CSC290 Communication Skills for Computer Scientists

Lisa Zhang

Lecture 8; Nov 5, 2018

Today

- Warm-up:
  - Breaking ideas into parts
  - Proofreading
- Technical Writing
- Midterm
- Titles and Subjects

Warm-up

Break down this idea

> *The researchers analyzed 500 job postings for IT positions and studied the different types of soft skills that are sought after and other soft skills that are ignored even though they are are crucial for the job, as discussed in research by Person et al. [1].*

How do we break down this idea?

Break down this idea

> *The researchers analyzed 500 job postings for IT positions and studied the different types of soft skills that are sought after and other soft skills that are ignored even though they are are crucial for the job, as discussed in research by Person et al. [1].*

How do we break down this idea?

- Researchers analyzed 500 job postings for IT positions
- Researchers studied the types of soft skills that are sought after vs. ignored
- There is evidence that the ignored soft skills may be crucial for job performance [1]

Break down this idea

> *There are many significant details to this article, however, despite much of the research that authors conducted, much of their evidence to support their thesis either lack detail or is biased after further research was conducted.*

How do we break down this idea?

Break down this idea

> *There are many significant details to this article, however, despite much of the research that authors conducted, much of their evidence to support their thesis either lack detail or is biased after further research was conducted.*

How do we break down this idea?

- The research is detailed.
- The authors conducted a lot of research.
- But the evidence is not enough to justify the thesis.

How to proofread

- Go line by line, sentence by sentence.
- Read your text out loud. (e.g. https://ttsreader.com/)
- Be familiar with all the issues we discussed in class, and more:
  - vague pronouns (they, them, it, this – are those really clear?)
  - vague verbs (get, do, make – can you choose a more specific verb?)
  - vague sentences / ideas (does your sentence really say anything?)
  - run-on sentences, commas, etc. . .

Proofreading

Their research, however, could have been improved upon by taking a closer look at individual companies and their specific needs as opposed to lumping all of them as if they are all the same. It makes sense for a large company to have a balanced set of soft skills since there's a lot of other people that people have to talk to, for a small company it might not make so much sense, because smaller companies might care more about people's tecnical skills so that code gets written.

Proofreading

Their research, however, could have been improved upon by taking a closer look at individual companies and their specific needs as opposed to lumping all of them as if they are all the same. It makes sense for a large company to have a balanced set of soft skills since there's a lot of other people that people have to talk to, for a small company it might not make so much sense, because smaller companies might care more about people's tecnical skills so that code gets written.

Use the same amount of focus when proofreading your own work!

Mathematical and Technical Writing

What is technical writing?

Any writing that communicates technical material.

What is technical writing?

Any writing that communicates technical material.

Examples:

What is technical writing?

Any writing that communicates technical material.

Examples:

- User manual
- Documentation
- Help files
- Bug report
- Instructions

How is technical writing different from normal writing?

How is technical writing different from normal writing?

- Dense, difficult material that needs many sign posts and transition words to keep readers on track.
- Clarity and precision is most important.
- Often involves symbols (mathematical symbols, function names, etc) that need to be differentiate from words.
  - Symbols increase clarity, but makes the writing more dense.
  - $\forall x \in y, x \leq 3$

Guidelines for technical writing

- We'll talk about a few guidelines for technical writing.
- The most important is to **keep the reader in mind**.
  - What is the background of the reader?
  - What do the readers know so far?
  - How will they be reading your material? Will they read it cover-to-cover, or will they be skipping sections?
  - **What do they want to do after they read your writing?**

Remember the SMCR model of communication!

Symbols in different formulas

Symbols in different formulas should be separated by words.

- **Bad**: Consider S_q, $q < p$.
- **Good**: Consider S_q, where $q < p$.

Symbols in different formulas

Symbols in different formulas should be separated by words.

- **Bad**: Consider S_q, $q < p$.
- **Good**: Consider S_q, where $q < p$.

Example:

- **Bad**: After calling the function `next`, `finish` should be called.

Symbols in different formulas

Symbols in different formulas should be separated by words.

- **Bad**: Consider S_q, $q < p$.
- **Good**: Consider S_q, where $q < p$.

Example:

- **Bad**: After calling the function `next`, `finish` should be called.
- **Good**: After calling the function `next`, call the function `finish`.

Symbols at beginning of sentences

Avoid beginning a sentence with a symbol.

- **Bad**: `print` is a Python builtin function.
- **Good**: The function `print` is a Python builtin function.
- **Good**: In Python, `print` is a builtin function.

Example:

- **Bad**: $x^n - a$ has n distinct zeros.

Symbols at beginning of sentences

Avoid beginning a sentence with a symbol.

- **Bad**: `print` is a Python builtin function.
- **Good**: The function `print` is a Python builtin function.
- **Good**: In Python, `print` is a builtin function.

