SCIENCE
PASSION
TECHNOLOGY

Introduction to Scientific Writing
02 Scientific Reading and Writing

Matthias Boehm

Graz University of Technology, Austria
Computer Science and Biomedical Engineering
Institute of Interactive Systems and Data Science
BMK endowed chair for Data Management

Last update: Nov 04, 2021

ISDS

Announcements/Org

- **#1 Virtual Lectures**
  - https://tugraz.webex.com/meet/m.boehm
  - Optional attendance (independent of COVID)

- **#2 Course Registrations** (as of Oct 29)
  - Changes in WS20/21, now max constraints
  - **Introduction to Scientific Writing**

ISDS Group Boehm

39/40

Agenda

- **Scientific Reading**
- **Scientific Writing**

Scientific Writing skills can only be learned hands on, and incrementally improved w/ experience

Scientific Reading

In Computer Science (Data Management)

Types of Reading

- **Skimming**
  - **Goal:** understand what the paper/thesis is about
  - Read abstract, and optionally introduction
    scan paper (sections/subsections, structure, figures)

What?

- **Understanding**
  - **Goal:** understand how the presented approach accomplishes the paper's goals
  - **#1** Skimming (see above)
  - **#2** Read the whole paper sequentially, add notes/annotations

How?

- **Reviewing**
  - **Goal:** evaluate potential impact, and limitations
  - **#1** Skimming (see above)
  - **#2** Understanding (see above) + strengths and weaknesses
  - **#3** Write summary, strong/weak points, detailed comments (incl reading)

What's Wrong?

Finding Relevant Work

- **Motivation**
  - Some research areas might be very large (e.g., index structures, compression)
  - How do you find relevant scientific papers/thesis via **multiple channels**
- **By Venue/Year**
  - Start of top-tier conferences/journals and find latest work
  - These papers' related work should provide a good categorization and discussion of related work → **recursive lookup**
- **By Author**
  - Sometimes there are well-known experts in a certain sub-area
  - Find author publications via DBLP and other libraries
- **By Keywords**
  - Broad survey of other related work, to augment the bias year/venue/author approach

Finding Relevant Work, cont.

- **Prefer Trustworthy Sources**
  - Archival publications, awareness of peer-review
  - From right communities (e.g., ML systems vs ML algorithms)
  - Reputation of website, authors, etc

- **Recap: Give Credit**
  - Cite broadly, **give credit to inspiring ideas**, create connections
  - Honestly acknowledge **limitations of your approach**

Process of Reading – Skimming/Understanding

- **Abstract and Structure**

- **#1 Partial Reading** (mostly skimming)
  - Read into each paragraph until you get what it's about
  - 1st sentence/label: **topic sentence**

- **#2 Fast Reading**
  - Normal reading vs **reading w/o vocalization**
  - Avoid need for rereading text
    - Back/forward references,
    - Misplacement after distractions
    - Rereading due to lack of understanding

➔ **Read according to your goals of reading**

Process of Reading – Understanding/Evaluation

- **Skepticism**
  - Critical reading is important for **understanding** and **evaluation**
  - **#1** Start open-minded, listen to arguments and trust provided evidence
  - **#2 Don't accept** superficial, contradictory, or unproven claims
  - **#3** If there are problems, which **constructive feedback** could you give or how could the problems be addressed?

- **Questions to Ask Yourself?**
  - What is the problem? Is it a real or artificial problem?
  - How would you solve the problem yourself?
  - How is the paper solving the problem?
  - Is this the simplest approach that yields these results (justified complexity)
  - Are there limitations that are not covered by the paper?
  - Is there existing work that already addresses the same problem?

Proofreading Your Own Paper

- **#1 Read Slowly & Carefully**
  - **Problem:** Brain interpolates between words
  - Awareness of **common syntactic issues**
    (the the, missing/wrong articles, adapt/adopt)
  - Awareness of **common semantic issues**
    (missing reference, inconsistent / no logical consequence)

→ **Read out loud**
→ **Use PDF-to-Speech**

- **#2 Read Fully**
  - **Read and annotate issue**, don't fix immediately (destroys the flow)
  - Take annotated document and fix issues

- **#3 Ask Big Questions**
  - **Pitfall:** Being **overly focused on syntactic/local issues**
  - Is the overall idea clearly communicated and does it make sense?
  - Are there missing pieces, missing experiments, missing related work?

