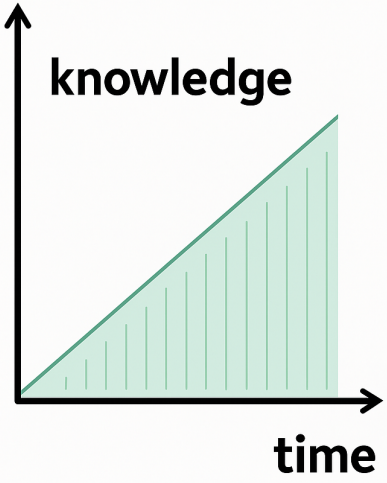
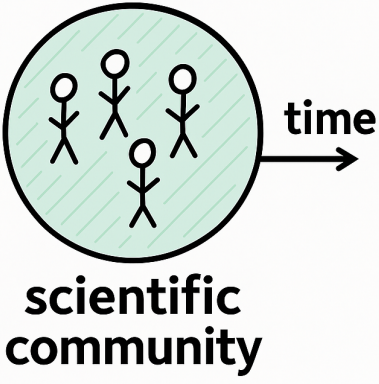
How to Write a Paper

Why do we Write Papers?





Why do We Read Papers?

Reading papers is a critical skill:

- Understand latest developments before they are covered by other media like books

Why do We Read Papers?

Reading papers is a critical skill:

- Understand latest developments before they are covered by other media like books
- Learn how to write and communicate ideas

Why do We Read Papers?

Reading papers is a critical skill:

- Understand latest developments before they are covered by other media like books
- Learn how to write and communicate ideas
- Learn to read critically:
  Ask the right questions, challenge assumptions

Why do We Read Papers?

Reading papers is a critical skill:

- Understand latest developments before they are covered by other media like books
- Learn how to write and communicate ideas
- Learn to read critically:
  Ask the right questions, challenge assumptions
- Synthesize new ideas:
  inspiration comes more often than not when reading the work of others

What to Extract From a Paper?

- **Research question**: Central message

What to Extract From a Paper?

- **Research question**: Central message
- **Impact**: Motivation, relevance, and impact

What to Extract From a Paper?

- **Research question**: Central message
- **Impact**: Motivation, relevance, and impact
- **Contributions**: What is new compared to previous work? How applicable is approach in general?

What to Extract From a Paper?

- **Research question**: Central message
- **Impact**: Motivation, relevance, and impact
- **Contributions**: What is new compared to previous work? How applicable is approach in general?
- **New info for yourself**: What can you learn from this paper? (e.g. good summary of related work)

What to Extract From a Paper?

- **Research question**: Central message
- **Impact**: Motivation, relevance, and impact
- **Contributions**: What is new compared to previous work?
  How applicable is approach in general?
- **New info for yourself**: What can you learn from this paper?
  (e.g. good summary of related work)
- **Conclusions**: Takeaways: Can we build upon this work? If so, how? Ideas for future work?

How to Read

You should read a paper in three passes:

1. Get general idea (5-10 minutes)

How to Read

You should read a paper in three passes:

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion

How to Read

You should read a paper in three passes:

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know what the paper is about now

How to Read

You should read a paper in three passes:

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know what the paper is about now
2. Understand Content ($>=$ 1 hour)

How to Read

You should read a paper in three passes:

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know what the paper is about now
2. Understand Content ($>=$ 1 hour)
   - Read full paper, ignore details (e.g. proofs)

How to Read

You should read a paper in three passes:

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know what the paper is about now
2. Understand Content ($>=$ 1 hour)
   - Read full paper, ignore details (e.g. proofs)
   - Find key points, take notes

How to Read

You should read a paper in three passes:

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know what the paper is about now
2. Understand Content ($>=$ 1 hour)
   - Read full paper, ignore details (e.g. proofs)
   - Find key points, take notes
   - Understand figures

How to Read

You should read a paper in three passes:

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know what the paper is about now
2. Understand Content ($>=$ 1 hour)
   - Read full paper, ignore details (e.g. proofs)
   - Find key points, take notes
   - Understand figures
   - Mark references for further reading

How to Read

You should read a paper in three passes:

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know what the paper is about now
2. Understand Content (>= 1 hour)
   - Read full paper, ignore details (e.g. proofs)
   - Find key points, take notes
   - Understand figures
   - Mark references for further reading
   - You should understand the key points now

How to Read

You should read a paper in three passes:

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know what the paper is about now
2. Understand Content (>= 1 hour)
   - Read full paper, ignore details (e.g. proofs)
   - Find key points, take notes
   - Understand figures
   - Mark references for further reading
   - You should understand the key points now
3. Understand in depth (>= 4 hours)

How to Read

You should read a paper in three passes:

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know what the paper is about now
2. Understand Content (>= 1 hour)
   - Read full paper, ignore details (e.g. proofs)
   - Find key points, take notes
   - Understand figures
   - Mark references for further reading
   - You should understand the key points now
3. Understand in depth (>= 4 hours)
   - Fully understand everything, pay attention to all details

How to Read

You should read a paper in three passes:

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know what the paper is about now
2. Understand Content (>= 1 hour)
   - Read full paper, ignore details (e.g. proofs)
   - Find key points, take notes
   - Understand figures
   - Mark references for further reading
   - You should understand the key points now
3. Understand in depth (>= 4 hours)
   - Fully understand everything, pay attention to all details
   - Check related work

How to Read

You should read a paper in three passes:

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know what the paper is about now
2. Understand Content (>= 1 hour)
   - Read full paper, ignore details (e.g. proofs)
   - Find key points, take notes
   - Understand figures
   - Mark references for further reading
   - You should understand the key points now
3. Understand in depth (>= 4 hours)
   - Fully understand everything, pay attention to all details
   - Check related work
   - Imagine your own implementation of the solution

How to Read

You should read a paper in three passes:

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know what the paper is about now
2. Understand Content (>= 1 hour)
   - Read full paper, ignore details (e.g. proofs)
   - Find key points, take notes
   - Understand figures
   - Mark references for further reading
   - You should understand the key points now
3. Understand in depth (>= 4 hours)
   - Fully understand everything, pay attention to all details
   - Check related work
   - Imagine your own implementation of the solution
   - Question everything

How to Read

You should read a paper in three passes:

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know what the paper is about now
2. Understand Content (>= 1 hour)
   - Read full paper, ignore details (e.g. proofs)
   - Find key points, take notes
   - Understand figures
   - Mark references for further reading
   - You should understand the key points now
3. Understand in depth (>= 4 hours)
   - Fully understand everything, pay attention to all details
   - Check related work
   - Imagine your own implementation of the solution
   - Question everything
   - Generate new ideas for your own work

How to Read for a Review in This Course

1. Get general idea (5-10 minutes)

How to Read for a Review in This Course

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion

How to Read for a Review in This Course

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know the main goal of the paper now

How to Read for a Review in This Course

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know the main goal of the paper now
2. Understand Content (>= 30 min)

How to Read for a Review in This Course

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know the main goal of the paper now
2. Understand Content (>= 30 min)
   - Read full report

How to Read for a Review in This Course

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know the main goal of the paper now
2. Understand Content (>= 30 min)
   - Read full report
   - Find key points, take notes

How to Read for a Review in This Course

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know the main goal of the paper now
2. Understand Content (>= 30 min)
   - Read full report
   - Find key points, take notes
   - Understand figures

How to Read for a Review in This Course

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know the main goal of the paper now
2. Understand Content (>= 30 min)
   - Read full report
   - Find key points, take notes
   - Understand figures
   - You should understand the key points now

How to Read for a Review in This Course

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know the main goal of the paper now
2. Understand Content (>= 30 min)
   - Read full report
   - Find key points, take notes
   - Understand figures
   - You should understand the key points now
3. Review Report (>= 20 min)

How to Read for a Review in This Course

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know the main goal of the paper now
2. Understand Content (>= 30 min)
   - Read full report
   - Find key points, take notes
   - Understand figures
   - You should understand the key points now
3. Review Report (>= 20 min)
   - Check each of the required points in the text

How to Read for a Review in This Course

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know the main goal of the paper now
2. Understand Content (>= 30 min)
   - Read full report
   - Find key points, take notes
   - Understand figures
   - You should understand the key points now
3. Review Report (>= 20 min)
   - Check each of the required points in the text
   - Take notes

How to Read for a Review in This Course

1. Get general idea (5-10 minutes)
   - Read abstract, introduction, headings and subheadings, and conclusion
   - You should know the main goal of the paper now
2. Understand Content (>= 30 min)
   - Read full report
   - Find key points, take notes
   - Understand figures
   - You should understand the key points now
3. Review Report (>= 20 min)
   - Check each of the required points in the text
   - Take notes
   - You should have all information for your review now

What You Might Write

- **Thesis**: Bachelor, Seminar, Master, PhD

What You Might Write

- **Thesis**: Bachelor, Seminar, Master, PhD
- **Research**: Research paper, project proposal

What You Might Write

- **Thesis**: Bachelor, Seminar, Master, PhD
- **Research**: Research paper, project proposal
- **Industry**: Documentation, design document, white paper, website

Writing is Thinking

- Writing forces you to make thoughts concrete

Writing is Thinking

- Writing forces you to make thoughts concrete
- Writing organizes thoughts

Writing is Thinking

- Writing forces you to make thoughts concrete
- Writing organizes thoughts
- Clear writing is explaining

Writing is Thinking

- Writing forces you to make thoughts concrete
- Writing organizes thoughts
- Clear writing is explaining
- Helps you to understand

Writing is Thinking

- Writing forces you to make thoughts concrete
- Writing organizes thoughts
- Clear writing is explaining
- Helps you to understand
- Along the process you will have new ideas

Writing is Thinking

- Writing forces you to make thoughts concrete
- Writing organizes thoughts
- Clear writing is explaining
- Helps you to understand
- Along the process you will have new ideas
- Will even improve your code

Writing is Thinking

- Writing forces you to make thoughts concrete
- Writing organizes thoughts
- Clear writing is explaining
- Helps you to understand
- Along the process you will have new ideas
- Will even improve your code
- Writing is hard

Good Writing is Important

- More and more papers are written and published

Good Writing is Important

- More and more papers are written and published
- Nobody likes badly written papers

Good Writing is Important

- More and more papers are written and published
- Nobody likes badly written papers
- Nobody reads badly written papers

Write Your Paper in Drafts

- Papers aren't written from beginning to end in one session.

