Scientific Communication CITS7200

Computer Science & Software Engineering

Lecture 6

Writing a paper

It is generally considered the case amongst academics and researchers that if you do not publish your work, you have not done it! Thus you write papers to communicate your ideas and discoveries — writing is a central part of your research work and it should form part of the *process* of doing research.

You should write every day: take notes on the papers you read, write down the details of the algorithms you implement, and tabulate the results of the experiments you perform. All these little scribblings will form the foundational material from which will eventually emerge your research paper.

That being said, you can be guaranteed of one sure fact: no-one really wants to read your paper. It is your task to make them want to read it. If you can write up your work in well-structured English, and you are capable of transferring your enthusiasm for your work to others, then you stand a better than average chance of getting your work communicated to others.

Generally, you will have only one chance to get someone to read your paper. There is so much published nowadays that most researchers have time for little more than browsing through the indexes of the multitude of journals that appear monthly. Having a concise and relevant title, a clear and inviting abstract, and a visually well-organised paper will enhance your prospects for a wider than usual audience for your work.

1 Organisation

As I've mentioned before, it is important to have a clear idea of your target audience. This will help you structure the organisation of your paper, and ensure that you choose the appropriate language and notation. Needless to say, you need to have done your research work *before* your start writing your paper. Thus you should have a fairly good idea about what it is you want to say. Before you start writing, rank your contributions. Work out what you consider to be your most important point, and what other results support your contribution. This will help you decide how to present your work, where to place the emphasis, and might even help with the title and the abstract.

You should have some initial idea of the length of your paper. The longer it is, the more important it is that it be well organised. You should aim to minimise length by avoiding repetition, to achieve general results that provide others as special cases (but the special cases are likely to be the most interesting), and to emphasise similarities and differences between different approaches to your topic.

If you were writing crime fiction, you could consider the ways in which "clues" are hidden from the reader; all these techniques should not only be avoided in writing scientific articles, they should be employed in the opposite sense.

Thus, the best way of tricking readers into seeing but not seeing what you put in front of them is by stating your fact in a way that seems clearly to be doing so for another purpose. Thus, if the low-order complexity of your algorithm relies upon the symmetric nature of your data, and yet you present this symmetry in your data as a reason for its elegant and simple graphical representation, then its role in the complexity argument will probably be lost to the reader.

A somewhat cruder way of concealing a crucial clue is by placing it somewhere in a long list of irrelevant facts. For example, not only is your data two-dimensional with integral values, it is also generated from a non-linear equation that describes the crystal growth of certain body sugars, it is symmetric, isotropic, generates a snow-flake fractal pattern, and can be displayed at a variety of scales.

Yet another trick of concealment relies upon "stock responses" or unthinking reactions to common facts. This is a difficult one to avoid in scientific writing, because it relies on you being able to move away from your own interpretations of the facts and to see how they might look to a naive reader. You might describe your data as symmetric: the stock response is that it is mirror symmetric about a central axis, when what you really mean is that it is symmetrically distributed about some arbitrary curve. These unthinking reactions are re-enforced if the clue is separated into two parts, either spatially or temporally. Perhaps you mention the underlying curves in your data in the introduction, or note in your literature review that they were first discovered in 1923, but then do not mention their relation to the symmetric data distribution until later in the paper.

Another technique, drawn from psychological evidence, relies on the fact that when observers are asked to watch out for something, they seldom remember what immediately preceded or followed that something. Thus, if you ask the reader to watch out for symmetry, and later on when symmetry is mentioned in your text you immediately precede this with the vital piece of information concerning the finiteness of your data, that clue will probably be overlooked.

In scientific writing, the "clues" must all be presented clearly, together, up front, and in a way that makes the deduction that is drawn from them completely obvious.

2 Title

The title is the single most important part of your paper and will directly determine the number of readers. Most readers of your paper will not get past the title, and if they do, it is probably the abstract and the reference section that will be looked at next.

Thus the title represents the *highest level of abstraction* of your paper—it captures the essence of your paper. The title should give a concise description of your paper; it is the content of your paper captured in one phrase. It also has to be catchy enough to attract the attention of the casual browser.

Do not make your title too vague. Avoid phrases like *A note on, Some results on,* etc. Here are some real titles:

- *Linear Scheduling is Nearly Optimal.* Titles that are complete sentences, and that summarise your paper, are excellent.
- *Learning the Unlearnable.* This is a very punchy title and would certainly attract the casual browser to at least read the abstract to see what it is about.
- *An application of the multiedit-condensing technique to the reference selection problem in a print recognition system.* This title is too vague. What application? What is the reference selection problem? The

word "multiedit" should normally be hyphenated, but then "multi-edit-condensing" is cumbersome.