Reviewing (how NOT to review a paper)

- **Paper Reviews**
  - **Goals:** paper selection, ensure high quality, constructive feedback and recommendations, widen own horizon
  - Lots of similarities to code reviews in OSS

- **Learning by What NOT to Do**
  - Accept if no time to review
  - The Goldilocks Method (examples, proofs, theoretical analysis, experiments)
  - If you can't say something nasty … (ignore good parts, focus on weaknesses)
  - Silent but deadly (low scores, no comments)
  - The Natives are Restless (recommend full rewrite by native English speaker)
  - The Referee Moves the Goalpost (changed problem)
  - Blind reviewing

[Graham Cormode: How NOT to review a paper: the tools and techniques of the adversarial reviewer. **SIGMOD Rec. 37(4) 2008**]

```
This paper leaves many questions unanswered.
      Some claims are questionable.
The paper is of limited interest.
```

Reviewing, cont. (how NOT to review a paper)

- **Introduction**
  - Disagree w/ “Interestingly...”, “Importantly...” or “In practice”,
- **Related Work**
  - “Many important references are omitted”
- **Proposed Method**
  - To simple, impractical, or well-known; correctness?
- **Experiments**
  - Datasets synthetic/real, not all aspects evaluated, too small datasets
- **Conclusions**
  - Disagree w/ every claim; future work can be dismissed

Adversarial Paper Summary

This paper attempts to address the well-studied problem of Graticule Optimization. It proposes the obvious approach of simply storing a set of reference points and calculating offsets. Such approaches are well known in this area. It goes on to propose some simple variations based on precalculating distances. This is an approach that I would expect any straightforward implementation to adopt. Some proof-of-concept experiments show that on a few data sets, the results are slightly better than the most naïve prior methods.

Excursus: An Automatic CS Paper Generator

- **SCIgen: https://pdos.csail.mit.edu/archive/scigen**
  - Generates random CS research papers, incl graphs and figures
  - Uses hand-written context-free grammar
  - Test for low-submission standards of conferences

- **Two Examples**

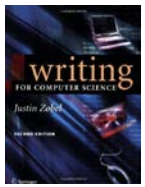
  [Jeremy Stribling, Daniel Aguayo and Maxwell Krohn: Rooter: A Methodology for the Typical Unification of Access Points and Redundancy]
  → **accepted** as "non-reviewed" **WMSCI 2005**

  [Thomer M. Gil: The Influence of Probabilistic Methodologies on Networking]
  → **rejected**

➔ **Meaningless mix of sentences and technical terms**
(today: GPT-3, what's the take-away for your own papers?)

Scientific Writing

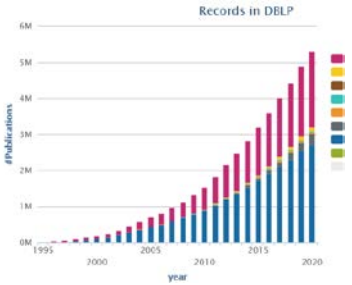
In Computer Science (Data Management)



[Justin Zobel: Writing for Computer Science,
2nd ed. Springer 2004, ISBN 978-1-85233-802-2]

Recap: Writing the Paper

- **#1 Know your Audience**
- **#2 Get your Workflow in Order / Incremental Paper Drafts**
- **#3 Mindset: Quality over Quantity**
  - Aim for top-tier conferences/journals (act as filter)
  - Make the paper useful for others (ideas, evidence, code)



- **#4 Easily Readable: Quality ∝ Time**
  - **Make it easy to skim the paper**
    → paragraph labels, self-explanatory figures (close to text), and structure
  - Avoid unnecessary formalism → as simple as possible
  - Shortening the text in favor of structure improves readability
  - Ex. Compressed Linear Algebra
    - Initial SIGMOD submission: **12+3 pages**
    - Final PVLDB submission: **12 pages**
      (+ more figures, experiments, etc)

Scope and Structure

- **Sections and Subsections**
  - Abstract (see 01 Structure of Scientific Papers)
  - Introduction → context, problem, contributions
  - Background / Preliminaries → necessary background for understanding
  - Main Part → your technical core contributions
  - Main Part 2
  - Experiments → setting, micro benchmarks end-to-end
  - Related Work → areas of related work, differences
  - Conclusions → summary, conclusions, and future work
  - References

- **Recommendations**
  - Avoid sections with only one subsection (e.g., 2 and 2.1)
  - Avoid more than two or at most three nesting levels
  - Clearly separate motivation/background from your own work

Scope and Structure, cont.

