

On Writing Technical Notes

Karl Stratos

Leonardo Da Vinci (1452–1519) was famous for his lifelong habit of recording ideas and drawings in his notebooks. The ability to render amorphous thoughts into something readable is the cornerstone of human intelligence. It forces us to confront flaws in our understanding, and allows us to compose ideas to generate more complex ones. There are certain principles I live by when I write technical notes.

Content is more important than writing. Don't waste your time trying to write about something without first having clear and significant content. If there is not enough content, your writing is insubstantial no matter how well written. Stop writing and develop content by brainstorming, deriving equations, and running experiments.¹

The RAM principle. We have limited amount of short term memory (“RAM”) that gets used up by each word we read. Thus, you will want to deliver the most important information first at all cost. Don't waste the reader's precious RAM with an empty prologue. You must deliver your main results economically in the first two or three pages so that the reader can store them in memory instead of other junk. Banish less important results to appendices. Forbid yourself from adding even a single superfluous word.

Maximize clarity. It's unfortunately possible for a paper to be substantial and unclear. But what a waste! Compared to content creation, writing clearly is easy. You just have to follow a few rules.

- Be consistent. Use the same symbol to always mean the same thing. If N is the number of training examples, don't use N to denote the length of a sequence.
- Use standard mathematical notation. When in doubt, explain your notation. After you write $\mathbb{1}(x < c)$, don't make the reader guess what it means even though it's pretty clear. Instead, say: “We write $\mathbb{1}(A)$ to denote the indicator function that outputs 1 if A is true and 0 otherwise”.
- But don't over-explain. Your writing should be concise while complete. It takes practice to strike a balance.
- Never define a variable that's unused.
- Never use a variable that's undefined.
- Avoid long gaps between variable definitions and usage. A variable must be either (i) so central that it is always kept in memory, or (ii) defined in close proximity. If I read “ $z_1 \dots z_n$ ” and I can't immediately figure out what n is, I conclude that the writing is poor.
- A pronoun such as “it” or “they” must be overwhelmingly clear in what it points to. Otherwise, just rewrite the previous expression.
- Use the present tense. Talk to the reader in the present. Say “We evaluate our model on X ” instead of “We evaluated our model on X ”. Only use the past tense to refer to an explicit past event (e.g., “In a preliminary experiment we found that X performed well”).
- Always run automatic spell check before publishing your writing.
- Pay special attention to equations (particularly subscripts and indices), algorithms, and tables. They are much more important than words. It is often helpful to print your note and read it on physical paper.
- For non-native English speakers (myself included): It is not unimportant to use grammatically correct and fluent sentences. Learning a language is a giant task and there is no easy solution. Read extensively, not just technical papers but general books. Actively hunt down grammatical mistakes with feedback.

Limit your vocabulary. Never use the word “utilize”. It means the same thing as “use”, but you use “utilize” because you mistakenly think it sounds more significant. Never append “etc.” or “so on” to a list. It only reduces clarity. Be sparing with words like “powerful” and “impressive” when describing a method, since there will always be more powerful and impressive methods later and you will not have words left for them. It is best to report results as they are in plain language and let their significance be self-evident.

¹That said, writing can help you develop content. Identify smaller intermediate results and fossilize them as lemmas. Then revisit the original question and see if you can make progress when you use those lemmas. Writing down intermediate results can help you avoid running in a loop.