Example:

- **Bad**: $x^n - a$ has n distinct zeros.
- **Good**: The polynomial $x^n - a$ has n distinct zeros.

Instead of starting a sentence with `<symbol>`, annotate its **type**.

Annotating **types** of pronouns and symbols will make your text easier to follow.

Avoid passive voice

In Computer Science, use “we” to avoid passive voice.

- **Bad**: The following result can be proved by contradiction.
- **Good**: We can prove the following result by contradiction.

Example:

- **Bad**: The design principles should be followed whenever code is written.

Avoid passive voice

In Computer Science, use "we" to avoid passive voice.

- **Bad**: The following result can be proved by contradiction.
- **Good**: We can prove the following result by contradiction.

Example:

- **Bad**: The design principles should be followed whenever code is written.
- **Good**: We should follow the design principles whenever we write code.

How to begin

Your opening paragraph should be your best paragraph. Its first sentence should be your best sentence.

Avoid starting sentences of the form "An x is y."

- **Bad**: An important method for internal sorting is quicksort.
- **Good**: Quicksort is an important method for internal sorting, because . . .

Example:

- **Bad**: A commonly used data structure is the priority queue.

How to begin

Your opening paragraph should be your best paragraph. Its first sentence should be your best sentence.

Avoid starting sentences of the form "An x is y."

- **Bad**: An important method for internal sorting is quicksort.
- **Good**: Quicksort is an important method for internal sorting, because . . .

Example:

- **Bad**: A commonly used data structure is the priority queue.
- **Good**: Priority queues are significant components of the data structures needed for many applications.

Left-to-right

Readers read left-to-right, your sentences should be understandable.

- **Bad**: We prove that `<grunt>` and `<snort>` implies `<blah>`.

Why?

Left-to-right

Readers read left-to-right, your sentences should be understandable.

- **Bad**: We prove that `<grunt>` and `<snort>` implies `<blah>`.

Why?

- We prove that `<grunt>`
  - *oh! okay!*

Left-to-right

Readers read left-to-right, your sentences should be understandable.

- **Bad**: We prove that `<grunt>` and `<snort>` implies `<blah>`.

Why?

- We prove that `<grunt>`
  - *oh! okay!*
- . . . and `<snort>`
  - *oh, so we're proving two things!*

Left-to-right

Readers read left-to-right, your sentences should be understandable.

- **Bad**: We prove that `<grunt>` and `<snort>` implies `<blah>`.

Why?

- We prove that `<grunt>`
  - *oh! okay!*
- . . . and `<snort>`
  - *oh, so we're proving two things!*
- . . . implies `<blah>`.
  - *wait, let me go back and read again. . .*

Left-to-right

Readers read left-to-right, your sentences should be understandable.

- **Bad**: We prove that `<grunt>` and `<snort>` implies `<blah>`.

Why?

- We prove that `<grunt>`
  - *oh! okay!*
- . . . and `<snort>`
  - *oh, so we're proving two things!*
- . . . implies `<blah>`.
  - *wait, let me go back and read again. . .*
- **Good**: We prove that the two conditions `<grunt>` and `<snort>` imply the result `<blah>`.

Left-to-right . . .

- **Bad**: We want to prove that `<grunt>` and `<snort>` are not true.

Left-to-right . . .

- **Bad**: We want to prove that `<grunt>` and `<snort>` are not true.
- **Good**: We want to disprove the claims `<grunt>` and `<snort>`

Again, annotating the **types** of things can help

- the *claim* `<grunt>`
- the *function* `print`
- the *polynomial* $x^n - a$

More on annotating types

- **Bad**: Some say that waiting until the last minute helps them work quickly, but *this* has pitfalls.

More on annotating types

- **Bad**: Some say that waiting until the last minute helps them work quickly, but *this* has pitfalls.
- **Good**: Some say that waiting until the last minute helps them work quickly, but *this approach* has pitfalls.

Avoid Jargon

Whether a terminology is appropriate or jargon depends on the *audience*. We need technical terminology to describe specific ideas. Ask yourself whether that specificity is worth trading off readability.

Example:

- **Bad**: The patient is being given positive-pressure ventilatory support.

Avoid Jargon

Whether a terminology is appropriate or jargon depends on the *audience*. We need technical terminology to describe specific ideas. Ask yourself whether that specificity is worth trading off readability.

Example:

- **Bad**: The patient is being given positive-pressure ventilatory support.
- **Good**: The patient is on a respirator.

Example:

- **Bad**: This program requires too many CPU cycles to complete.

Avoid Jargon

Whether a terminology is appropriate or jargon depends on the *audience*. We need technical terminology to describe specific ideas. Ask yourself whether that specificity is worth trading off readability.

Example:

- **Bad**: The patient is being given positive-pressure ventilatory support.
- **Good**: The patient is on a respirator.

Example:

- **Bad**: This program requires too many CPU cycles to complete.
- **Good**: This program is too slow.