- **Bullet Lists**
  - begin{itemize} … \item \end{itemize}
  - begin{enumerate} … \end{enumerate}
- **Figures and Tables**
  - Captions below figures, above tables
- **Code**
  - \begin{verbatim} … \end{verbatim}
- **Theorem, Definition, Examples**
  - Refine theorem environments as needed
- **Algorithms**
  - Can be clearer than text, but not always
  - Carefully select the right level of abstraction

Data Structure: The MNC sketch $\mathbf{h}_A$ of an $m \times n$ matrix $\mathbf{A}$ comprises the following information, where we use $\mathbf{h}$ as a shorthand whenever the context is clear.

- *Row/Column NNZs:* Count vectors $\mathbf{h}^r = \text{rowSums}(\mathbf{A} \neq 0)$ and $\mathbf{h}^c = \text{colSums}(\mathbf{A} \neq 0)$ indicate the NNZs per row and column, where $\mathbf{h}^r_i$ is the count of the ith row.
- *Extended Row/Column NNZs:* Count vectors $\mathbf{h}^{er} = \text{rowSums}((\mathbf{A} \neq 0) \cdot (\mathbf{h}^c = 1))$ and $\mathbf{h}^{ec} = \text{colSums}((\mathbf{A} \neq 0) \cdot (\mathbf{h}^r = 1))$ indicate the NNZs per row/column that appear in columns/rows with a single non-zero.

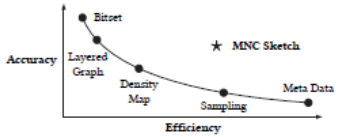


Figure 2: Accuracy/Efficiency Goal of the MNC Sketch.

Table 1: Analysis of Existing Sparsity Estimators.

Estimator	Space	Time	%	Bias
MetaAC E_{ac}	$O(1)$	$O(1)$	✓	
MetaWC E_{wc}	$O(1)$	$O(1)$	✓	$\overline{sc}$
Bitset E_{bmm}	$O(mn + nl + ml)$	$O(mnl)$	✓	
DMap E_{dm}	$O(\frac{mn+nl+ml}{b^2})$	$O(\frac{mnl}{b^2})$	✓	
Sample E_{smpl}	$O(\lvert S\rvert)$	$O(\lvert S\rvert(m+l))$		$\underline{sc}$
LGraph E_{gph}	$O(rd + \text{nnz}(A,B))$	$O(r(d + \text{nnz}(A,B)))$	✓	
MNC E_{mnc}	$O(d)$	$O(d + \text{nnz}(A,B))$	✓	

THEOREM 3.1. *Given MNC sketches $\mathbf{h}_A$ and $\mathbf{h}_B$ for matrices $\mathbf{A}$ and $\mathbf{B}$, the output sparsity s_C of the matrix product $\mathbf{C} = \mathbf{A}\,\mathbf{B}$ can be exactly computed under the assumptions A1 and A2 via a dot product of $\mathbf{h}^c_A$ and $\mathbf{h}^r_B$:*

$$s_C \equiv \hat{s}_C = \mathbf{h}^c_A \mathbf{h}^r_B / (ml) \text{ if } \max(\mathbf{h}^r_A) \leq 1 \vee \max(\mathbf{h}^c_B) \leq 1. \quad (7)$$

Algorithm 1 MNC Sparsity Estimation

Input: MNC sketches $\mathbf{h}_A$ and $\mathbf{h}_B$ for matrices $\mathbf{A}$ and $\mathbf{B}$
Output: Output sparsity s_C

```
// a) basic and extended sparsity estimation, incl upper bound
if max(h^r_A) ≤ 1 ∨ max(h^c_B) ≤ 1 then                    // see Theorem 3.1
    nnz ← h^c_A h^r_B
else if exists(h^{er}_A) ∨ exists(h^{ec}_B) then // extended NNZ counts
    nnz ← h^{ec}_A h^r_B + (h^c_A − h^{ec}_A) h^{er}_B          // exact fraction
    p ← (nnz(h^r_A) − |h^r_A = 1|) · (nnz(h^c_B) − |h^c_B = 1|)   // #cells
    nnz ← nnz + E_dm(h^c_A − h^{ec}_A, h^r_B − h^{er}_B, p) · p  // generic rest
else                                                  // generic fallback estimate
    p ← nnz(h^r_A) · nnz(h^c_B)                                  // #cells
    nnz ← E_dm(h^c_A, h^r_B, p) · p
// b) apply lower bound, see Theorem 3.2
nnz ← max(nnz, |h^r_A > n/2| · |h^c_B > n/2|)              // lower bound
return s_C ← nnz/(ml)
```

Formatting

- **Motivation**
  - A carelessly formatted paper (layout, figures, fonts, underlining) creates a bad first impression
  - Recap: **skimming** and **anchoring**

