Writing Technical Papers: Some Practical Advice

Kristen LeFevre

Much content borrowed from Jennifer Widom (http://infolab.stanford.edu/~widom/paper-writing.html)

1

First, Some Caveats

- Every area of CS is a bit different
  - Different conferences and journals have different expectations
- Try to present an algorithm for writing a conference paper in database systems
  - Really, problem is "AI Complete"
  - Still, there are principles you can learn

2

Typical Paper Organization

1. Paper Title
2. Abstract
3. Introduction
4. Related Work**
5. Technical Body
6. Experiments
7. Conclusion, Future Work

3

(Fictitious) Running Example

- New algorithm for external merge-sort
- Reduces complexity from O(n *log* n) to O(n) by allowing some bounded "unsortedness" in the result
- Plan to write up results for a conference

4

Paper Title

- Can be long and descriptive
  - Linear-Time External Multipass Sort with Approximation Guarantees
- Short and sweet
  - Approximate External Sort
- Catchy and Cute
  - Floosh: A Linear-Time Algorithm for Approximate External Sort

5

Abstract

- State the problem
- State your solution and technical contributions
- Short and precise
- Do not repeat the abstract word-for-word in the introduction

6

The Introduction

- **Simple approach:** Include five paragraphs answering the following questions:
  - What is the problem?
  - Why is it interesting and important?
    - Remember your audience
  - Why is it hard?
    - Why do naïve approaches fail?
  - Why hasn't it been solved before?
    - Or, how is my approach better than previous solutions?
  - What are the key components of my approach and results? Include specific limitations.

7

The Introduction

- Final paragraph or subsection: Summary of Contributions
  - Bulleted list of major contributions, mentioning where they can be found

Your high school English teacher said bullets were poor style, but they are a staple in CS conference papers.

8

The Introduction -- Example

- What is the problem?
  In this paper, we revisit the classical database problem of external sort…
- Why is it interesting and important?
  While memory sizes are increasing, the sizes of the largest databases are growing at a faster rate. Thus, external sort remains critical to many core database operations including joins, aggregation, …
- Exercise: *Write the remainder of the introduction for the running example.*

9

Related Work

- Where should I put related work?
  - **Beginning** (right after Introduction)
    - If it can be short, yet detailed enough
    - If you need many details of related work to describe your solution
  - **End** (right before Conclusion)
    - If it can be summarized early on (in Intro)
    - If sufficient comparison requires the technical content of your paper

10

Related Work

- Some thoughts
  - It's very important to be comprehensive
  - Related work shouldn't just be a laundry list of references (common mistake)
  - Try to make insightful comparisons, especially for closely-related work

11

Technical Body

- Rule of Thumb: A clear new technical contribution should be articulated by the end of page 3
- Important Components:
  - Running Example: When possible, illustrate your ideas using a single running example
    - *Choose the simplest example that is sufficient to illustrate all of your points!*

12

Technical Body

- Important Components (cont.):
  - Preliminaries: Set up notation and terminology; explain material that is not original, but is needed for the paper
    - Terminology & notation are like programming -- You need to declare them before you use them!
    - Strive for precision and simplicity
      - Definitions and descriptions should be unambiguous
      - But, don't introduce notation that is unnecessarily complex

13

Technical Body

- Important Components (cont.)
  - Content: Algorithms, system descriptions, novel analyses, etc.
    - Varies significantly based on the paper, but there are some general principles
    - Try to write "Top-Down":
      - Start with the high-level idea, and drill down to details
    - Write for "skipability"
      - Use backward and forward "pointers"
      - When you refer to an idea elsewhere in the paper, indicate where it can be found

14

Experimental Evaluation

- Many conferences and journals expect experiments
  - Not productive to run experiments just to produce graphs (some papers do)
- Ask Yourself: *Why am I running these experiments?*

15

Remember the Scientific Method?

(We are Computer *Scientists*…)

1. Define the question
2. Gather information, resources, observations
3. Form hypothesis
4. Perform experiment, collect data
5. Analyze data
6. Interpret data and draw conclusions (starting point for new hypothesis)
7. Retest (often done by others)

16

Experimental Evaluation

- **Rule of Thumb:** For each table / graph, you should be able to describe
  - The question
  - The hypothesis
  - The experiment you did, and how it tests the hypothesis
  - Your conclusion
- *The above is an excellent way to write the text of your experimental section!*

17

Running Example -- Hypothetical Experiments

- Recall: Floosh runs in O(n) time; standard external merge-sort runs in O(n log n)
- Hypothesis (High-Level): On standard equipment and "real" data, Floosh runs faster than SEMS
- Experiment(s): Using data from several real data warehouses, observe the difference in (wall clock) running time

18

Running Example -- Hypothetical Experiments

- Hypothesis (More Detailed): The performance of Floosh is influenced by the distribution of input data. On heavily-skewed data, it is slower than SEMS.
- Experiment:
  - Consider using synthetic data for this
  - Vary the sort attribute's *skewness*
  - Compare performance between Floosh and SEMS for different amounts of skew

19

Running Example -- Hypothetical Experiments

- Hypothesis: In practice, the amount of "unsortedness" is often much less than the theoretical upper bound
  - (Assumes you have some way of quantifying unsortedness)
- Exercise: What experiment would you use to test this hypothesis?

20

Experiments: More Thoughts

- Easy to run experiments showing your idea in the best possible light
  - Many papers do; often obvious
- Synthetic data / workload vs. Real data / workload?
  - Real data and workload illustrate *some* realistic setting
  - Synthetic often gives you more control to vary the parameters of the experiment
- Think carefully about what is important to test
  - E.g., Performance experiments won't tell you anything about UI usability.

21

Experiments: Common Sense

- Report all meaningful parameters
  - If measuring performance, report details of hardware / software setup
  - Report important characteristics of data and experimental workload
- Don't forget to include *units* !
  - (My pet peeve)

22

Miscellaneous

- Always run the spell checker
- Every table, figure, and graph should appear on the page where it is first referenced, or the next page
- Make sure references are complete and consistent
  - Don't just copy random Bibtex entries

23

Lessons / Summary

- Assume your reader is lazy
  - Be as direct as possible in stating key points, contributions
  - In addition to precise technical descriptions, illustrate key points with examples

- Experiments are an often-overlooked place to do interesting work, communicate interesting results
- Your class project is an excellent chance to put these lessons to work!

24