Write Your Paper in Drafts

- Papers aren't written from beginning to end in one session.
- Typical progress

Write Your Paper in Drafts

- Papers aren't written from beginning to end in one session.
- Typical progress
  - First draft (structure your thoughs)

Write Your Paper in Drafts

- Papers aren't written from beginning to end in one session.
- Typical progress
  - First draft (structure your thoughs)
  - 1. revised draft (e.g., after feedback)

Write Your Paper in Drafts

- Papers aren't written from beginning to end in one session.
- Typical progress
  - First draft (structure your thoughs)
  - 1. revised draft (e.g., after feedback)
  - 2. revised draft

Write Your Paper in Drafts

- Papers aren't written from beginning to end in one session.
- Typical progress
  - First draft (structure your thoughs)
  - 1. revised draft (e.g., after feedback)
  - 2. revised draft
  - ...

Write Your Paper in Drafts

- Papers aren't written from beginning to end in one session.
- Typical progress
  - First draft (structure your thoughs)
  - 1. revised draft (e.g., after feedback)
  - 2. revised draft
  - ...
  - Final paper

Write Your Paper in Drafts

- Papers aren't written from beginning to end in one session.
- Typical progress
  - First draft (structure your thoughs)
  - 1. revised draft (e.g., after feedback)
  - 2. revised draft
  - ...
  - Final paper
- The earlier the draft, the less time you should invest in polishing

Write Your Paper in Drafts

- Papers aren't written from beginning to end in one session.
- Typical progress
  - First draft (structure your thoughs)
  - 1. revised draft (e.g., after feedback)
  - 2. revised draft
  - ...
  - Final paper
- The earlier the draft, the less time you should invest in polishing
- This helps a lot with avoiding writers block

Write Your Paper in Drafts

- Papers aren't written from beginning to end in one session.
- Typical progress
  - First draft (structure your thoughs)
  - 1. revised draft (e.g., after feedback)
  - 2. revised draft
  - ...
  - Final paper
- The earlier the draft, the less time you should invest in polishing
- This helps a lot with avoiding writers block
- Think about writing early draft in a simple text editor

Write Your Paper in Drafts

- Papers aren't written from beginning to end in one session.
- Typical progress
  - First draft (structure your thoughs)
  - 1. revised draft (e.g., after feedback)
  - 2. revised draft
  - ...
  - Final paper
- The earlier the draft, the less time you should invest in polishing
- This helps a lot with avoiding writers block
- Think about writing early draft in a simple text editor
- Iterations...

Write Your Paper in Drafts

- Papers aren't written from beginning to end in one session.
- Typical progress
  - First draft (structure your thoughs)
  - 1. revised draft (e.g., after feedback)
  - 2. revised draft
  - ...
  - Final paper
- The earlier the draft, the less time you should invest in polishing
- This helps a lot with avoiding writers block
- Think about writing early draft in a simple text editor
- Iterations...
  - Require extreme concentration

Write Your Paper in Drafts

- Papers aren't written from beginning to end in one session.
- Typical progress
  - First draft (structure your thoughs)
  - 1. revised draft (e.g., after feedback)
  - 2. revised draft
  - ...
  - Final paper
- The earlier the draft, the less time you should invest in polishing
- This helps a lot with avoiding writers block
- Think about writing early draft in a simple text editor
- Iterations...
  - Require extreme concentration
  - Maybe do it every morning for an hour

Write Your Paper in Drafts

- Papers aren't written from beginning to end in one session.
- Typical progress
  - First draft (structure your thoughs)
  - 1. revised draft (e.g., after feedback)
  - 2. revised draft
  - ...
  - Final paper
- The earlier the draft, the less time you should invest in polishing
- This helps a lot with avoiding writers block
- Think about writing early draft in a simple text editor
- Iterations...
  - Require extreme concentration
  - Maybe do it every morning for an hour
- Kill your darlings: don't be afraid of deleting parts that you invested efforts into but don't fit anymore

Revising is Difficult

- When reading, it is hard to view your own text form the perspective of a reader

Revising is Difficult

- When reading, it is hard to view your own text form the perspective of a reader
- You can try the following:

Revising is Difficult

- When reading, it is hard to view your own text form the perspective of a reader
- You can try the following:
- **Dead Trees**: Print your paper and read at another physical location than you write

Revising is Difficult

- When reading, it is hard to view your own text form the perspective of a reader
- You can try the following:
- **Dead Trees**: Print your paper and read at another physical location than you write
- **Random**: Don't read from start. Jump to random sections and start revising

Revising is Difficult

- When reading, it is hard to view your own text form the perspective of a reader
- You can try the following:
- **Dead Trees**: Print your paper and read at another physical location than you write
- **Random**: Don't read from start. Jump to random sections and start revising
- **Read Aloud**: Helps catching issues you are otherwise blind to

Revising is Difficult

- When reading, it is hard to view your own text form the perspective of a reader
- You can try the following:
- **Dead Trees**: Print your paper and read at another physical location than you write
- **Random**: Don't read from start. Jump to random sections and start revising
- **Read Aloud**: Helps catching issues you are otherwise blind to
- Still: **Feedback** from other persons is most valuable!

Title

- The title of a paper is the first impression people have **and will use for filtering**

Title

- The title of a paper is the first impression people have **and will use for filtering**
- Should be catchy, intriguing, and **meaningful**

Title

- The title of a paper is the first impression people have **and will use for filtering**
- Should be catchy, intriguing, and **meaningful**
- Naming things is very important: (components, algorithms, methods, products, ...)

Title

- The title of a paper is the first impression people have **and will use for filtering**
- Should be catchy, intriguing, and **meaningful**
- Naming things is very important: (components, algorithms, methods, products, ...)
  - Makes it easier to talk about things

Title

- The title of a paper is the first impression people have **and will use for filtering**
- Should be catchy, intriguing, and **meaningful**
- Naming things is very important: (components, algorithms, methods, products, ...)
  - Makes it easier to talk about things
  - Makes it easier to think about things

Title

- The title of a paper is the first impression people have **and will use for filtering**
- Should be catchy, intriguing, and **meaningful**
- Naming things is very important: (components, algorithms, methods, products, ...)
  - Makes it easier to talk about things
  - Makes it easier to think about things
  - Makes it easier to remember things

Title

- The title of a paper is the first impression people have **and will use for filtering**
- Should be catchy, intriguing, and **meaningful**
- Naming things is very important: (components, algorithms, methods, products, ...)
  - Makes it easier to talk about things
  - Makes it easier to think about things
  - Makes it easier to remember things
- The title should give the reader an idea what your paper is about

Abstract

- Readers will only read title and (maybe) abstract to decide whether to read your paper

Abstract

- Readers will only read title and (maybe) abstract to decide whether to read your paper
- Abstract should summarize the whole paper

Abstract

- Readers will only read title and (maybe) abstract to decide whether to read your paper
- Abstract should summarize the whole paper
  - What is it about

Abstract

- Readers will only read title and (maybe) abstract to decide whether to read your paper
- Abstract should summarize the whole paper
  - What is it about
  - Motivation

Abstract

- Readers will only read title and (maybe) abstract to decide whether to read your paper
- Abstract should summarize the whole paper
  - What is it about
  - Motivation
  - Core ideas, methods, and solutions

Abstract

- Readers will only read title and (maybe) abstract to decide whether to read your paper
- Abstract should summarize the whole paper
  - What is it about
  - Motivation
  - Core ideas, methods, and solutions
  - Impact, conclusions, and findings

Abstract

- Readers will only read title and (maybe) abstract to decide whether to read your paper
- Abstract should summarize the whole paper
  - What is it about
  - Motivation
  - Core ideas, methods, and solutions
  - Impact, conclusions, and findings
- Should be as short and expressive as possible

Goal Oriented Writing

- Your text has to **provide value** to the reader
  Why bother writing it otherwise?