"The paper's approach is probably equally sloppy"

- **Figures**
  - Use same font and font size as the main text / code in main paper
  - Avoid text overlap, too aggressive **colors**

- **Orphans and Widows**
  - Imprecise definition
  - Avoid few words per line, single line at next page

Strength Reduction: Note that cumsumprod($\mathbf{X}$) uses cumsumprod($\mathbf{B}$)$_{n1}$—i.e., the last block entry—as part of f_{agg}. Similarly, for cumsum($\mathbf{X}$), we could use cumsum($\mathbf{B}$)$_{n:}$. However, this simplifies to colSums($\mathbf{B}$), which avoids materializing the cumsum output block.

Looks ugly and wastes lots of space

- **Highlighting**
  - Use \emph{} (emphasize) over underlining or bold

Punctuation

- **Commas**
  - Whenever a pause is appropriate, or required to avoid ambiguity

```
When using disk[,] tree algorithms
were found to be particularly poor.

A woman without her man is nothing.
A woman: without her, man is nothing.
```

  - **Lists:** red, blue, black, and white (oxford/serial comma)
  - **Special sentence start:** `However, Hence, Therefore, In this paper,`
- **Semicolons**
  - Divide a long sentence into sub-sentences, or separation for emphasis
  - Lists with sublists

```
We use index structures like b-trees, tries,
and hash tables; as well as compression
techniques like run-length encoding,
dictionary encoding, and null suppression.
```

- **Exclamations**
  - `Avoid exclamation marks! Never use more than one!!`

Writing Style

- **Goal: Clear, easy-to-read writing**
- **Variation**
  - Diversity (structure, length of sentences/paragraphs, choice of words, sentence beginning) helps keeping the reader's attention

The system of rational numbers is incomplete. This was discovered 2000 years ago by the Greeks. The problem arises in squares with sides of unit length. The length of the diagonals of these squares is irrational. This discovery was a serious blow to the Greek mathematicians.

The Greeks discovered 2000 years ago that the system of rational numbers is incomplete. The problem is that some quantities, such as the length of the diagonal of a square with unit sides, are irrational. This discovery was a serious blow to the Greek mathematicians.

Writing Style, cont.

- **Prefer Active Voice**
  - Easier to understand, shorter, more interesting to read
  - Use we over I

```
In this section, the background and motivation
for compressed linear algebra is introduced.
```

```
In this section, we provide the background and
motivation for compressed linear algebra.
```

- **Prefer Present Tense**
  - Most content of a research paper can be described in present
  - Exceptions: user studies, (specific experimental setup), related work

- **Use of References**
  - Use \cite{key1,key2}
  - **Don't use refs as nouns**
  - **Prefer primary sources**

```
Later, [40] investigated query
processing on heavyweight Huffman
coding schemes,
```

```
Later, Raman and Swart investigated
query processing on heavyweight
Huffman coding schemes [40],
```

Writing Style, cont.

- **Articles and Spaces**
  - Plural allows to drop articles
  - Use **guarded spaces** for references that should not appear on a new line

```
employ general-purpose
compression techniques
```

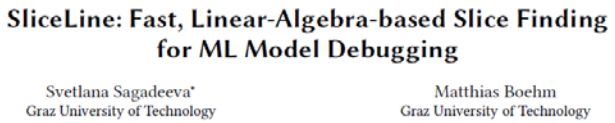
```
employ a general-purpose
compression technique
```

- **Clear References**
  - Make sure there are no unclear "it" or "this" references
  - Add descriptive nouns

```
Each entry q_i can be expressed
over columns as q_i = v^T X_{i:}. We
rewrite this in […]
```

```
Each entry q_i can be expressed
over columns as q_i = v^T X_{i:}. We
rewrite this multiplication in […]
```

- **Titles and Names**
  - Titles: capitalize meaning-carrying words
  - Names: capitalize, e.g., Bayesian, Euclidean
  - References like Figure 1, Table 2, Section 3, Chapter 4, Equation 5 are names as well



SliceLine: Fast, Linear-Algebra-based Slice Finding for ML Model Debugging

Svetlana Sagadeeva* Graz University of Technology

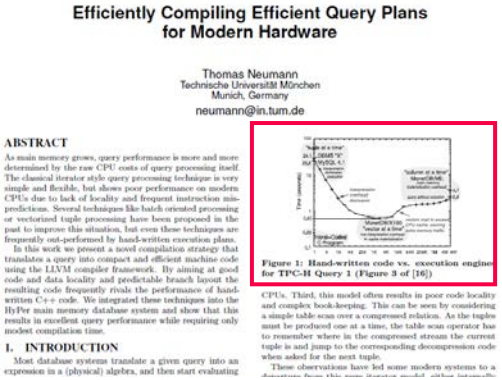
Matthias Boehm Graz University of Technology

```
Figure~\ref{fig:exp1}
Equation~\eqref{eq:e1}
```