Define the unfamiliar

If you are defining a new term, *italicize* its first occurence, and define them right away.

Choose singular over plural noun

Singular nouns are more clear:

– **Bad**: Lexical analyzers translate regular expressions into nondeterministic finite automata.

Choose singular over plural noun

Singular nouns are more clear:

– **Bad**: Lexical analyzers translate regular expressions into nondeterministic finite automata.

– **Good**: A lexical analyzer translates each regular expression into a nondeterministic finite automaton.

How will you know if a single lexical analyzer translates one expression or many?

Structure

Each paragraph should have a clear purpose.

Give plenty of examples for new definitions.

Use lists, tables, and other structures when appropriate.

Highlight Important Information

We made a bunch of changes: The registration chapter has been split in two, between adding registration and then associating users with objects. The chapter was giant before so this makes it more managable. Screenshots of the admin have been updated to reflect the new style. The few minor typos have been fixed. Updated the version of django-registration-redux that we use to 1.3. Last but not least, the Introduction has been updated.

Highlight Important Information

We made a bunch of changes:

- **The registration chapter has been split in two**, between adding registration and then associating users with objects. The chapter was giant before so this makes it more managable.
- **Screenshots of the admin have been updated** to reflect the new style.
- **The few minor typos** have been fixed.
- **Updated the version** of django-registration-redux that we use to 1.3.
- **The Introduction has been updated.**

Exercise

Explain what the modulo operator % does in Python to a student who is learning how to program. You can assume that the student completed high school math courses.

Recall:

```
>>> 5 % 2
1
>>> 4 % 2
0
```

Midterm

Midterm Averages

Generally well done:

- Individual Average: 72%
- Group Average: 83%

Remark Request

Midterm request are due **end of next class** (Nov 12th). You must submit:

- Your original midterm.
- Cover sheet explaining the specific issue with the grading.

Your entire midterm will be re-graded, so your grade can potentially decrease.

Titles and Subjects

Titles and subjects

- Summarize the content of the message, so that readers know what to expect.
- Write the titles, subjects, and summaries *last*.

Example:

Write a **subject** for the following email:

```
Hello Lisa,

I submitted the CSC290 critical review article three
hours late, but MarkUs seemed to have deducted 2 late
tokens instead of 1. Can you please check this?

Thank you,
Student
```

Example:

Write a **subject** for the following email:

```
Hello Lisa,

I submitted the CSC290 critical review article three
hours late, but MarkUs seemed to have deducted 2 late
tokens instead of 1. Can you please check this?

Thank you,
Student
```

(Lisa's answer: **CSC290 late token discrepancy**)

Example:

Write a **title** for the following question on a course message board:

```
I don't quite understand the rule for when round()
rounds up or down with a number that has 5 as its
last digit.

>>>round(3.5)
4
>>>round(4.5)
4

I thought it depends if the number to the left of
the 5 is even or odd, but then using round with
more decimal places seems to always round down and
never up.
```

Example:

Write a **title** for the following question on a course message board:

```
I don't quite understand the rule for when round()
rounds up or down with a number that has 5 as its
last digit.

>>>round(3.5)
4
>>>round(4.5)
4

I thought it depends if the number to the left of
the 5 is even or odd, but then using round with
more decimal places seems to always round down and
never up.
```

(Lisa's answer: **Behaviour of `round` for `x.5`**)

Example:

Write a **title** for the following question on a course message board:

```
Hi. I was just wondering when Assignment 2 will be
posted? Just wondering because the professor told us
last class that it would be posted yesterday.
```

Example:

Write a **title** for the following question on a course message board:

`Hi. I was just wondering when Assignment 2 will be posted? Just wondering because the professor told us last class that it would be posted yesterday.`

(Lisa's answer: **Assignment 2 posting date**)

Example:

Write a **subject** for the following email:

```
Hi Professor,

I handed in my assignment at 7:55pm, then made some
changes and submitted around 8:51pm. The screen was
green and the resubmission time stamp was posted for
8:51pm. Then around 9pm, I refreshed the page and
the time stamp said 9:00:23, which is after the
deadline.  Will I be penalized?
```

Example:

Write a **subject** for the following email:

```
Hi Professor,

I handed in my assignment at 7:55pm, then made some
changes and submitted around 8:51pm. The screen was
green and the resubmission time stamp was posted for
8:51pm. Then around 9pm, I refreshed the page and
the time stamp said 9:00:23, which is after the
deadline.  Will I be penalized?
```

(Lisa's answer: **Markus resubmission timestamp**)

References

[1] Knuth on Mathematical Writing
http://jmlr.csail.mit.edu/reviewing-papers/knuth_mathematical_writing.pdf

[2] http://web.mit.edu/me-ugoffice/communication/technical-writing.pdf

[3] https://www.cs.tufts.edu/~nr/pubs/learn.pdf