UTSA CS 4593: CS-CURE

Course-based Undergraduate Research Experience in CS

Amanda Fernandez, Ph.D
UTSA Department of Computer Science

Spring 2024

Week 11: **Technical Writing**

UTSA CS-CURE

Week 11

- Objectives:
    - Understand the key principles of effective technical writing for CS.
    - Learn strategies for clear and concise communication of technical concepts.

- Deliverables:
    - Activity 7 worksheet (*in-class Thursday*)
    - SIG Meeting 3 worksheet (*in-class Thursday*)

Getting Started: **Technical Writing**

Technical Writing

In CS

Before you write, know **why** you are writing.

- Who is it for? *Know your audience*
- Where will it be published? *If applicable*
- What is the format? *Template, length, criteria*
- How long do you have? *Submission deadline*

Understanding Your Audience

Technical Writing

- What **do** they care about?
- What ***don't*** they care about?

- *Examples:*
  - Theory-driven communities mostly want proofs for strong statements, not limited experiments or details on the exact hardware specs.
  - AI communities mostly want to see experiments & ablations on the given benchmarks (and beyond), including evaluation metrics (e.g. accuracy), system info (hardware), & runtime, rather than proofs.

Understanding Your Audience

Technical Writing

- **Tone**
    - Is it very formal?
    - Are fun/witty titles encouraged?
    - Be careful writing about comparative works - *give them **credit**, don't say "everything else is terrible and ours is amazing".*

Understanding Your Audience

Technical Writing

- **General guidelines**
    - Writing is an iterative & collaborative process. *Write it terribly, revise it with a friend.*
    - Every sentence should serve a purpose. *If you can remove it entirely and the content stays the same, it isn't worth keeping.*
    - Cite your sources while you're writing. *Every statement should be evaluated to see if a citation is needed.*

Technical Writing

Getting Started

1. Get the template for the publication
2. Set up the general sections - *abstract, introduction, related work, (your method), experiments, results, conclusion*
3. Outline each section - *ordered list of what needs to be discussed in the section or what subsections should be present.*
4. Write out the bullets you have - *your doc will be messy, but the structure is there. Fill in what you have (e.g. dataset info) and leave the others until later (e.g. experiments).*
5. Write the story - *once all/most bullets are replaced with sentences, read through to ensure the sentences fit together to tell a good story.*
6. Revise & edit - *a lot! This can be harder than it seems. Don't worry about page length until this step!*

The **Abstract**

Abstracts

In CS

Brief summary of the purpose and contribution of the paper.

- 150-300 words, typically 5-7 sentences.
- Should be understandable to those in your larger field (e.g. anyone in CS)

Must address all of the following (typically found in this order):

- What is the problem & why do we care?
- What existing solutions are there?
- What flaw in these solutions does your approach address?
- What is the technical contribution of the paper?
- How do you assess its efficacy?

The Dragon Story

Writing an Abstract for a Grant

Once upon a time, there was a big mean dragon.

He threatened the kingdom and all of the people there with his fiery breath.

Fortunately, there were some brave knights who lived in the kingdom.

Their shields and swords were the best in the land.

These weapons were fire proof and their swords were even sharper than dragons teeth.

The brave knights have trained in a land far far away and are ready to take on the terrible beast.

With the dragon gone, the kingdom will live in harmony happily ever after.

The Dragon Story

Writing an Abstract for a Grant

- What is the overall problem you are trying to solve? *Big mean dragon*
- Why is the problem important? *Threatening the kingdom*
- How is your team qualified to do this work? *Brave knights*
- What is your core technology? *Shields & swords*
- How will your technology solve the problem? *Fireproof shields, swords sharper*
- Why is your idea feasible *now*? *Trained & ready*
- What will success look like? *Dragon gone, harmony ensues*

Abstracts

In CS

General guidelines:

- Get an understanding of what is expected by **reading** a number of abstracts from papers in the conference you plan to submit to.
- Write out your answers to the **questions** first, then weave them together to create a better flow.
- If you can, **write this last!** *The questions are easier to answer once everything is proven/experiments completed.*

The **Introduction**

Introduction

In CS

Brief summary of the purpose and contribution of the paper.