- *A Frame-based Computer Vision System.* Too vague.
- *An Initial View on Size Estimation for Expert System Applications.* Sounds like unfinished work.
- *Active Intelligent Vision using the Dynamic Generalised Hough Transform.* Too many adjectives.
- *RAPID - a video rate object tracker.* Catchy, and to the point.

3 Authors

After the title comes the author list, with affiliations. Always use the same declaration for each paper you write. For example, I should not appear as R. A. Owens on one paper, then Robyn Owens on another, then Robyn A. Owens on another. This makes bibliographic and citation searches difficult, since all your papers might not be grouped together in such databases.

There are no hard and fast rules about the order in which authors should be listed when there are more than one. Some disciplines, and some journals, always list authors alphabetically. In some publications, it is the author that has played the principal part in the publication who comes first; sometimes it is the more senior author. However, being the first-named author is very advantageous, as it is that name that appears in any citations in the form of "Bloggs et al.", and many citation services ignore all but the first name.

Always give your affiliation. For example,

Amanda Lin
School of Music,
The University of Western Australia,
35 Stirling Highway,
Crawley, W. A. 6009.
E-mail: `amanda@music.uwa.edu.au`

Always include the name of the University, and always include your email address as part of your affiliation. Most readers would then use this to contact you if they wished to pursue further communication on your paper.

Always date your work. If your pass around unpublished manuscripts to others, they will want to know when the work was done. Dating is important for establishing precedence when many researchers are working in the same field. Also, you can confuse different drafts if they are not dated. The date is usually placed on a line by itself after the author field.

4 Abstract

The purpose of the abstract is to summarise the contents of your paper. As such, the abstract represents the *second highest level of abstraction* of your paper. It should not be a perfect copy of the first paragraph of your introduction, nor should it replicate other parts of your paper word for word. The abstract, like the title, is a mini-paper, and its purpose is to inform the reader, in about 200 – 600 words, about the whole paper. This usually means that the abstract should be written from scratch, once the rest of the paper is completed. The shorter it is, the better. Most abstracts occupy a single paragraph.

Try to avoid mathematical equations in your abstract. Do not cite references by number, since your abstract will appear without the reference list in review journals or on CD ROMs that list paper abstracts. If you need to cite existing work, spell out the reference in full.

Make the abstract easy to understand by those whose first language is not English. Your paper, or at least your abstract, might one day be translated into a foreign language.

Your abstract should lay claim to some new results, unless your paper is a survey. However, do not make claims that are not justified by the content of the paper. If your abstract does not say what the new results are, and state them clearly, then you are making it easy for a referee to reject your paper for publication.

Try to avoid starting your abstract with the common but unnecessary phrases "This paper", or "In this paper".

5 Keywords

It is common to give a list of key words, usually 10 or less, that can be used in computer searches by someone looking for an article on a certain topic. Try to anticipate the words that a reader might search for, and make them specific enough to give a good indication of the content of the paper.

6 CR classification

The Computing Reviews Classification System is a four-level tree that has three numbered levels and an unnumbered level of descriptors. The top level consists of eleven nodes, denoted by letters A through to K. For example, level D is for Software, D.3 for Programming Languages, D.3.3 is Language Constructs and Features, and a full CR classification might be

D.3.3 [Software]: Programming Languages — Abstract data types.

It is common to give at least the first two top level descriptors with any work you write.

7 Introduction

The most important part of the core of your paper is the introduction, and this represents the *third highest level of abstraction* of your paper. It is in the introduction that you state your thesis or hypothesis. A thesis is a proposition laid down or stated as a theme to be discussed and proved, or to be maintained against attack. An hypothesis is a supposition or conjecture put forth to account for known facts, and it serves as a starting point for further investigation by which it may be proved or disproved. Through your compelling writing style and the force of your arguments, you must convince your reader of the validity of your thesis or hypothesis.

Many papers in computer science concern the experimental testing of an hypothesis, and much of what follows is relevant to that style of paper.

Your opening paragraph should represent an invitation to the reader, explaining why the hypothesis is a sensible thing to test. This can begin as a description of a current problem within your research domain and a description of why one might want to solve it. Your first sentence must carry some essential information, but at the same time gently introduce the reader to the rest of the article.

The second paragraph should be short and to the point. It should state clearly your hypothesis. Now the reader knows what the problem is, and what your hypothesis is for solving it.

Ideally, the introduction should be fairly short: explain the problem, explain what your hypothesis is for solving it, and outline your plan of attack. Unless there is a good reason not to do so, summarise the results that you have achieved. Now the reader can decide whether to read the rest of your paper or not.