Goal Oriented Writing

- Your text has to **provide value** to the reader
  Why bother writing it otherwise?
- You cannot express everything you have to say in one text

Goal Oriented Writing

- Your text has to **provide value** to the reader
  Why bother writing it otherwise?
- You cannot express everything you have to say in one text
- **Define one clear goal** that your text has to achieve (e.g. answer a research question)

Goal Oriented Writing

- Your text has to **provide value** to the reader
  Why bother writing it otherwise?
- You cannot express everything you have to say in one text
- **Define one clear goal** that your text has to achieve (e.g. answer a research question)
- **Kill your darlings**: Remove text that does not help to achieve the goal
  See Chekhov's Gun: (If the gun is there it will be important)

Goal Oriented Writing

- Your text has to **provide value** to the reader
  Why bother writing it otherwise?
- You cannot express everything you have to say in one text
- **Define one clear goal** that your text has to achieve (e.g. answer a research question)
- **Kill your darlings**: Remove text that does not help to achieve the goal
  See Chekhov's Gun: (If the gun is there it will be important)
- **Know your audience**: The text should be optimized for the reader

Structure Top Down

- It's natural to start writing small pieces of text

Structure Top Down

- It's natural to start writing small pieces of text
  - You need to figure out how to combine them

Structure Top Down

- It's natural to start writing small pieces of text
  - You need to figure out how to combine them
  - Makes it hard to get a good overall structure

Structure Top Down

- It's natural to start writing small pieces of text
  - You need to figure out how to combine them
  - Makes it hard to get a good overall structure
- Plan your paper top down

Structure Top Down

- It's natural to start writing small pieces of text
  - You need to figure out how to combine them
  - Makes it hard to get a good overall structure
- Plan your paper top down
  - Start with a very rough outline

Structure Top Down

- It's natural to start writing small pieces of text
  - You need to figure out how to combine them
  - Makes it hard to get a good overall structure
- Plan your paper top down
  - Start with a very rough outline
  - Iteratively specify section contents with increasing detail

Structure Top Down

- It's natural to start writing small pieces of text
  - You need to figure out how to combine them
  - Makes it hard to get a good overall structure
- Plan your paper top down
  - Start with a very rough outline
  - Iteratively specify section contents with increasing detail
  - Works for whole paper as well as individual sections

Write Sections with the Onion Principle

- Sometimes you need to explain several independent things to lead to the next point

Write Sections with the Onion Principle

- Sometimes you need to explain several independent things to lead to the next point
- Use the onion principle:

Write Sections with the Onion Principle

- Sometimes you need to explain several independent things to lead to the next point
- Use the onion principle:
  - Start with the main point / complete overview but do not go into detail

Write Sections with the Onion Principle

- Sometimes you need to explain several independent things to lead to the next point
- Use the onion principle:
  - Start with the main point / complete overview but do not go into detail
  - Add several layers: Each layer may go down further into detail

Write Sections with the Onion Principle

- Sometimes you need to explain several independent things to lead to the next point
- Use the onion principle:
  - Start with the main point / complete overview but do not go into detail
  - Add several layers: Each layer may go down further into detail
- This holds for sections as well as the whole paper:

Write Sections with the Onion Principle

- Sometimes you need to explain several independent things to lead to the next point
- Use the onion principle:
  - Start with the main point / complete overview but do not go into detail
  - Add several layers: Each layer may go down further into detail
- This holds for sections as well as the whole paper:
  - Title

Write Sections with the Onion Principle

- Sometimes you need to explain several independent things to lead to the next point
- Use the onion principle:
  - Start with the main point / complete overview but do not go into detail
  - Add several layers: Each layer may go down further into detail
- This holds for sections as well as the whole paper:
  - Title
  - Abstract

Write Sections with the Onion Principle

- Sometimes you need to explain several independent things to lead to the next point
- Use the onion principle:
  - Start with the main point / complete overview but do not go into detail
  - Add several layers: Each layer may go down further into detail
- This holds for sections as well as the whole paper:
  - Title
  - Abstract
  - Introduction

Write Sections with the Onion Principle

- Sometimes you need to explain several independent things to lead to the next point
- Use the onion principle:
  - Start with the main point / complete overview but do not go into detail
  - Add several layers: Each layer may go down further into detail
- This holds for sections as well as the whole paper:
  - Title
  - Abstract
  - Introduction
  - Main part (may contain several layers)

Avoid the Wall of Text

- Reader looses attention when reading (or even seeing) long text

Avoid the Wall of Text

- Reader looses attention when reading (or even seeing) long text
- Use subsections

Avoid the Wall of Text

- Reader looses attention when reading (or even seeing) long text
- Use subsections
- Use small titles

Avoid the Wall of Text

- Reader looses attention when reading (or even seeing) long text
- Use subsections
- Use small titles
- Use short paragraphs

Avoid the Wall of Text

- Reader looses attention when reading (or even seeing) long text
- Use subsections
- Use small titles
- Use short paragraphs
- Use bullet points

Examples are Crucial

- Help reader to check whether they understood something

Examples are Crucial

- Help reader to check whether they understood something
- Should be minimal but interesting

Examples are Crucial

- Help reader to check whether they understood something
- Should be minimal but interesting
- May be used in whole paper, also as motivation

Examples are Crucial

- Help reader to check whether they understood something
- Should be minimal but interesting
- May be used in whole paper, also as motivation
- Best case: One simple, but sufficient example used throughout the entire paper

Examples are Crucial

- Help reader to check whether they understood something
- Should be minimal but interesting
- May be used in whole paper, also as motivation
- Best case: One simple, but sufficient example used throughout the entire paper
- *Finding good examples is extremly difficult, but worth it!*

The Baseline Idea

- When writing about an approach it may be useful to compare against a well-known baseline

The Baseline Idea

- When writing about an approach it may be useful to compare against a well-known baseline
- Problems of baseline lead to solution

The Baseline Idea

- When writing about an approach it may be useful to compare against a well-known baseline
- Problems of baseline lead to solution
- This might help with the whole story of a paper

The Baseline Idea

- When writing about an approach it may be useful to compare against a well-known baseline
- Problems of baseline lead to solution
- This might help with the whole story of a paper
- Example:

The Baseline Idea

- When writing about an approach it may be useful to compare against a well-known baseline
- Problems of baseline lead to solution
- This might help with the whole story of a paper
- Example:
  - Linear search on data

The Baseline Idea

- When writing about an approach it may be useful to compare against a well-known baseline
- Problems of baseline lead to solution
- This might help with the whole story of a paper
- Example:
  - Linear search on data
  - Improvement: Sort data + perform binary search

The Baseline Idea

- When writing about an approach it may be useful to compare against a well-known baseline
- Problems of baseline lead to solution
- This might help with the whole story of a paper
- Example:
  - Linear search on data
  - Improvement: Sort data + perform binary search
  - Improvement: Use sorted search-tree

Repeat Important Points

- Readers may read superficially or only parts of the text

Repeat Important Points

- Readers may read superficially or only parts of the text
- Important points should be clear to them too

Repeat Important Points

- Readers may read superficially or only parts of the text
- Important points should be clear to them too
- Repeat them in: (abstract), introduction, main part, conclusion

Why is Good Writing so Difficult?

- Text is linear - Humans think associatively

Why is Good Writing so Difficult?

- Text is linear - Humans think associatively
- Curse of knowledge:

Why is Good Writing so Difficult?

- Text is linear - Humans think associatively
- Curse of knowledge:
  - Once you understand something, it is hard to identify with someone who doesn't

Example: Clear Sentences

- Our lack of knowledge about local conditions precluded determination of committee action effectiveness in fund allocation to those areas in greatest need of assistance. ❌

Example: Clear Sentences

- Our lack of knowledge about local conditions precluded determination of committee action effectiveness in fund allocation to those areas in greatest need of assistance. ❌
- Because we knew nothing about local conditions, we could not determine how effectively the committee had allocated funds to areas that most needed assistance. ✅

Example: Clear Sentences

- Our lack of knowledge about local conditions precluded determination of committee action effectiveness in fund allocation to those areas in greatest need of assistance. ❌
- Because we knew nothing about local conditions, we could not determine how effectively the committee had allocated funds to areas that most needed assistance. ✅
- Why is the second version easier to understand?

Example: Clear Sentences

- Our lack of knowledge about local conditions precluded determination of committee action effectiveness in fund allocation to those areas in greatest need of assistance. ❌
- Because we knew nothing about local conditions, we could not determine how effectively the committee had allocated funds to areas that most needed assistance. ✅
- Why is the second version easier to understand?
- Unclear who does what!

Clear Sentences Like Stories

- People think in stories

Clear Sentences Like Stories

- People think in stories
- **Who does what**?

Clear Sentences Like Stories

- People think in stories
- **Who does what**?
  - Clear actors (subjects)

Clear Sentences Like Stories

- People think in stories
- **Who does what**?
  - Clear actors (subjects)
  - Clear actions (verbs)

Actors - Subjects

- Actors should be clear: "Knuth developed TeX" ✅

Actors - Subjects

- Actors should be clear: "Knuth developed TeX" ✅
- Things or representations can be actors:

Actors - Subjects

- Actors should be clear: "Knuth developed TeX" ✅
- Things or representations can be actors:
  - "The compiler tells you..." ✅,

Actors - Subjects

- Actors should be clear: "Knuth developed TeX" ✅
- Things or representations can be actors:
  - "The compiler tells you..." ✅,
  - "The community observes..." ✅

Actors - Subjects

- Actors should be clear: "Knuth developed TeX" ✅
- Things or representations can be actors:
  - "The compiler tells you..." ✅,
  - "The community observes..." ✅
- Avoid hidden actors:

Actors - Subjects

- Actors should be clear: "Knuth developed TeX" ✅
- Things or representations can be actors:
  - "The compiler tells you..." ✅,
  - "The community observes..." ✅
- Avoid hidden actors:
  - "In your paper there is an explanation for ..." ❌

Actors - Subjects

- Actors should be clear: "Knuth developed TeX" ✅
- Things or representations can be actors:
  - "The compiler tells you..." ✅,
  - "The community observes..." ✅
- Avoid hidden actors:
  - "In your paper there is an explanation for ..." ❌

Actors - Subjects

- Actors should be clear: "Knuth developed TeX" ✅
- Things or representations can be actors:
  - "The compiler tells you..." ✅,
  - "The community observes..." ✅
- Avoid hidden actors:
  - "In your paper there is an explanation for ..." ❌
  - "You explain ... in your paper." ✅

Verbs and Adjectives

- Avoid nouns that are hidden verbs:

Verbs and Adjectives

- Avoid nouns that are hidden verbs:
  - "We made an analysis of the behavior of ..." ❌

Verbs and Adjectives

- Avoid nouns that are hidden verbs:
  - "We made an analysis of the behavior of ..." ❌

