the original organization?

11. Find examples of slide decks, such as lecture notes, that each treat the same topic. (For instance, on the Web there are many introductory slide decks on topics such as recursion, different advanced data structures, or topologies for cluster computers.) Examine them from a range of perspectives: the slide design and font choices, the kinds of pictures used as illustrations, the extent to which they are cluttered or pleasing to look at, and so on. Contrast them: which elements work best? What could be adopted for your next talk?
12. Choose a straightforward paper that you are familiar with, and draft a five-to-ten minute presentation explaining the problem, findings, and results. Set it aside for a week, then return to it; what would you change?
13. Browse journal papers to find a figure that you find clear and helpful, but which has a reasonable amount of detail; see for example the network diagram in Fig. 11.8. Use a line-drawing tool to imitate the figure as closely as you can. (If your imitation is of lower quality, you may need to find another tool.)
14. Download a large data set and explore it using a tool such as R (which was used for the graphs in the third edition of this book). Create:

    - An error-bar plot showing the mean and standard deviation for different variables.
    - A histogram visualizing the distribution of values for a single variable.
    - A scatter plot visualizing the distribution of values for one variable versus another.

15. Summarize a passage, perhaps the introduction of a paper, by jotting down the important points. These notes should be as brief as possible. Now write your own version of the passage using only your notes, without reference to the original. (Mary-Claire van Leunen attributes this exercise to Benjamin Franklin.)
16. Choose a popular article about computer science (from *Scientific American*, say) and summarize it in 500 words. Put the summary aside for a day or two, then review it. Did you include all the important details? Have you represented the article fairly? Would a reader of the summary arrive at the same conclusions about the work as a reader of the original article?
17. Iteratively edit a passage to reduce its length. Start with a passage of, say 300 words, then reduce it in length by 10 %, that is, about 30 words; then reduce by a further 30 words; and so on, for at least seven iterations. (To make this exercise more challenging, reduce by *exactly* 30 words at each step.) The aim at each step is to preserve the information content but not necessarily the original wording. Consider the resulting sequence of versions. With this exercise it is not uncommon for the passage to improve during the first couple of iterations, then become cryptic or incomplete as the text becomes too short for the content. Rate the versions: Which is best? Which is worst?

18. Rewrite the following passage to make it easier to understand. You may find it helpful to introduce mathematical symbols.

    > The cross-reference algorithm has two data structures: an array of documents, each of which is a linked list of words; and a binary tree of distinct words, each node of which contains a linked list of pointers to documents. When a document is added its linked list of words is traversed, and for each word in the list a pointer to the document is added to the word's linked list of documents. An order-one expansion of a document is achieved by pooling the linked lists of document pointers for each word in the document's linked list of words.

19. Choose a passage of 1,000 words or so, either a piece of your own work or any passage that you understand well. Revise it to improve the writing—that is, edit for flow, expression, clarity, and so on. Make the changes on paper, then type up the result, retaining the paper copy as a record.

    Put the revised passage aside for a few days, then repeat the exercise. Aim to make significant further improvements. (Did you undo any of the previous changes?) Revise again after a break of a few days; and continue until you have five revisions in total. Such revision is the best way to learn how to produce really good text, and many of the best writers revise this thoroughly.

20. The following fragments are flawed. They are ambiguous, or inelegant, or do not parse, or do not make sense. For each fragment, identify the problems—many of them have multiple shortcomings—and suggest revisions. If you need to make assumptions, state them clearly. (Most of these examples are from papers.)

    (a) As search engine systems emerge as the principle information finding tool within commercial enterprises due to the enormous popularity of WWW technology, the lack of options for integrating text and relational data on the Web is becoming crucial.
    (b) Information retrieval systems appear in the Web with the purpose of managing, retrieving and filtering the information available there.
    (c) The first approach is not practical. Thus the changes to the architecture of the system, including threads for the dictionary and client response components.
    (d) Concerning answer locality, usual tools tolerate lower first guess accuracy by returning multiple responses and allow the user to interact with the system to localize answers.
    (e) The difference in the previous results and the results from this study can be an artifact of the different collections that are being used in the two studies.
    (f) Authority work, the need to discover and reconcile variant forms of the same record will become more critical in the future.
    (g) The age of the mobile internet is dawning rapidly day by day and will demand more and more efficient solutions as disparate online resources are integrated in numbers of new ways.