Here is a good example from the *Journal of Visual Languages and Computing*:

> Graphical representations share a problem with textual ones: quality is not guaranteed. There are bad diagrams as well as good ones, just as there are both bad and good textual representations. Similarly, there are more or less gifted practitioners as well as more or less experienced ones, and these factors appear to affect performance in terms of comprehension and inspection strategy, leaving less experienced users unable to exploit (or even notice) the cues that might help them. So, despite the widespread wish (sometimes mistaken for fact) that graphics might be universally and instantly accessible, it appears that training and experience play a significant role in determining what is salient.
>
> This paper pursues the notion that training influences what 'readers' of graphical representations (such as graphical programming languages) look at and hence what they see, so that inspection strategies for graphical notations must be learned.
>
> We draw evidence from two sets of studies: observational studies of expert hardware designers using electronic schematics, and experiments comparing readability of textual and graphical programming notations. We conclude that graphical representations, especially those used in notation, far from being intuitively obvious, may require more training from both the originator and the reader to achieve the most effective communication - but may prove richer for expert users.

Unless the paper is very short, it is common to outline the organisation of the rest of the paper towards the end of the introduction. Do not simply list the section titles that follow; give a short summary that could only be obtained by reading the individual sections.

In terms of levels of abstraction, the conclusion has the *fourth highest level of abstraction* in your paper, and the remaining sections have the fifth, or lowest, level of abstraction. Thus, the sections that follow the introduction, up until the conclusion, need to get into the detail of your work.

8 Literature Review

Almost all scientific work is done in context; others have worked on the same or similar problems, and they have developed techniques for solving such problems that you will need to explain and compare.

The purpose of the literature review is to consider the many approaches that have been adopted to solve your problem, or similar problems, and to determine which of these approaches is the best for your current purposes. A scientific literature review is *not* a review on the author's writing style, not a review on the complexity of the ideas used in the paper. It is a review of the *science*. As such, it is inappropriate to say that a particular paper is badly written with many faults of grammar, or that although the ideas are interesting the mathematics is too difficult for the average reader. Both writing styles and conceptual difficulties are things you must get above when you write your literature review.

You must also rise above any individual author's notation. Often the papers you are reviewing use different notation, or the authors argue their case at different levels of abstraction. In your literature review you can introduce the notation and level of abstraction that you will use throughout your paper, and use this notation consistently across all the papers you are discussing.

9 Methodology

It is common to now develop the methodology of your paper. Some of this will have been done in your literature review, where you have introduced some notation and described in clear detail the techniques that others have used. Now you need to describe the technique that you use. Often in computer science, this will be a new algorithm you have developed, or a modification of an existing algorithm. It is not enough to simply list the steps in the algorithm; you must show that you algorithm is correct (that is, given appropriate input it terminates with appropriate output), and show, by proof, experiment, or both, that it meets some claimed performance bound.

Thus, the reader would expect to find throughout the rest of your paper 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, such as preconditions, postconditions, and loop invariants.
- A demonstration of correctness.
- A complexity analysis, for both space and time requirements.
- Experiments confirming the theoretical results.

The description of your algorithm should be given using mathematical notation, rather than programming notation. Thus, use x_i rather than `x[i]`, for example. Also, use standard mathematical operators and avoid expressions that are language specific, such as

```
==, a = b = c, a++, for (i=0; i<n; i++)
```

for example.

The algorithm should be described in sufficient detail to allow the reader to implement it without too much trouble or inventiveness. The validation of another's work is an important part of the scientific process. Part of your research will involve validating other's work, and others must be able to validate yours.

Try not to give your algorithm in pseudo-code. It is difficult to read and a poor way of explaining the ideas that went into the construction of your algorithm. Instead, use English prose to give the background to the algorithm, to introduce the notation, to explain what the major steps in the algorithm are, and to work through these steps in detail.

10 Experiments

Many papers in computer science describe computational experiments. This forms the *science* in computer science: that is, the iterative procedure of hypothesise, test, refine your model, and then repeat. Your experiments might be done simply to gain insight into someone else's model, or to compare two existing models. They might be to verify theoretical predictions, to tune algorithm or code parameters, or to measure the performance of your software. As such, all these experiments need to be carefully designed and meticulously carried out.

When you report the results of a computational experiment, you need to give enough information so that the reader can reproduce your experiment. In particular, you should consider stating the machine precision, the type of random numbers you use (if appropriate), the programming language, the version of the compiler, and compiler options and optimisations that were selected. All these points can significantly influence run-times.