Verbs and Adjectives

- Avoid nouns that are hidden verbs:
  - "We made an analysis of the behavior of ..." ❌
  - "We analyzed the behavior of ..." ✅

Verbs and Adjectives

- Avoid nouns that are hidden verbs:
  - "We made an analysis of the behavior of ..." ❌
  - "We analyzed the behavior of ..." ✅
- Avoid nouns that are hidden adjectives:

Verbs and Adjectives

- Avoid nouns that are hidden verbs:
  - "We made an analysis of the behavior of ..." ❌
  - "We analyzed the behavior of ..." ✅
- Avoid nouns that are hidden adjectives:
  - "The implementation of this data structure poses difficulties" ❌

Verbs and Adjectives

- Avoid nouns that are hidden verbs:
  - "We made an analysis of the behavior of ..." ❌
  - "We analyzed the behavior of ..." ✅
- Avoid nouns that are hidden adjectives:
  - "The implementation of this data structure poses difficulties" ❌

Verbs and Adjectives

- Avoid nouns that are hidden verbs:
  - "We made an analysis of the behavior of ..." ❌
  - "We analyzed the behavior of ..." ✅
- Avoid nouns that are hidden adjectives:
  - "The implementation of this data structure poses difficulties" ❌
  - "The implementation of this data structure is difficult" ✅

Not All Nouns are Bad

Use nouns to:

- Refer to the previous sentence:

Not All Nouns are Bad

Use nouns to:

- Refer to the previous sentence:
  - "These *arguments* are convincing" ✅

Not All Nouns are Bad

Use nouns to:

- Refer to the previous sentence:
  - "These *arguments* are convincing" ✅
- Create an object instead of a complicated construct:

Not All Nouns are Bad

Use nouns to:

- Refer to the previous sentence:
  - "These *arguments* are convincing" ✅
- Create an object instead of a complicated construct:
  - "I do not understand either his *meaning* or her *intention*" ✅

Not All Nouns are Bad

Use nouns to:

- Refer to the previous sentence:
  - "These *arguments* are convincing" ✅
- Create an object instead of a complicated construct:
  - "I do not understand either his *meaning* or her *intention*" ✅
  - "I do not understand either what he means or what she intends" ❌

Not All Nouns are Bad

Use nouns to:

- Refer to the previous sentence:
  - "These *arguments* are convincing" ✅
- Create an object instead of a complicated construct:
  - "I do not understand either his *meaning* or her *intention*" ✅
  - "I do not understand either what he means or what she intends" ❌
- Identify the correct actor (subject):

Not All Nouns are Bad

Use nouns to:

- Refer to the previous sentence:
  - "These *arguments* are convincing" ✅
- Create an object instead of a complicated construct:
  - "I do not understand either his *meaning* or her *intention*" ✅
  - "I do not understand either what he means or what she intends" ❌
- Identify the correct actor (subject):
  - "The fact that I set the correct flags was crucial" ❌

Not All Nouns are Bad

Use nouns to:

- Refer to the previous sentence:
  - "These *arguments* are convincing" ✅
- Create an object instead of a complicated construct:
  - "I do not understand either his *meaning* or her *intention*" ✅
  - "I do not understand either what he means or what she intends" ❌
- Identify the correct actor (subject):
  - "The fact that I set the correct flags was crucial" ❌
  - "My correct *choice* of flags was crucial" ✅

Not All Nouns are Bad

Use nouns to:

- Refer to the previous sentence:
  - "These *arguments* are convincing" ✅
- Create an object instead of a complicated construct:
  - "I do not understand either his *meaning* or her *intention*" ✅
  - "I do not understand either what he means or what she intends" ❌
- Identify the correct actor (subject):
  - "The fact that I set the correct flags was crucial" ❌
  - "My correct *choice* of flags was crucial" ✅
- Name repeated concepts (create new actors):

Not All Nouns are Bad

Use nouns to:

- Refer to the previous sentence:
  - "These *arguments* are convincing" ✅
- Create an object instead of a complicated construct:
  - "I do not understand either his *meaning* or her *intention*" ✅
  - "I do not understand either what he means or what she intends" ❌
- Identify the correct actor (subject):
  - "The fact that I set the correct flags was crucial" ❌
  - "My correct *choice* of flags was crucial" ✅
- Name repeated concepts (create new actors):
  - "*Anyblob* is a download manager" ✅

More Useful Methods

- Be careful with passive voice

More Useful Methods

- Be careful with passive voice
- Write specific and concrete

More Useful Methods

- Be careful with passive voice
- Write specific and concrete
- Fewer prepositions

More Useful Methods

- Be careful with passive voice
- Write specific and concrete
- Fewer prepositions
- Shorter sentences

More Useful Methods

- Be careful with passive voice
- Write specific and concrete
- Fewer prepositions
- Shorter sentences
- Logical order

More Useful Methods

- Be careful with passive voice
- Write specific and concrete
- Fewer prepositions
- Shorter sentences
- Logical order
- Clear logical relationships

Connected Information Flow

- Sentences often start with a transition or evaluation:

Connected Information Flow

- Sentences often start with a transition or evaluation:
  - Hence, but, fortunately, importantly

Connected Information Flow

- Sentences often start with a transition or evaluation:
  - Hence, but, fortunately, importantly
- The beginning of a sentence should contain known easy to recognize things

Connected Information Flow

- Sentences often start with a transition or evaluation:
  - Hence, but, fortunately, importantly
- The beginning of a sentence should contain known easy to recognize things
- The end of a sentence should contain newest and significant information

Connected Information Flow

- Sentences often start with a transition or evaluation:
  - Hence, but, fortunately, importantly
- The beginning of a sentence should contain known easy to recognize things
- The end of a sentence should contain newest and significant information
- You automatically stress the end of a sentence

Balance Clarity and Flow

- Sometimes clarity and flow are in conflict

1. "A black hole is created by the collapse of a dead star into a point perhaps no larger than a marble."
2. "The collapse of a dead star into a point perhaps no larger than a marble creates a black hole."

Balance Clarity and Flow

- Sometimes clarity and flow are in conflict

1. "A black hole is created by the collapse of a dead star into a point perhaps no larger than a marble."
2. "The collapse of a dead star into a point perhaps no larger than a marble creates a black hole."

- **(2)** is more clear
- "Some astonishing questions about the nature of the universe have been raised by scientists exploring the nature of black holes in space. **(1) or (2)** So much matter compressed into so little volume changes the fabric of space around it in profoundly puzzling ways."

Balance Clarity and Flow

- Sometimes clarity and flow are in conflict

1. "A black hole is created by the collapse of a dead star into a point perhaps no larger than a marble."
2. "The collapse of a dead star into a point perhaps no larger than a marble creates a black hole."

- **(2)** is more clear
- "Some astonishing questions about the nature of the universe have been raised by scientists exploring the nature of black holes in space. **(1) or (2)** So much matter compressed into so little volume changes the fabric of space around it in profoundly puzzling ways."
- **(1)** connects better to the previous sentence: "black hole"
- **(1)** connects better to the next sentence: "... no larger than a marble." "So much matter compressed"

Be Concise

- Use as few words as possible to express what you mean

Be Concise

- Use as few words as possible to express what you mean
  - "In my personal opinion, we must listen to and think over in a punctilious manner each and every suggestion that is offered to us." ❌

Be Concise

- Use as few words as possible to express what you mean
  - "In my personal opinion, we must listen to and think over in a punctilious manner each and every suggestion that is offered to us." ❌
  - "We must consider each suggestion carefully." ✅

Be Concise

- Use as few words as possible to express what you mean
  - "In my personal opinion, we must listen to and think over in a punctilious manner each and every suggestion that is offered to us." ❌
  - "We must consider each suggestion carefully." ✅
- Another example:

Be Concise

- Use as few words as possible to express what you mean
  - "In my personal opinion, we must listen to and think over in a punctilious manner each and every suggestion that is offered to us." ❌
  - "We must consider each suggestion carefully." ✅
- Another example:
  - "Imagine a picture of someone engaged in the activity of trying to learn the rules for playing the game of chess." ❌

Be Concise

- Use as few words as possible to express what you mean
  - "In my personal opinion, we must listen to and think over in a punctilious manner each and every suggestion that is offered to us." ❌
  - "We must consider each suggestion carefully." ✅
- Another example:
  - "Imagine a picture of someone engaged in the activity of trying to learn the rules for playing the game of chess." ❌
  - "Imagine someone trying to learn the rules of chess." ✅

Avoid Wordy Phrases

- Avoid wordy phrases:

❌	✅
the reason for, due to the fact that, this is	because, since, why
despite the fact that, regardless of the fact that	although, even though
in the event that	if
on the occasion of	when
it is crucial that	must, should
is able to	can
it is possible that	may, might, can, could
prior to	before
does not have	lacks

Figures Matter

- Figures are an integral part of communication

Amazon Redshift and the Case for Simpler Data Warehouses

Anurag Gupta, Deepak Agarwal, Derek Tan, Jakub Kulesza, Rahul Pathak, Stefano Stefani, Vidhya Srinivasan
Amazon Web Services

Abstract

Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse solution that makes it simple and cost-effective to efficiently analyze large volumes of data using existing business intelligence tools. Since launching in February 2013, it has been Amazon Web Service's (AWS) fastest growing service, with many thousands of customers and many petabytes of data under management.

Amazon Redshift's pace of adoption has been a surprise to many participants in the data warehousing community. While Amazon Redshift was priced disruptively at launch, available for as little as $1000/TB/year, there are many open-source data warehousing technologies and many commercial data warehousing engines that provide free editions for development or under some usage limit. While Amazon Redshift provides a modern MPP, columnar, scale-out architecture, so too do many other data warehousing engines. And, while Amazon Redshift is available in the AWS cloud, one can build data warehouses using EC2 instances and the database engine of one's choice with either local or network-attached storage.