(h) There are increasingly more online databases in the current climate of electronic publishing.
(i) There are several challenges to be associated with the data management of this information because the associated databases are highly multidimensional and dynamic.
(j) Ambiguity resolution was investigated by Klein [4]. Reverse parsing was shown in [4] to be a better method.
(k) Costing was performed on each option.
(l) The method, to be chosen is active mapping, as it is definitely superior in each experiment.
(m) One of these tools is one which automatically creates a short version which contains as much of the content as possible as the original.
(n) To compute whether the expected performance is achieved in a way that is automatic the only difficulty is to have a definition of similarity that is consistent with the user's perception.
(o) An effective alignment method that employs dynamic programming is presented to locate optimal points of match between the original text and the optically recognized version provided.
(p) An important phase of any system development process is the evaluation phase.
(q) It is also of interest how well the terms reflect the content of the indexed document as it is well known that assessing the quality of manual keywords is difficult, due to the fact that there is no general correct set of keywords for any given document and the preferred terms may vary from task to task, user to user, and even system to system, depending on the factors to be considered such as retrieval mechanism and search context.
(r) There are some audio-visual speech recognition systems that processes both the audio and visual channels, and complete recognition in real-time.
(s) The sudden growth of the WWW observed over recent times has triggered a lot of research fields to occur, Web services being only one of them.
(t) Association rules are rules that identify associations between items in transactions.
(u) A number of software packages exist, which are capable of designing relational models online.
(v) Most of today's complex systems are based on a hardware architecture that makes a physical separation of memory and processing and a software architecture that divides functionality into a hierarchy.
(w) The rest of this paper is organized, as follows.
(x) Given a range of options usually people are more interested in the extremes than in the middle part of the range since the two ends are more distinctive.
(y) Given a set of reference points, or cluster centroids, for a vector space and a quantization rule that provides a mapping to no more than 2^b distinct values then a filtering method consists of no more than building an injection from each site in the vector space to a binary signature which is just the concatenation of the binary expression of the quantized values.

(z) There are many applications, however, whose needs relational database systems do not meet, including diverse applications such as geographical information systems, CAD/CAM systems, expert systems, and the new kid on the block, text retrieval systems. And although not common today, text retrieval systems will undoubtably propagate as paper technologies such as offices and libraries are automated and the volume of text available in electronic form to the average user grows far beyond what it is today, already much more than it was in the recent past. Text retrieval is not well served by the current generation of database systems, despite the improvements they represent over earlier network, hierarchical, and file systems. Ironically, relational systems have only superficially incorporated text support, while the many purpose-built text retrieval systems usually don't support other kinds of data, or even complex forms of text, that might well be useful and important in some applications.

21. Typeset the following mathematical expressions.

(a) $\hat{\beta}_0 = \frac{\sum y_i - \hat{\beta}_1 \cdot \sum x_i}{m}$
(b) $y = \beta_0 + \beta_1 \cdot x$
(c) $\sum_{j=1}^{k-1} v_j < x \leq \sum_{j=1}^{k} v_j$
(d) $b = \left\lceil \frac{\log(2-p)}{-\log(1-p)} \right\rceil$
(e) $f(x) = e^{2^{g(x)}}$ where $g(x) = -\frac{b}{a}x\sqrt{1 - \frac{a^2}{x^2}}$
(f) $\hat{\beta}_0 \pm t_{\omega/2,m-2} \cdot \hat{\sigma} \cdot \sqrt{\left\{ \frac{1}{m} + \left(\frac{\sum x_i}{m} \right)^2 \cdot \frac{1}{\sum x_i^2 - (\sum x_i)^2/m} \right\}}$

22. Revise a mathematical argument to use less mathematics and more explanation. In a paper with a long proof or mathematical argument, identify the pivotal points of the argument. Is the argument complete? Are too many or too few details provided?

