Writing a Technical Paper

Michael Ernst

Associate Professor, Department of Electrical Engineering and Computer Science,
Massachusetts Institute of Technology (MIT)

The goal of technical writing is clarity and understanding: the purpose is to communicate specific ideas, and everything about the document should contribute to this goal. If any part of the document does not do so, then delete or change that part of the document. This rule also indicates that you should know exactly what points your document intends to make; if you don't know the purpose, you cannot effectively achieve it.

Some people believe that writing papers, giving talks, and similar "marketing" activities are not part of research, but an adjunct to it or even an undesirable distraction. This view is inaccurate. The purpose of research is to increase the store of human knowledge, and so even the very best work is useless if you cannot effectively communicate it to the rest of the world. Additionally, even if you believe that you understand your ideas and contributions, you are likely to find that when you try to write or speak them, you are unable to clearly enunciate them. The process of clarifying your thinking, of which writing papers and giving talks is one aspect, is a valuable part of improving your research.

Organization

A paper should communicate the main ideas of your research (such as the techniques and results) early and clearly. Then, the body of the paper can expand on these points; a reader who understands the structure and big ideas can better appreciate the details. This advice also applies at the level of sections and paragraphs. Do not start with a mass of details, hoping that the reader will somehow figure out which of those are relevant to your main point, and only later tell the reader what the main point was.

For each section of the paper, consider writing a mini-introduction that says what its organization is, what is in each part, and how the parts relate to one another. For the whole paper, this is probably a paragraph. For a section or sub-section, it can be as short as a sentence. This may feel redundant to you (the author), but readers haven't spent as much time with the paper's structure as you have, so they will truly appreciate these signposts that help them orient themselves within your text.

Some people like to write the abstract, and often also the introduction, last. Doing so makes them easier to write, because the rest of the paper is already complete and can just be described. However, I prefer to write these sections early in the process (and then revise it as needed), because they frame the paper. If you know the paper's organization and outlook, then writing the front matter will take little effort. If you don't, then it is a better use of your time to determine that information (in the context of writing the front matter). To write the body of the paper without knowing its broad outlines will take more time in the long run.

A paper should never first detail a technique, then (without forewarning) indicate that the technique is flawed and proceed to discuss another technique. Such surprises confuse (and infuriate) readers.

Do not write your paper as a **chronological narrative of all the things that you tried**, and do not devote space to the paper proportionately to the amount of time you spent on each task. Most work that you do will never show up in the paper; the purpose of infrastructure-building and exploration of blind alleys is to enable you to do the small amount of work that is worth writing about. The audience is interested in what worked, and why, so start with that. If you discuss approaches that were not successful, do so briefly, with a focus on differences from the successful technique, and typically only after you have discussed the successful approach.

Write for the readers, rather than writing for yourself. In particular, think about what matters to the intended audience, and focus on that. Do not focus on what you personally find most interesting, or what you spent the most time on, or what was most difficult, or on implementation details. (This is a particularly important piece of advice for software documentation, where you need to focus on the software's benefits to the user, and how to use it, rather than how you implemented it. However, it holds for technical papers as well — and remember that readers expect different things from the two types of writing!)

Writing style

Passive voice has no place in technical writing. It obscures who the actor was, what caused it, and when it happened. Use active voice and simple, clear, direct phrasing.

First person is rarely appropriate in technical writing. First person should never be used to describe the operation of a program or system. It is only appropriate when discussing something that the author of the paper did manually. (And recall that your paper should not be couched as a narrative.) It is confusing to use “we” to mean “the author and the reader” or “the paper” (“In this section, we ...”) or even “the system being described” (“we compute a graph” makes it sound like the authors did it by hand). As a related point, do not anthropomorphize computers: they *hate* it. Anthropomorphism, such as “the program thinks that ...”, is unclear and vague.

Be brief. Make every word count. If a word does not support your point, cut it out, because excess verbiage and fluff only make it harder for the reader to appreciate your message. Use shorter and more direct phrases wherever possible. Avoid puffery, self-congratulation, and value judgments: give the facts and let the reader judge.

Do not use words like “obviously” or “clearly”, as in “Obviously, this Taylor series sums to pi.” If the point is really obvious, then you are just wasting words by pointing it out. And if the point is not obvious (readers won't be intimately familiar with the subject matter the way the author is), then you are offending readers by insulting their intelligence, and demonstrating your own inability to communicate the intuition.

Prefer singular to plural number. In “sequences induce graphs”, it is not clear whether the two collections are in one-to-one correspondence, or the set of sequences collectively induces a set of graphs; “each sequence induces a graph” avoids this confusion. Likewise, in “graphs

might contain paths”, it is unclear whether a given graph might contain multiple paths, or might contain at most one path.

Some of the suggestions in this document are about good writing, and that might seem secondary to the research. But writing more clearly will help you think more clearly and often reveals flaws (or ideas!) that had previously been invisible even to you. Furthermore, if your writing is not good, then either readers will not be able to comprehend your good ideas, or readers will be (rightly) suspicious of your technical work. If you do not (or cannot) write well, why should readers believe you were any more careful in the research itself? The writing reflects on you, so make it reflect well.

Figures

Use figures! Different people learn in different ways, so you should complement a textual or mathematical presentation with a graphical one. Even for people whose primary learning modality is textual, another presentation of the ideas can clarify, fill gaps, or enable the reader to verify his or her understanding. Figures can also help to illustrate concepts, draw a skimming reader into the text (or at least communicate a key idea to that reader), and make the paper more visually appealing.

It is extremely helpful to give an example to clarify your ideas: this can make concrete in the reader's mind what your technique does (and why it is hard). A running example used throughout the paper is also helpful in illustrating how your algorithm works, and a single example permits you to amortize the time spent explaining the example (and the reader's time in appreciating it).