In this paper, we discuss an oft-overlooked differentiating characteristic of Amazon Redshift – simplicity. Our goal with Amazon Redshift was not to compete with other data warehousing engines, but to compete with non-consumption. We believe the vast majority of data is collected but not analyzed. We believe, while most database vendors target larger enterprises, there is little correlation in today's economy between data set size and company size. And, we believe the models used to procure and consume analytics technology need to support experimentation and evaluation. Amazon Redshift was designed to bring data warehousing to a mass market by making it easy to buy, easy to tune and easy to manage while also being fast and cost-effective.

1. Introduction

Many companies augment their transaction-processing database systems with data warehouses for reporting and analysis. Analysts estimate the data warehouse market segment at 1/3 of the overall relational database market segment ($14B vs. $42B for software licenses and support), with an 8-11% compound annual growth rate (CAGR)[1]. While this is a strong growth rate for a large, mature market, over the past ten years, analysts also estimate data storage at a typical enterprise growing at 30-40%. CAGR. Over the past 12-18 months, new market research has begun to show an increase to 50-60%, with data doubling in size every 20 months.

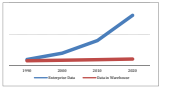


Figure 1: Data Analysis Gap in the Enterprise [10]

This implies most data in an enterprise is "dark data"[2]: data that is collected but not easily analyzed. We see this as unfortunate. If our customers didn't see this data as having value, they would not retain it. Many companies are trying to become increasingly data-driven. And yet, not only is most data already dark, the overall data landscape is only getting darker. Storing this data in NoSQL stores and/or Hadoop is one way to bridge the gap for certain use cases. However it doesn't address all scenarios.

In our discussions with customers, we found the "analysis gap" between data being collected and data available for analysis as due to four major causes.

1. Cost – Most commercial database solutions capable of analyzing data at scale require significant up-front expense. This is hard to justify for large datasets with unclear value.
2. Complexity – Database provisioning, maintenance, backup, and tuning are complex tasks requiring specialized skills. They require IT involvement and cannot easily be performed by line of business data scientists or analysts.
3. Performance – It is difficult to grow a data warehouse without negatively impacting query performance. Once built, IT teams sometimes discourage augmenting data or adding queries as a way of protecting current reporting SLAs.
4. Rigidity – Most databases work best on highly structured relational data. But a large and increasing percentage of data consists of machine-generated logs that mutate over time, audio and video, not readily accessible to relational analysis.

We see each of the above issues only increasing with data set size. To take one large-scale customer example, the Amazon Retail team collects about 5 billion web log records daily (2TB/day,

SIGMOD'15, May 31–June 4, 2015, Melbourne, Victoria, Australia
ACM 978-1-4503-2758-9/15/05.
http://dx.doi.org/10.1145/2723372.2742795

[1] Forecast data combined from IDC, Gartner, and 451 Research.
[2] http://www.gartner.com/it-glossary/dark-data

First page figure ✅

Figures Matter

- Figures are an integral part of communication
- Many people only look at figures and captions

Amazon Redshift and the Case for Simpler Data Warehouses

Anurag Gupta, Deepak Agarwal, Derek Tan, Jakub Kulesza, Rahul Pathak, Stefano Stefani, Vidhya Srinivasan
Amazon Web Services

Abstract

Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse solution that makes it simple and cost-effective to efficiently analyze large volumes of data using existing business intelligence tools. Since launching in February 2013, it has been Amazon Web Service's (AWS) fastest growing service, with many thousands of customers and many petabytes of data under management.

Amazon Redshift's pace of adoption has been a surprise to many participants in the data warehousing community. While Amazon Redshift was priced disruptively at launch, available for as little as $1000/TB/year, there are many open-source data warehousing technologies and many commercial data warehousing engines that provide free editions for development or under some usage limit. While Amazon Redshift provides a modern MPP, columnar, scale-out architecture, so too do many other data warehousing engines. And, while Amazon Redshift is available in the AWS cloud, one can build data warehouses using EC2 instances and the database engine of one's choice with either local or network-attached storage.

In this paper, we discuss an oft-overlooked differentiating characteristic of Amazon Redshift – simplicity. Our goal with Amazon Redshift was not to compete with other data warehousing engines, but to compete with non-consumption. We believe the vast majority of data is collected but not analyzed. We believe, while most database vendors target larger enterprises, there is little correlation in today's economy between data set size and company size. And, we believe the models used to procure and consume analytics technology need to support experimentation and evaluation. Amazon Redshift was designed to bring data warehousing to a mass market by making it easy to buy, easy to tune and easy to manage while also being fast and cost-effective.

1. Introduction

Many companies augment their transaction-processing database systems with data warehouses for reporting and analysis. Analysts estimate the data warehouse market segment at 1/3 of the overall relational database market segment ($14B vs. $42B for software licenses and support), with an 8-11% compound annual growth rate (CAGR)[1]. While this is a strong growth rate for a large, mature market, over the past ten years, analysts also estimate data storage at a typical enterprise growing at 30-40%. CAGR. Over the past 12-18 months, new market research has begun to show an increase to 50-60%, with data doubling in size every 20 months.

Figure 1: Data Analysis Gap in the Enterprise [10]

This implies most data in an enterprise is "dark data"[2]: data that is collected but not easily analyzed. We see this as unfortunate. If our customers didn't see this data as having value, they would not retain it. Many companies are trying to become increasingly data-driven. And yet, not only is most data already dark, the overall data landscape is only getting darker. Storing this data in NoSQL stores and/or Hadoop is one way to bridge the gap for certain use cases. However it doesn't address all scenarios.

In our discussions with customers, we found the "analysis gap" between data being collected and data available for analysis as due to four major causes.

1. Cost – Most commercial database solutions capable of analyzing data at scale require significant up-front expense. This is hard to justify for large datasets with unclear value.
2. Complexity – Database provisioning, maintenance, backup, and tuning are complex tasks requiring specialized skills. They require IT involvement and cannot easily be performed by line of business data scientists or analysts.
3. Performance – It is difficult to grow a data warehouse without negatively impacting query performance. Once built, IT teams sometimes discourage augmenting data or adding queries as a way of protecting current reporting SLAs.
4. Rigidity – Most databases work best on highly structured relational data. But a large and increasing percentage of data consists of machine-generated logs that mutate over time, audio and video, not readily accessible to relational analysis.

We see each of the above issues only increasing with data set size. To take one large-scale customer example, the Amazon Retail team collects about 5 billion web log records daily (2TB/day,

[1] Forecast data combined from IDC, Gartner, and 451 Research.
[2] http://www.gartner.com/it-glossary/dark-data

First page figure ✅

Figures Matter

- Figures are an integral part of communication
- Many people only look at figures and captions
- Papers are recognized by graphs: Add a first page figure

Amazon Redshift and the Case for Simpler Data Warehouses

Anurag Gupta, Deepak Agarwal, Derek Tan, Jakub Kulesza, Rahul Pathak, Stefano Stefani, Vidhya Srinivasan
Amazon Web Services

Abstract

Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse solution that makes it simple and cost-effective to efficiently analyze large volumes of data using existing business intelligence tools. Since launching in February 2013, it has been Amazon Web Service's (AWS) fastest growing service, with many thousands of customers and many petabytes of data under management.

Amazon Redshift's pace of adoption has been a surprise to many participants in the data warehousing community. While Amazon Redshift was priced disruptively at launch, available for as little as $1000/TB/year, there are many open-source data warehousing technologies and many commercial data warehousing engines that provide free editions for development or under some usage limit. While Amazon Redshift provides a modern MPP, columnar, scale-out architecture, so too do many other data warehousing engines. And, while Amazon Redshift is available in the AWS cloud, one can build data warehouses using EC2 instances and the database engine of one's choice with either local or network-attached storage.

In this paper, we discuss an oft-overlooked differentiating characteristic of Amazon Redshift – simplicity. Our goal with Amazon Redshift was not to compete with other data warehousing engines, but to compete with non-consumption. We believe the vast majority of data is collected but not analyzed. We believe, while most database vendors target larger enterprises, there is little correlation in today's economy between data set size and company size. And, we believe the models used to procure and consume analytics technology need to support experimentation and evaluation. Amazon Redshift was designed to bring data warehousing to a mass market by making it easy to buy, easy to tune and easy to manage while also being fast and cost-effective.

1. Introduction

Many companies augment their transaction-processing database systems with data warehouses for reporting and analysis. Analysts estimate the data warehouse market segment at 1/3 of the overall relational database market segment ($4-9B vs. $40B for software licenses and support), with an 8-11% compound annual growth rate (CAGR)[1]. While this is a strong growth rate for a large, mature market, over the past ten years, analysts also estimate data storage at a typical enterprise growing at 30-40% CAGR. Over the past 12-18 months, new market research has begun to show an increase to 50-60%, with data doubling in size every 20 months.

Figure 1: Data Analysis Gap in the Enterprise [10]

This implies most data in an enterprise is "dark data"[2]: data that is collected but not easily analyzed. We see this as unfortunate. If our customers didn't see this data as having value, they would not retain it. Many companies are trying to become increasingly data-driven. And yet, not only is most data already dark, the overall data landscape is only getting darker. Storing this data in NoSQL stores and/or Hadoop in one way to bridge the gap for certain use cases. However it doesn't address all scenarios.

In our discussions with customers, we found the "analysis gap" between data being collected and data available for analysis as due to four major causes.

