Helpful Hints for Technical Paper Writing

Armando Fox

Note: This is a page about writing technical papers, but many of the techniques also seem applicable to both non-technical writing and giving presentations.

1 Vision, Implementation, and Survey Papers

In a *vision paper*, you describe your grand scheme of the world and why it is good. You need some data to back up your statements, but this is not a detailed measurements paper. The goal of this paper is to convince the reader that your scheme is interesting, different, better than other schemes that have addressed similar problems, raises legitimate research questions, and is therefore worth spending the time to pursue research on.

If you're writing a vision paper, you have to be absolutely convinced of your vision, or no one else will be. Make no statement that cannot be backed up by citation, quantitative data, or at least a very good first-cut experiment ("preliminary results suggest....")

The *implementation paper*, by contrast, gives detailed measurements of a system that was perhaps described in a previous vision paper. The goal here is to demonstrate what you learned from actually building the system: Did it validate your research hypothesis? What came out differently than you expected, and why? How much better, quantitatively, is your design than others'?

If you're writing a *measurement-and-results* paper, first determine which graphs will convey the results you think are important. Given those, the paper will practically write itself.

Survey papers: TBD...

2 Before You Write...

If possible, present your work in a short 5-10 minute talk to your colleagues before starting to write. This helps identify strengths and weaknesses and will give you an idea of what other people see as the important contributions. Surprisingly often, they will spot a significant contribution that you totally overlooked, or suggest a novel application of your ideas that dramatically increases the relevance and impact of your paper.

Don't cram. Recall the old saw about how nine pregnant women cannot produce a baby in one month. You can't throw all your time into a paper at the last minute and expect a good result: you will become saturated, lose perspective because you are too close to the material, and ultimately be spinning your wheels, changing stuff back and forth without a really good feeling for why you're doing it. Exceptions to this occur, but they're rare.

Know when to say when. Even if you have written the paper with plenty of time and had a lot of outside review, after a certain point you will not be able to add much value without taking a break for a while (maybe a week or two). When that point comes, further work on the paper is just thrashing and not likely to improve it much, though it will leave you feeling dissatisfied. Wait for the reviews from the PC before doing much more.

3   Starting Checks

Write from an outline. Let me say that again, because it's really important: write from an outline. I know no one who can reel off any coherent technical writing more than one page long without some kind of top-down strategy. At least sketch out the major sections of the paper, and what points you want to make in each, from 10000 feet. If you write any complete sentences during this phase, you're getting mired in detail already. Bullets are what you want.

Don't even try to write the title or abstract until after the whole rest of the paper is written. Then, and only then, will you actually know what the hell it is you want to say.

Unless you're writing a PhD thesis, your paper will make only a small number of discrete points–say 2 to 4. Each important point should appear 3 times: once in the abstract/introduction, once in the body of the paper (where it is explained in detail), and once in the conclusions (where you derive some implications of this point for the future of systems research, or whatever). Bulleted conclusions can help. Remember that conference referees are at least as busy as you and they have to read several of these. Make sure they remember yours.

4   The Actual Paper: Writing

1. Start from the outline.

2. Make the outline reflect the level of subsections: for each subsection, write no more than two lines describing the purpose/goal of that subsection. This text will NOT be part of the paper - it is only there to remind you what you are trying to accomplish. It is ESSENTIAL that you be able to capture the purpose of a subsection in one or two lines. If you cannot do this, then you probably don't understand what the subsection is really about, and when you try to write the text, it will be jumbled.

3. Then, for each subsection, map out specific paragraphs: for each paragraph, write one sentence that explains the topic or main goal of just that paragraph. Again, this sentence probably will NOT make it into the actual text. It's important to keep it to one sentence. (As every style manual will tell you, including Strunk & White, virtually every well-formed paragraph does indeed have one sentence that explains the point of the paragraph, with the other sentences supporting or expanding on the point of the topic sentence.) If you cannot fit the point of the pargraph into 1 sentence, the paragraph is probably making > 1 point, so it should be split into multiple paragraphs.

4. Read through everything you have written and see if it has a logical flow, ie if you believe it represents your work adequately.

5. Give what you have written to a technical colleague completely unfamiliar with your work (but able to understand the computer science part), have them read it, then have them tell you (without looking at it) what s/he thinks the main point and contributions are.

6. If all goes well, now replace the topic sentences with complete paragraphs.

This way of writing will not yield a Shakespearean work of literature, but it is consistent and will result in readable, logically organized prose by construction.

5 The Actual Paper: Revising/Editing

Your section organization will change. Sometimes it will be shuffled dramatically. This is fine; it means you're understanding what presentation order works best. If you don't go through at least three or four major revisions (where you move around or chop entire sections), it's probably lousy.