Figure captions should give all the information that is necessary to understand the figure. Never write a caption like “The Foobar technique”; also say what the Foobar technique does or what it is good for. Many captions also need to explain the meaning of columns in a table or symbols in a figure. If the figure is not self-explanatory, then the reader is forced to hunt all over the paper in search of this information, which should have been in the caption (or in a legend), and which doesn't belong in the main text of the paper where it interrupts the flow of the writing. Figures that stand on their own are more effective at drawing in a reader who is scanning the paper.

As with naming, use pictorial elements consistently. Only use two different types of arrows (or boxes, shading, etc.) when they denote distinct concepts; do not introduce inconsistencies just because it pleases your personal aesthetic sense. Almost any diagram with multiple types of elements requires a legend (either explicitly in the diagram, or in the caption) to explain what each one means; and so do many diagrams with just one type of element, to explain what is happening.

I am not fond of having many different types of figures in a paper — some labeled “figure”, others labeled “table” or “graph” or “picture”. This makes it very hard to find “table 3”, which might appear after “figure 7” but before “freehand drawing 1”. It's best to simply call them all figures and number them sequentially; the body of each figure can be a table, a graph, a drawing, or whatever.

Your code examples should either be real code, or should be close to real code. Never use synthetic examples such as methods or variables named `foo` or `bar`. Made-up examples are much harder for readers to understand and to build intuition regarding. Furthermore, they give the reader the impression that your technique is not applicable in practice — you couldn't find any real examples to illustrate it, so you had to make something up.

Any boldface or other highlighting should be used to indicate the most important parts of a text. In code snippets, it should never be used to highlight syntactic elements such as “public” or “int”, because that is not the part to which you want to draw the reader's eye. (Even if your IDE happens to do that, it isn't appropriate for a paper.) For example, it would be acceptable to use boldface to indicate the names of methods (helping the reader find them), but not their return types.

Naming

Give each concept in your paper a descriptive name. Never use terms like “approach 1”, “approach 2”, or “our approach”, and avoid acronyms when possible. If you can't think of a good name, then quite likely you don't really understand the concept. Think harder about it to determine its most important or salient features.

Use terms consistently and precisely. Avoid “elegant variation”, which uses different terms for the same concept, to avoid boredom on the part of the reader or to emphasize different aspects of the concept. While elegant variation may be appropriate in novels or some essays, it is not acceptable in technical writing, where you should clearly define terms when they are first introduced, then use them consistently. The reader of a technical paper expects that use of a different term flags a different meaning; you will confuse the reader and muddle your point if you switch wording gratuitously. Don't confuse the reader by substituting “program”, “library”, “component”, “system”, and “artifact”, nor by conflating “technique”, “idea”, and “method”. Choose the best word for the concept, and stick with it.

Do not use a single term to refer to multiple concepts. If you use the term “technique” for every last idea that you introduce in your paper, then readers will become confused. This is a place that use of synonyms to distinguish concepts that are unrelated (from the point of view of your paper) is acceptable. For instance, you might always use “phase” when describing an algorithm but “step” when describing how a user uses a tool.

When you present a list, be consistent in how you introduce each element, and either use special formatting to make them stand out or else state the size of the list. Don't use, “There are several reasons I am smart. I am intelligent. Second, I am bright. Also, I am clever. Finally, I am brilliant.” Instead, use “There are four reasons I am smart. First, I am intelligent. Second, I am bright. Third, I am clever. Fourth, I am brilliant.” Especially when the points are longer, this makes the argument much easier to follow. (Some people worry that such consistency and repetition is pedantic or stilted, or it makes the writing hard to follow. There is no need for such concerns: none of these is the case.)

Choose good names not only for the concepts that you present in your paper, but for the document source file. Don't name the file after the conference to which you are submitting (the paper might be rejected) or the year. Even if the paper is accepted, such a name won't tell

you what the paper is about when when you look over your source files in later years. Instead, give the paper a name that reflects its content.

Rejection

If you submit technical papers, you will experience rejection. In some cases, rejection indicates that you should move on and begin a different line of research. In most cases, the reviews offer an opportunity to improve the work — and you would rather have a good paper appear at a later date than a poor paper appear earlier.

There is noise in the refereeing system, and even small flaws or omissions in an otherwise good paper may lead to rejection, particularly at the elite venues with small acceptance rates that you should aim. Referees are generally people of good will, but different referees at a conference may have different standards, so the luck of the draw in referees is one small factor in acceptance. When readers misunderstand the paper, that is always at least partly the author's fault! Even if you think the referees have missed the point, you will learn how your work can be misinterpreted, and eliminating those ambiguities will improve the paper.

The wrong lesson to learn from rejection is discouragement or a sense of personal failure. Many papers — even papers that later win awards — are rejected at least once, and as you return to your work, your results will improve. Getting feedback on your paper will help you to improve it. (On the other hand, you don't want to get a reputation for submitting half-baked work. If you know the flaws that will make the referees reject your paper, then don't bother to submit it. Only submit to obtain new information, and reviews do often indicate concerns you did not predict ahead of time.)

Miscellaneous

Use a consistent number of digits of precision. If the measured data are 1.23, 45.67, and 891.23, for example, you might report them as 1.23, 45.7, and 891, or as 1.2, 46, and 890, or as 1, 50, and 900. Use an appropriate number of digits of precision that reflects the measurement process — if you don't have confidence in the 3rd digit of precision (and there is rarely reason to have confidence in it!), omit it. Keep in mind the message you wish to convey to readers — too many digits of precision can distract readers from the larger trends and the big picture.

A related work section should not only explain what research others have done, but in each case should compare and contrast that to your work. Additionally, for each significant piece of related work, after reading your related work, readers should understand the key idea and contribution of that work.