1. Cost – Most commercial database solutions capable of analyzing data at scale require significant up-front expense. This is hard to justify for large datasets with unclear value.
2. Complexity – Database provisioning, maintenance, backup, and tuning are complex tasks requiring specialized skills. They require IT involvement and cannot easily be performed by line of business data scientists or analysts.
3. Performance – It is difficult to grow a data warehouse without negatively impacting query performance. Once built, IT teams sometimes discourage augmenting data or adding queries as a way of protecting current reporting SLAs.
4. Rigidity – Most databases work best on highly structured relational data. But a large and increasing percentage of data consists of machine-generated logs that mutate over time, audio and video, not readily accessible to relational analysis.

We see each of the above issues only increasing with data set size. To take one large-scale customer example, the Amazon Retail team collects about 5 billion web log records daily (2TB/day,

[1] Forecast data combined from IDC, Gartner, and 451 Research.
[2] http://www.gartner.com/it-glossary/dark-data

First page figure ✅

Figures Matter

- Figures are an integral part of communication
- Many people only look at figures and captions
- Papers are recognized by graphs: Add a first page figure
- Figures are equally important as text

Amazon Redshift and the Case for Simpler Data Warehouses

Anurag Gupta, Deepak Agarwal, Derek Tan, Jakub Kulesza, Rahul Pathak, Stefano Stefani, Vidhya Srinivasan
Amazon Web Services

Abstract

Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse solution that makes it simple and cost-effective to efficiently analyze large volumes of data using existing business intelligence tools. Since launching in February 2013, it has been Amazon Web Service's (AWS) fastest growing service, with many thousands of customers and many petabytes of data under management.

Amazon Redshift's pace of adoption has been a surprise to many participants in the data warehousing community. While Amazon Redshift was priced disruptively at launch, available for as little as $1000/TB/year, there are many open-source data warehousing technologies and many commercial data warehousing engines that provide free editions for development or under some usage limit. While Amazon Redshift provides a modern MPP, columnar, scale-out architecture, so too do many other data warehousing engines. And, while Amazon Redshift is available in the AWS cloud, one can build data warehouses using EC2 instances and the database engine of one's choice with either local or network-attached storage.

In this paper, we discuss an oft-overlooked differentiating characteristic of Amazon Redshift – simplicity. Our goal with Amazon Redshift was not to compete with other data warehousing engines, but to compete with non-consumption. We believe the vast majority of data is collected but not analyzed. We believe, while most database vendors target larger enterprises, there is little correlation in today's economy between data set size and company size. And, we believe the models used to procure and consume analytics technology need to support experimentation and evaluation. Amazon Redshift was designed to bring data warehousing to a mass market by making it easy to buy, easy to tune and easy to manage while also being fast and cost-effective.

1. Introduction

Many companies augment their transaction-processing database systems with data warehouses for reporting and analysis. Analysts estimate the data warehouse market segment at 1/3 of the overall relational database market segment ($4-6B vs. $40B for software licenses and support), with an 8-11% compound annual growth rate (CAGR)[1]. While this is a strong growth rate for a large, mature market, over the past ten years, analysts also estimate data storage at a typical enterprise growing at 30-40% CAGR. Over the past 12-18 months, new market research has begun to show an increase to 50-60%, with data doubling in size every 20 months.

Figure 1: Data Analysis Gap in the Enterprise [10]

This implies most data in an enterprise is "dark data"[2]: data that is collected but not easily analyzed. We see this as unfortunate. If our customers didn't see this data as having value, they would not retain it. Many companies are trying to become increasingly data-driven. And yet, not only is most data already dark, the overall data landscape is only getting darker. Storing this data in NoSQL stores and/or Hadoop in one way to bridge the gap for certain use cases. However it doesn't address all scenarios.

In our discussions with customers, we found the "analysis gap" between data being collected and data available for analysis as due to four major causes:

1. Cost – Most commercial database solutions capable of analyzing data at scale require significant up-front expense. This is hard to justify for large datasets with unclear value.
2. Complexity – Database provisioning, maintenance, backup, and tuning are complex tasks requiring specialized skills. They require IT involvement and cannot easily be performed by line of business data scientists or analysts.
3. Performance – It is difficult to grow a data warehouse without negatively impacting query performance. Once built, IT teams sometimes discourage augmenting data or adding queries as a way of protecting current reporting SLAs.
4. Rigidity – Most databases work best on highly structured relational data. But a large and increasing percentage of data consists of machine-generated logs that mutate over time, audio and video, not readily accessible to relational analysis.

We see each of the above issues only increasing with data set size. To take one large-scale customer example, the Amazon Retail team collects about 5 billion web log records daily (2TB/day,

[1] Forecast data combined from IDC, Gartner, and 451 Research.
[2] http://www.gartner.com/it-glossary/dark-data

First page figure ✅

Figures Matter

- Figures are an integral part of communication
- Many people only look at figures and captions
- Papers are recognized by graphs: Add a first page figure
- Figures are equally important as text
- Text should discuss the takeaways of a figure

Amazon Redshift and the Case for Simpler Data Warehouses

Anurag Gupta, Deepak Agarwal, Derek Tan, Jakub Kulesza, Rahul Pathak, Stefano Stefani, Vidhya Srinivasan

Amazon Web Services

Abstract

Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse solution that makes it simple and cost-effective to efficiently analyze large volumes of data using existing business intelligence tools. Since launching in February 2013, it has been Amazon Web Service's (AWS) fastest growing service, with many thousands of customers and many petabytes of data under management.

Amazon Redshift's pace of adoption has been a surprise to many participants in the data warehousing community. While Amazon Redshift was priced disruptively at launch, available for as little as $1000/TB/year, there are many open-source data warehousing technologies and many commercial data warehousing engines that provide free editions for development or under some usage limit. While Amazon Redshift provides a modern MPP, columnar, scale-out architecture, so too do many other data warehousing engines. And, while Amazon Redshift is available in the AWS cloud, one can build data warehouses using EC2 instances and the database engine of one's choice with either local or network-attached storage.

In this paper, we discuss an oft-overlooked differentiating characteristic of Amazon Redshift – simplicity. Our goal with Amazon Redshift was not to compete with other data warehousing engines, but to compete with non-consumption. We believe the vast majority of data is collected but not analyzed. We believe, while most database vendors target larger enterprises, there is little correlation in today's economy between data set size and company size. And, we believe the models used to procure and consume analytics technology need to support experimentation and evaluation. Amazon Redshift was designed to bring data warehousing to a mass market by making it easy to buy, easy to tune and easy to manage while also being fast and cost-effective.

1. Introduction

Many companies augment their transaction-processing database systems with data warehouses for reporting and analysis. Analysts estimate the data warehouse market segment at 1/3 of the overall relational database market segment ($14B vs. $40B for software licenses and support), with an 8-11% compound annual growth rate (CAGR)[1]. While this is a strong growth rate for a large, mature market, over the past ten years, analysts also estimate data storage at a typical enterprise growing at 30-40%. CAGR. Over the past 12-18 months, new market research has begun to show an increase to 50-60%, with data doubling in size every 20 months.

Figure 1: Data Analysis Gap in the Enterprise [10]

This implies most data in an enterprise is "dark data"[2]: data that is collected but not easily analyzed. We see this as unfortunate. If our customers didn't see this data as having value, they would not retain it. Many companies are trying to become increasingly data-driven. And yet, not only is most data already dark, the overall data landscape is only getting darker. Storing this data in NoSQL stores and/or Hadoop is one way to bridge the gap for certain use cases. However it doesn't address all scenarios.

In our discussions with customers, we found the "analysis gap" between data being collected and data available for analysis as due to four major causes:

1. Cost – Most commercial database solutions capable of analyzing data at scale require significant up-front expense. This is hard to justify for large datasets with unclear value.
2. Complexity – Database provisioning, maintenance, backup, and tuning are complex tasks requiring specialized skills. They require IT involvement and cannot easily be performed by line of business data scientists or analysts.
3. Performance – It is difficult to grow a data warehouse without negatively impacting query performance. Once built, IT teams sometimes discourage augmenting data or adding queries as a way of protecting current reporting SLAs.
4. Rigidity – Most databases work best on highly structured relational data. But a large and increasing percentage of data consists of machine-generated logs that mutate over time, audio and video, not readily accessible to relational analysis.

We see each of the above issues only increasing with data set size. To take one large-scale customer example, the Amazon Retail team collects about 5 billion web log records daily (2TB/day,

[1] Forecast data combined from IDC, Gartner, and 451 Research.
[2] http://www.gartner.com/it-glossary/dark-data

First page figure ✅

Figures Should be Stand-Alone

- It should be possible to roughly understand a figure without the text

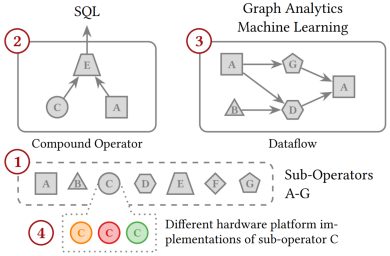


Figure 1: Sub-operators ① build more complex data operations ② or dataflows ③, where each sub-operator can be implemented on multiple hardware platforms ④.

Stand-alone figure ✅

Figures Should be Stand-Alone

- It should be possible to roughly understand a figure without the text
- Use captions to explain figures

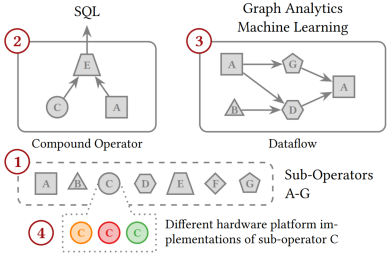


Figure 1: Sub-operators ① build more complex data operations ② or dataflows ③, where each sub-operator can be implemented on multiple hardware platforms ④.