After doing some edits on each draft, give it a full top-to-bottom reading to evaluate its coherence and flow of ideas. Then, take a couple of hours and do something else; once you get close enough to your paper, you start missing the forest for the trees.

Even early drafts are valuable for getting your colleagues' comments. Get comments from people who you think may be skeptical of your approach. Get comments from people who will really rip your writing style apart. Remember, at least they are your friends; the conference referees probably are not.

Cite, cite, cite! Ask your colleagues for suggestions and pointers. You never want to be asked: "What about the work done by xxx, which obviously has something in common with your own?" (or worse: "...which refutes your own?") Give due credit to those whose efforts you build on, as well as pointing out how your approach is different from (and better than) previous ones.

6 About Writing

It's often said, correctly I think, that most technical people don't write well. This doesn't mean that they lack knowledge of grammar or spelling (though this is sometimes the case), but that they don't know how to organize their writing at the level of paragraphs.

Don't artificially formalize your writing style. Technical writing must be clear and concise. Overblown writing rarely fools anyone and it makes the paper boring to read.

- Bad: "Problem X is clearly a critical area that impacts our research agenda and hypothesis. Our ideas about problem X are embryonic and still evolving, and doubtless our ongoing work in this area will quickly yield fruitful results."

- Better: "We recognize that problem X is central to our agenda, but we have only begun to investigate it."

If you haven't read Strunk and White's *The Elements of Style*, read it now. If you have, read it again. If you can't organize a paragraph, you won't have much luck organizing a chapter.

Omit needless words. Don't be surprised if this turns out to be 30-40% of the words you originally wrote. Your first effort rarely captures the most vigorous or concise way to say something. Spend time tersifying.

Run your paper by someone who is anal retentive about grammar to catch common errors: misuse of which and that, non-words and non-phrases such as for all intensive purposes or irregardless, lack of parallel sentence structure...

7 Final Checks

Remember that this will be read by people who (a) have never heard of you and the review is anonymous anyway, (b) have never heard of your project, (c) are reading about 15-20 papers apiece, all in different subject areas. They will spend the first 5 minutes deciding if your paper is actually good enough to be worth a fully detailed read; they will then spedn an hour or so reading it in detail, trying to figure out (a) what your contribution is, (b) if the contribution is substantial enough to be worth publishing, (c) if the contribution is "feasible" (ie it is implementable and therefore would be useful to someone).

- Does the paper make clear precisely what your new contributions are, and how they are different/better than existing approaches to this or similar problems?

- Does the outline of the paper (sections, subsections, etc.) cohere regardless of the granularity at which you view it? (The Outline mode of MS Word is a valuable feature for this check. I also wrote a simple Perl script that does this for LaTeX files.)

- Have you observed the following invariant: Before telling me what you did, tell me why I should care.

- Have you made every important point three times–once in the introduction/abstract, once in the body of the paper, and once in the conclusions? (Bulleted conclusions are usually a good idea)

- Have you had it read by at least one person familiar with each of the areas the paper impinges on? (Think of them as consultants in that area. There is a risk that you will get some of the details wrong in talking about an area that is tangential to the paper but that you're not very familiar with, and if a reviewer happens to be versed in that area, it decreases your credibility. Such references are easy to get right, so there is no excuse.)

- Have you searched carefully for any related work, and properly acknowledged it? The availability of papers and search indices on the Web makes it worse than ever to overlook significant related work.

- Are you able to capture the non-experts in the audience with the opening of your paper, and impress the experts in the body of the paper?

- Can you read only the abstract and conclusions and be able to give someone else a 30-second digest of what the paper claims it says?

Appendix: Mike's comments

- Write from an outline as described above. I like to add one step: after the subsection-level outline is done, flesh out the *introduction* section with a paragraph-level outline and the *results* with a graph-level outline (that sketches each graph and states what point each graph makes.) Iterate on this a few times until you have these sections right.

- Talking someone else through the outline before you start writing is a really valuable step. We have found it really useful to put the outline up on an LCD projector with all authors in the room and to have someone talk through it while everyone looks for ways to improve the argument. This process improves the organization with much less effort than evaluating full prose drafts. Also, this approach helps all of the co-authors know how the piece they are writing fits into the whole.

- Be prepared to re-write a section from scratch. Multiple times.

- Once you have written the introduction, read the first (topic) sentence of each paragraph to see if you actually have made a coherent argument.

- "Rule number one is you must not fool yourself and you're the easiest one to fool." (Feynman) One of the most difficult things to do is to take a step back from something you have written and evaluate what will happen when a third party reads it. Learn to do this.