Plagiarism

- **#1 Self-Plagiarism (Bad Idea)**
  - Avoid reusing motivation, introduction, figures, and examples
  - Start writing every thesis / paper from scratch (unless thesis summaries/extends previous papers)

- **#2 Figure Plagiarism (Bad Idea)**
  - Never copy figures from other papers, web, etc
  - Create all figures yourself, even for surveys (can be based on ideas of existing papers)
  - **Exceptions** do exist w/ explicit references



Efficiently Compiling Efficient Query Plans for Modern Hardware

Thomas Neumann
Technische Universität München
Munich, Germany
neumann@in.tum.de

ABSTRACT

As main memory grows, query performance is more and more determined by the raw CPU costs of query processing itself. The classical iterator style query processing technique is very simple and flexible, but shows poor performance on modern CPUs due to lack of locality and frequent instruction mispredictions. Several techniques like batch oriented processing or vectorized tuple processing have been proposed in the past to improve this situation, but even these techniques are frequently out-performed by hand-written execution plans.

In this work we present a novel compilation strategy that translates a query into compact and efficient machine code using the LLVM compiler framework. By aiming at good code and data locality and predictable branch layout the resulting code frequently rivals the performance of hand-written C++ code. We integrated these techniques into the HyPer main memory database system and show that this results in excellent query performance while requiring only modest compilation time.

1. INTRODUCTION

Most database systems translate a given query into an expression in a (physical) algebra, and then start evaluating

Figure 1: Hand-written code vs. execution engines for TPC-H Query 1 (Figure 3 of [16])

CPUs. Third, this model often results in poor code locality and complex book-keeping. This can be seen by considering a simple table scan over a compressed relation. As the tuples must be produced one at a time, the table scan operator has to remember where in the compressed stream the current tuple is and jump to the corresponding decompression code when asked for the next tuple.

These observations have led some modern systems to a

- **#3 Plagiarisms (Really Bad Idea)**
  - Never copy figures or text from other peoples work and claim its yours (slight rewording does not change that)
  - For archival scientific publications, there is a high chance it will be detected

Plagiarism – Duplicate Submission

- **Example SIGMOD'21**

A research paper submitted to SIGMOD 2021 cannot be under review for any other publishing forum or presentation venue, including conferences, workshops, and journals, during the time it is being considered for SIGMOD. Furthermore, after you submit a research paper to SIGMOD, you must await the response from SIGMOD and only re-submit elsewhere if your paper is rejected - or withdrawn at your request - from SIGMOD. This restriction applies not only to identical papers but also to papers with a substantial overlap in scientific content and results.

Every research paper submitted to SIGMOD 2021 must present substantial novel research not described in any prior publication. In this context, a prior publication is (a) a paper of five pages or more presented, or accepted for presentation, at a refereed conference or workshop with proceedings; or (b) an article published, or accepted for publication, in a refereed journal. If a SIGMOD 2021 submission has overlap with a prior publication, the submission must cite the prior publication, along with all other relevant published work, following the guidelines in the Anonymity Requirements for Double-Blind Reviewing section below.

Page Limits

[Credit: https://twitter.com/fadeladib/status/1322646406088347649]

- **Most Conferences/Journals**
  - Given predefined template, changes not permitted
  - SIGMOD/PVDLB: **12 pages + unlimited references**
  - ICDE: 12 pages incl. references
- **#1 Avoid Cheating**
  - Don't change the template, fonts, or margins (at least not too excessively)
  - Condensing more text into the paper will make it harder to read
- **#2 Carefully Trim Down Draft**
  - Write unlimited paper, then select, and revise
  - Write and revise section by section as you write

[Eamonn Keogh: How to do good research, get it published in SIGKDD and get it cited!, **KDD 2009**]

- **#3 Never Excuse Missing Content by "lack of space"**

```
Due to the lack of space, we omit
[essential details] /
[essential experiments]
```

Summary and Q&A

- **Scientific Reading**
- **Scientific Writing**

- **Remaining Questions?**
- **Paper Project Selection by Nov 11**

- **Next Lectures**
  - 03 Experiments, Reproducibility, and Projects [Nov 11]