Stand-alone figure ✅

Figures Should be Stand-Alone

- It should be possible to roughly understand a figure without the text
- Use captions to explain figures
- Captions may be several lines long

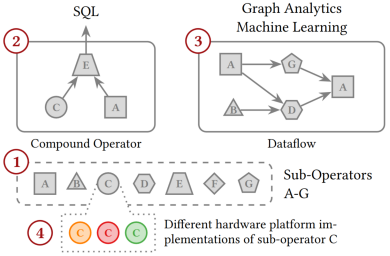


Figure 1: Sub-operators ① build more complex data operations ② or dataflows ③, where each sub-operator can be implemented on multiple hardware platforms ④.

Stand-alone figure ✅

Figures Should be Stand-Alone

- It should be possible to roughly understand a figure without the text
- Use captions to explain figures
- Captions may be several lines long
- Use text within figures to explain

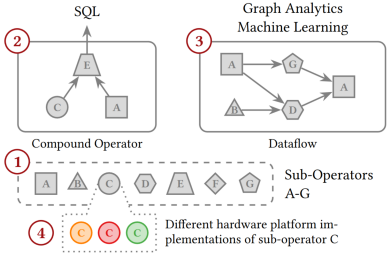


Figure 1: Sub-operators ① build more complex data operations ② or dataflows ③, where each sub-operator can be implemented on multiple hardware platforms ④.

Stand-alone figure ✅

Figures Should be Stand-Alone

- It should be possible to roughly understand a figure without the text
- Use captions to explain figures
- Captions may be several lines long
- Use text within figures to explain
- But: every figure needs to have a purpose for the text and must be referenced!

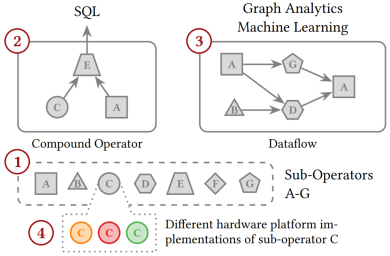


Figure 1: Sub-operators ① build more complex data operations ② or dataflows ③, where each sub-operator can be implemented on multiple hardware platforms ④.

Stand-alone figure ✅

Figures Should be Stand-Alone

- It should be possible to roughly understand a figure without the text
- Use captions to explain figures
- Captions may be several lines long
- Use text within figures to explain
- But: every figure needs to have a purpose for the text and must be referenced!
- For complex graphics, also use the `circledsteps` package to ① reference figure parts

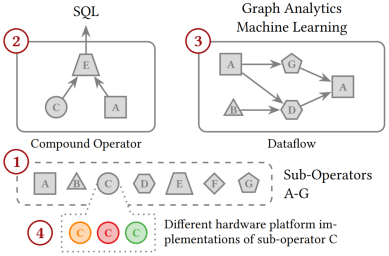


Figure 1: Sub-operators ① build more complex data operations ② or dataflows ③, where each sub-operator can be implemented on multiple hardware platforms ④.

Stand-alone figure ✅

Figures Should be Simple

- Make figures as simple as possible

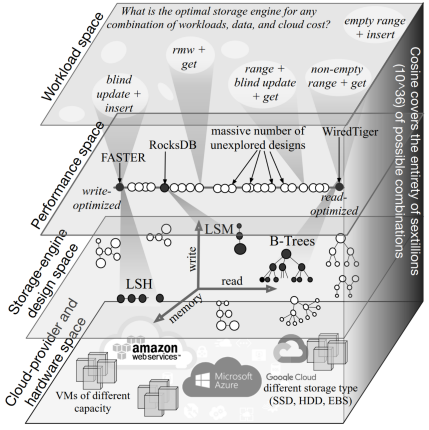


Figure 1: Fixed-design systems capture only a small fraction of the possible storage-engine design space on the cloud.

Complex figure ❌

Figures Should be Simple

- Make figures as simple as possible
- Better use two figures than one complex figure

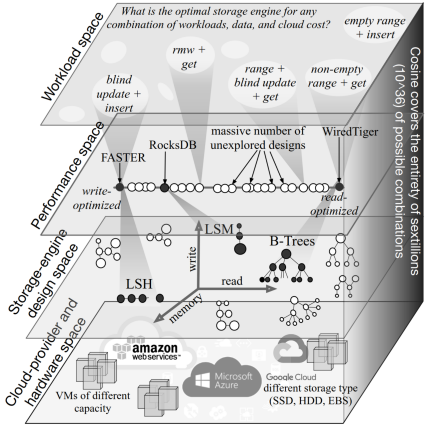
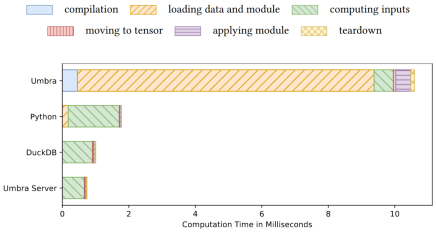


Figure 1: Fixed-design systems capture only a small fraction of the possible storage-engine design space on the cloud.

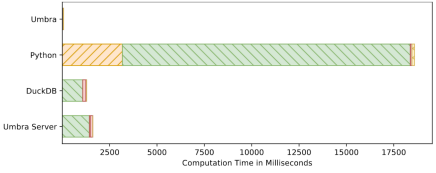
Complex figure ❌

Rely on Color?

- Good coloring can make a graph more understandable



(a) Linear regression execution for 100 tuples



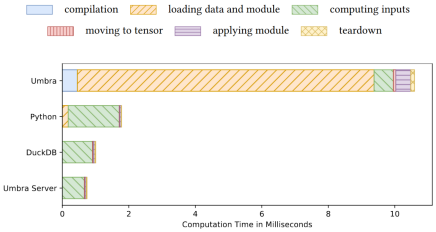
(b) Linear regression execution for 1 million tuples

Figure 10: Runtime decomposition of linear regression.

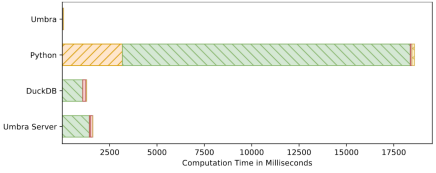
Colored figure that works on black and white ✅

Rely on Color?

- Good coloring can make a graph more understandable
- Should be accessible to colorblind and people with black and white printers



(a) Linear regression execution for 100 tuples



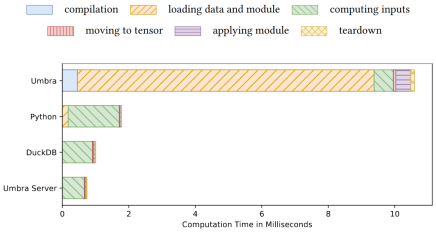
(b) Linear regression execution for 1 million tuples

Figure 10: Runtime decomposition of linear regression.

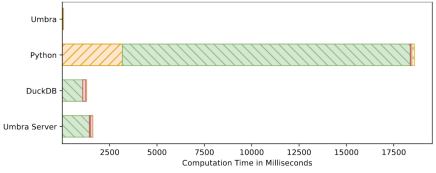
Colored figure that works on black and white ✅

Rely on Color?

- Good coloring can make a graph more understandable
- Should be accessible to colorblind and people with black and white printers
  - Important to many, ignored by many



(a) Linear regression execution for 100 tuples



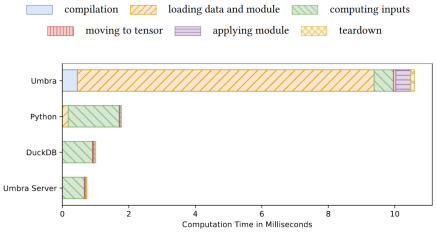
(b) Linear regression execution for 1 million tuples

Figure 10: Runtime decomposition of linear regression.

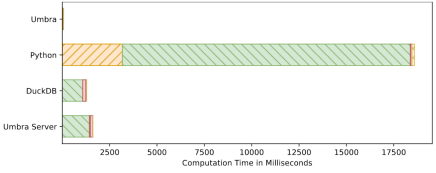
Colored figure that works on black and white ✅

Rely on Color?

- Good coloring can make a graph more understandable
- Should be accessible to colorblind and people with black and white printers
  - Important to many, ignored by many
- Our advice:



(a) Linear regression execution for 100 tuples



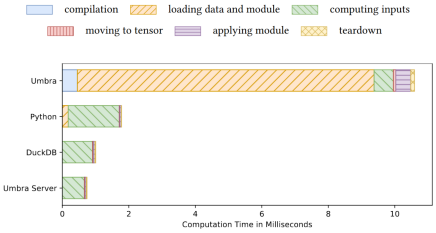
(b) Linear regression execution for 1 million tuples

Figure 10: Runtime decomposition of linear regression.

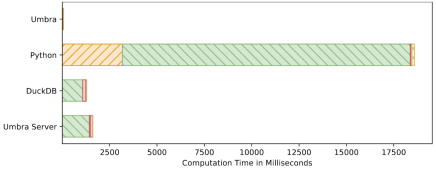
Colored figure that works on black and white ✅

Rely on Color?

- Good coloring can make a graph more understandable
- Should be accessible to colorblind and people with black and white printers
  - Important to many, ignored by many
- Our advice:
  - Use color to make figures as expressive as possible



(a) Linear regression execution for 100 tuples



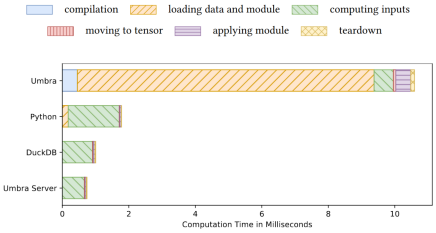
(b) Linear regression execution for 1 million tuples

Figure 10: Runtime decomposition of linear regression.