- Should go into more technical depth than the abstract
- *Should not restate the abstract*

Must address all of the points in the abstract, expanding with background detail about the problem and highlighting the important aspects of your contribution.

Introduction

In CS

General guidelines:

- If you can, **write this last too!** *The overall "story" of your paper will be clearer with more details and analysis of related work.*
- **Avoid overly broad statements** or known facts/definitions (e.g. "deep learning is a type of AI", "cybersecurity is important because..")
- **Tell a good story** - this is where you get to provide general background and get the reader excited about the topic.

The rest of your manuscript

Related Works Section

Technical Writing

- **Introduce** the section with a brief overview of categories of related approaches.
- Include **each category as a subsection**.
    - Define it, and differentiate it from other categories.
    - For each related work, try writing 1 sentence that concisely states how it is different from the others *(you should be able to get this info from their abstract)*
- **Important: be careful not to plagiarize.**

Experiments Section

Justifying your contribution

- **Introduce** the section with a brief overview of what experiments will be run *(justify them if you include something that isn't an established benchmark)*
- **Setup -** details on the hardware, implementations, as applicable
- **Data** - if not a benchmark, provide details about the dataset (*size, format, source, analysis)*
- **Experiment** - each experiment should have its own subsection

Results Section

Showcasing your work

- You may include results in the experiments section, or move them to their own, depending on the flow of the paper.
- **Evaluate** your approach **in comparison with** state-of-the-art.
- **Ablation studies** can help justify your contributions
    - *If you remove your new feature, performance should suffer.*

Conclusions Section

Discussion of your findings

- Summarize the paper again in 2-3 sentences.
- Identify **limitations** of your approach.
- Detail **future work** to either expand the approach or recommend relevant applications.

General Guidelines

Technical Writing

General guidelines

- **NEVER** copy/paste from another paper, even if “it’s just my notes”
    - Take the time to paraphrase it at least, write your own words then cite the source.

Technical Writing

General guidelines

- **References**
    - There isn’t a set number of references you should have.
        - *Review recent papers from the same venue to get a general idea.*
    - Be sure to include papers from that conference within the last 2 years.
        - *Demonstrates familiarity with the community and helps fit your work in.*

Technical Writing

General guidelines

- **Clarity**
    - The goal is clearly & concisely getting your idea across
        - *Not to obfuscate or overcomplicate, or go on for pages & pages..*

Q&A

Wrap-Up

Tuesday

- Understand the key principles of effective technical writing for CS.
- Learn strategies for clear and concise communication of technical concepts.

- To Do:
    - Activity 7 worksheet (*in-class Thursday*)
    - SIG Meeting 3 worksheet (*in-class Thursday*)

See you Thursday!

Activity 7:
Technical Writing

Activity 7: **Technical Writing**

Practice clear and concise communication of technical concepts

- Goal: draft an abstract for your research paper

- Activity:
    - Answer each of the abstract writing questions individually.
    - String together each answer and revise into a “story”

SIG Meeting 3:
Abstracts & Peer Feedback

SIG Meeting 3: **Abstracts & Peer Feedback**

Engaging in the iterative & collaborative research writing process

- Goal: get feedback on your abstract, help edit/revise abstracts in your field

- Activity:
    - Trade abstracts with someone in your SIG
    - Review each of the abstract questions - is there anything missing or unclear?
    - What would make this “story” more interesting?

Wrap-Up

Thursday

- Understand the key principles of effective technical writing for CS.
- Learn strategies for clear and concise communication of technical concepts.

- To Do:
    - Activity 7 worksheet (*in-class Thursday*)
    - SIG Meeting 3 worksheet (*in-class Thursday*)

See you next week!