To make your experiments reproducible, it is a good idea to use standard test data. In many areas, standard data exist. For example, standard test images for vision work, machine learning, and graphics can be obtained over the net.

Try to make a clear distinction between objective statements, and speculations. Objective statements are backed up by hard facts. Print out every conceivable statistic that might be of interest when you run your experiments, and use these results to help you formulate your conclusions.

11 Tables, images, and graphs

You will often want to present your results in the form of tables or graphs and, in vision work, images are imperative.

With regard to tables, it is easier to compare quantities that are arranged in columns, rather than rows. Consider the following:

complexity	terminology
$O(1)$	Constant
$O(logn)$	Logarithmic
$O(n)$	Linear
$O(n^b)$	Polynomial
$O(b^n)$	Exponential
$O(n!)$	Factorial

Table 1. Commonly used terminology for algorithm complexity.

Complexity	$O(1)$	$O(logn)$	$O(n)$	$O(n^b)$	$O(b^n)$	$O(n!)$
Terminology	Const.	Log.	Linear	Poly.	Exp.	Fact.

Table 2. Commonly used terminology for algorithm complexity.

Only essential information should be given in tables. Do not include more significant figures than are known for the data. Large amounts of data are probably better displayed in graphs. If a large amount of data must be displayed in tabular form, put it in the appendix.

Make your table design as simple as possible. In particular, minimise the number of rules in your table, even if it means putting some of your headings into the caption. Give a clear reference to your table at the appropriate place in the text, but do not summarise the whole content of the table, otherwise the table might as well be omitted.

When you are displaying graphs, make sure that the axes are properly labelled and that units are well-established. Sometimes 3-D data are better displayed as 2-D contour maps, rather than as surfaces. Try different perspective views to enhance the comprehension of graphical data.

When you are displaying images, use a good quality postscript printer. Both graphs and images should appear at their appropriate place in the text. LaTeX treats figures, tables and images as floating bodies, since these items cannot be broken up over a page. Because of this, LaTeX will often place them at a good distance from the textual description, which can be very frustrating. To overcome this problem, you need to include a *placement specifier* as an optional parameter to the `figure` or `table` environments. A placement specifier is constructed by building a string of float placing permissions, as listed below:

`h`	place item *here* in the text
`t`	place item at the *top* of a page
`b`	place item at the *bottom* of a page
`p`	place item on a special page containing only floats
`!`	override other parameters even if the result looks awful.

If no placement specifier is given, the standard classes assume `[tbp]`. A typical command of `\begin{table}[htbp]` usually works quite well.

All tables and figures should come with a caption, and the caption should explain the content of the table or figure clearly and independently of the text. Often a reader will skip the text and simply look at your figures; the point the figure is trying to make should be spelled out in the caption.

If your captions tend to be long, and you are using the commands `\listoffigures` and `\listoftables` then you will need to have a shorter version available for inclusion in these lists. This is done by entering the short version in brackets after the `\caption` command:

```
\caption[Short version]{A very long label to a figure or table}
```

12 Conclusion

If you have a conclusion section, it should not simply repeat earlier sections of your work. The purpose of a conclusion is to draw all your previous work together, point out new ways of looking at your results, discuss the limitations of your work, and to suggest further avenues of research.

13 Acknowledgements

It is customary to acknowledge any financial assistance you have had throughout the duration of your research. If you are working on a research grant, you might write something like "This work was supported by an ARC grant". If you are working on a scholarship you can write "This work was supported by an Australian Postgraduate Research Scholarship".

It is usual to thank anyone who has helped you in carrying out your research when that help is not just someone carrying out their normal working duties. This would usually include people who have read and commented upon earlier drafts of your work, or those who have contributed ideas to your research. Any special help by others in setting up, or providing you with, special software tools also deserves mention.

14 References

If you are writing a paper for a specific journal, check the Instructions to Authors for the format of the reference section. The details of compiling a reference section were dealt with last week, but this is one part of your research writing that should be done continuously as you do the research. Do not put papers in your reference section that are not specifically cited in the text. Do not rely on secondary sources; always find the original source when it is available to you. If you are referring to a technical report that is more than a few months old, check whether it has appeared since as a conference publication or in a journal. It is always best to give the bibliographic source that is easiest for the reader to find.

15 Appendices

An appendix contains information that is essential to the paper, but which would interrupt the flow of the text if it were placed in the body of the document. The most common use of an appendix is to list a detailed analysis of some mathematical equations, to tabulate large amounts of data, or to include program listings. An appendix should not be used to squeeze inessential information into a paper, nor should it hold information vital to the body of the text but placed in an appendix to comply with length restrictions on the body of the text.