23. Choose a simple algorithm and a standard description of it, such as linear search in a sorted array. Rewrite the algorithm in prosecode. Repeat the exercise with a more interesting algorithm, such as heapsort. Now choose an algorithm with an asymptotic cost analysis. Rewrite the algorithm as literate code, incorporating the important elements of the analysis into the algorithm's description.

24. Design an experiment to compare two well-known algorithms for solving some problem. An elementary example is binary search in an array versus a hash table with separate chaining, but a more sophisticated example such as a comparison of sorting algorithms will make the exercise more interesting.

(a) What outcome do you expect—that is, what is the hypothesis?
(b) Will successful results confirm an asymptotic cost analysis?
(c) What resources should be measured? How should they be measured?
(d) What are appropriate sources of test data?

(e) To what extent are the results likely to be dependent on characteristics or peculiarities of the data?
(f) What properties would the test data have to have to confound your hypothesis?
(g) Is quality of implementation likely to affect the results?
(h) In the light of these issues, do you expect the experiment to yield unambiguous results?

25. Consider a research activity that is straightforward to describe such as "improvements to a robot's vision system for detecting obstacles". The goal of this exercise is to develop a research plan for evaluating whether the research activity is successful.

    (a) Describe in your own words the research ideas to be tested, that is, the hypothesis and research question.
    (b) Identify the criteria (qualitative aims) by which an improvement to the system would be regarded as successful.
    (c) Identify specific measures (quantitative aims) to be employed.
    (d) Consider all of: initial observational work; preliminary experiments; and large-scale trials.
    (e) Explain the steps involved in testing the research ideas, such as:
        - software to be implemented,
        - data to be collected,
        - comparisons to be made,
        - properties or characteristics to be observed or measured,
        - baselines to be used,
        - processes to be followed,
        - experimental methodology.
    (f) List variables that may influence the outcomes of the experiments, and how these influences might be controlled.
    (g) Identify possible confounds (explanations for positive or negative results that may be unrelated to the algorithmic improvements) and identify methods for eliminating these confounds.
    (h) Explain how these experiments relate to the aims of the research.

26. Write a program to find out how likely a tennis player is to win a match. (See Chap. 15.) How many matches are needed to converge on the result to a reasonable level of accuracy?

    Suppose a tennis tournament is to be played under the usual rules: players who lose a match are immediately eliminated, producing rounds in which the number of surviving players is successively halved, starting initially with 128 participants. Suppose all the players are equally good, with one exception, the champ, who wins 60 % of the points. What is the likelihood that the champ wins the tournament?

Suppose instead that the players are ranked from 1 to 128, and that player n wins 51 % of the points against player $n + 1$. Can the probability of the top-ranked player winning the tournament be investigated with the same method? Explain.

27. Choose a well-known researcher and identify the area in which the researcher is expert. Using the Web, identify other researchers in the same area. Which of these researchers might be regarded as authorities? Which are the key papers in the area? What evidence did you use to make these judgements?

28. The following bibliography has several faults and inconsistencies. Identify them.

    T. Cornish and J. Warren, "Networks without wires", 16(3):11–17, 2001.

    Frank Dean, "Wireless transaction resolution with do-nothing devices", *International Journal of Portable Computing*, 2(1):75–81, 2003.

    L .T. Lee, B. Clarke, and C. C. Cheng, "Systems analysis and systems design in Mobile Databases", *Jour. Portable Computing*, vol. 2, pp. 72–74, May 2003.

    Macic, V., et al., "Connectedness in low-bandwith local area networks", *Proc. International Mobile and Wireless Computing Conference*, Toby Thomas (ed.), ICM, June 2002, pp. 166–176.

Index

© Springer-Verlag London 2014

J. Zobel, *Writing for Computer Science*, DOI 10.1007/978-1-4471-6639-9

E

F

M

N

O

P