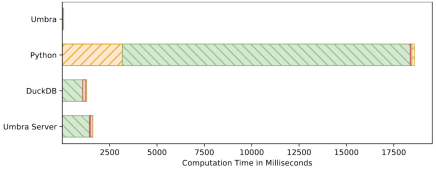
Colored figure that works on black and white ✅

Rely on Color?

- Good coloring can make a graph more understandable
- Should be accessible to colorblind and people with black and white printers
  - Important to many, ignored by many
- Our advice:
  - Use color to make figures as expressive as possible
  - Use shades ($<= 4$), shapes, or text to present the same information without color



(a) Linear regression execution for 100 tuples



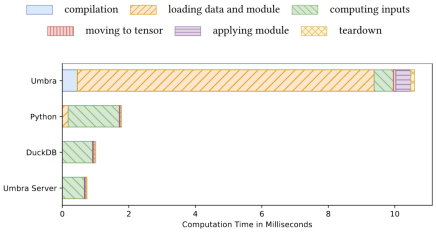
(b) Linear regression execution for 1 million tuples

Figure 10: Runtime decomposition of linear regression.

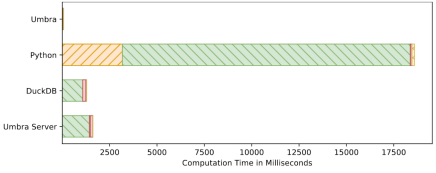
Colored figure that works on black and white ✅

Rely on Color?

- Good coloring can make a graph more understandable
- Should be accessible to colorblind and people with black and white printers
  - Important to many, ignored by many
- Our advice:
  - Use color to make figures as expressive as possible
  - Use shades ($<= 4$), shapes, or text to present the same information without color
  - Ignore this rule if it is impossible



(a) Linear regression execution for 100 tuples



(b) Linear regression execution for 1 million tuples

Figure 10: Runtime decomposition of linear regression.

Colored figure that works on black and white ✅

Revise Figures

- Just as for text, you should spend most of your time revising a figure

Revise Figures

- Just as for text, you should spend most of your time revising a figure
- Start with sketches of figures or quick plots

Revise Figures

- Just as for text, you should spend most of your time revising a figure
- Start with sketches of figures or quick plots
- If they stay in the paper, you polish them later

Graphs and Tables

- Usually, graphs are better than tables

Official ScanNet Benchmark

Method	avg IoU	Chair	Floor	Other Furniture	Picture	Sofa	Table	Wall	...
PointNet++	0.339	0.360	0.677	0.183	0.117	0.346	0.232	0.523	...
PointNet++ [1]	0.481	0.686	0.931	0.299	0.102	0.580	0.470	0.711	...
Additional Features (ours)	**0.557**	**0.744**	**0.946**	**0.376**	**0.205**	**0.643**	**0.497**	**0.756**	...

Table with visual hints ✅

Graphs and Tables

- Usually, graphs are better than tables
- Tables allow to perform lots of comparisons

Official ScanNet Benchmark

Method	avg IoU	Chair	Floor	Other Furniture	Picture	Sofa	Table	Wall	...
PointNet++	0.339	0.360	0.677	0.183	0.117	0.346	0.232	0.523	...
PointNet++ [1]	0.481	0.686	0.931	0.299	0.102	0.580	0.470	0.711	...
Additional Features (ours)	**0.557**	**0.744**	**0.946**	**0.376**	**0.205**	**0.643**	**0.497**	**0.756**	...

Table with visual hints ✅

Graphs and Tables

- Usually, graphs are better than tables
- Tables allow to perform lots of comparisons
  - You can provide a lot of detail in tables

Official ScanNet Benchmark

Method	avg IoU	Chair	Floor	Other Furniture	Picture	Sofa	Table	Wall	...
PointNet++	0.339	0.360	0.677	0.183	0.117	0.346	0.232	0.523	...
PointNet++ [1]	0.481	0.686	0.931	0.299	0.102	0.580	0.470	0.711	...
Additional Features (ours)	**0.557**	**0.744**	**0.946**	**0.376**	**0.205**	**0.643**	**0.497**	**0.756**	...

Table with visual hints ✅

Graphs and Tables

- Usually, graphs are better than tables
- Tables allow to perform lots of comparisons
  - You can provide a lot of detail in tables
- Use visual hints in tables:

Official ScanNet Benchmark

Method	avg IoU	Chair	Floor	Other Furniture	Picture	Sofa	Table	Wall	...
PointNet++	0.339	0.360	0.677	0.183	0.117	0.346	0.232	0.523	...
PointNet++ [1]	0.481	0.686	0.931	0.299	0.102	0.580	0.470	0.711	...
Additional Features (ours)	**0.557**	**0.744**	**0.946**	**0.376**	**0.205**	**0.643**	**0.497**	**0.756**	...

Table with visual hints ✅

Graphs and Tables

- Usually, graphs are better than tables
- Tables allow to perform lots of comparisons
  - You can provide a lot of detail in tables
- Use visual hints in tables:
  - Bold for **best**

Official ScanNet Benchmark

Method	avg IoU	Chair	Floor	Other Furniture	Picture	Sofa	Table	Wall	...
PointNet++	0.339	0.360	0.677	0.183	0.117	0.346	0.232	0.523	...
PointNet++ [1]	0.481	0.686	0.931	0.299	0.102	0.580	0.470	0.711	...
Additional Features (ours)	**0.557**	**0.744**	**0.946**	**0.376**	**0.205**	**0.643**	**0.497**	**0.756**	...

Table with visual hints ✅

Graphs and Tables

- Usually, graphs are better than tables
- Tables allow to perform lots of comparisons
  - You can provide a lot of detail in tables
- Use visual hints in tables:
  - Bold for **best**
  - Color for good/neutral/bad

Official ScanNet Benchmark

Method	avg IoU	Chair	Floor	Other Furniture	Picture	Sofa	Table	Wall	...
PointNet++	0.339	0.360	0.677	0.183	0.117	0.346	0.232	0.523	...
PointNet++ [1]	0.481	0.686	0.931	0.299	0.102	0.580	0.470	0.711	...
Additional Features (ours)	**0.557**	**0.744**	**0.946**	**0.376**	**0.205**	**0.643**	**0.497**	**0.756**	...

Table with visual hints ✅

Pseudocode

- Pseudocode can be very helpful

Pseudocode

- Pseudocode can be very helpful
- Describes algorithms very accurately

Pseudocode

- Pseudocode can be very helpful
- Describes algorithms very accurately
  - Be precise on details, there is no room for interpretation

Pseudocode

- Pseudocode can be very helpful
- Describes algorithms very accurately
  - Be precise on details, there is no room for interpretation
- Is difficult to understand

Pseudocode

- Pseudocode can be very helpful
- Describes algorithms very accurately
  - Be precise on details, there is no room for interpretation
- Is difficult to understand
  - Reduce cognitive load as much as possible

Pseudocode

- Pseudocode can be very helpful
- Describes algorithms very accurately
  - Be precise on details, there is no room for interpretation
- Is difficult to understand
  - Reduce cognitive load as much as possible
  - De-clutter syntax (Python-like syntax can be good)

Pseudocode

- Pseudocode can be very helpful
- Describes algorithms very accurately
  - Be precise on details, there is no room for interpretation
- Is difficult to understand
  - Reduce cognitive load as much as possible
  - De-clutter syntax (Python-like syntax can be good)
  - Use syntax highlighting

Pseudocode

- Pseudocode can be very helpful
- Describes algorithms very accurately
  - Be precise on details, there is no room for interpretation
- Is difficult to understand
  - Reduce cognitive load as much as possible
  - De-clutter syntax (Python-like syntax can be good)
  - Use syntax highlighting
  - Name variables well

Pseudocode

- Pseudocode can be very helpful
- Describes algorithms very accurately
  - Be precise on details, there is no room for interpretation
- Is difficult to understand
  - Reduce cognitive load as much as possible
  - De-clutter syntax (Python-like syntax can be good)
  - Use syntax highlighting
  - Name variables well
  - Name algorithms / functions / methods

Pseudocode

- Pseudocode can be very helpful
- Describes algorithms very accurately
  - Be precise on details, there is no room for interpretation
- Is difficult to understand
  - Reduce cognitive load as much as possible
  - De-clutter syntax (Python-like syntax can be good)
  - Use syntax highlighting
  - Name variables well
  - Name algorithms / functions / methods
  - Add descriptive captions

Pseudocode

- Pseudocode can be very helpful
- Describes algorithms very accurately
  - Be precise on details, there is no room for interpretation
- Is difficult to understand
  - Reduce cognitive load as much as possible
  - De-clutter syntax (Python-like syntax can be good)
  - Use syntax highlighting
  - Name variables well
  - Name algorithms / functions / methods
  - Add descriptive captions
  - Specify input and output

Large Language Models (LLMs)

- Use LLMs for

Large Language Models (LLMs)

- Use LLMs for
  - checking spelling and grammar, get feedback

Large Language Models (LLMs)

- Use LLMs for
  - checking spelling and grammar, get feedback
  - getting inspiration for how to write a certain sentence or section

Large Language Models (LLMs)

- Use LLMs for
  - checking spelling and grammar, get feedback
  - getting inspiration for how to write a certain sentence or section
- Do not use LLMs for

Large Language Models (LLMs)

- Use LLMs for
  - checking spelling and grammar, get feedback
  - getting inspiration for how to write a certain sentence or section
- Do not use LLMs for
  - structure your paper or work on your very first draft (e.g., bulletpoints) of a section

Large Language Models (LLMs)

- Use LLMs for
  - checking spelling and grammar, get feedback
  - getting inspiration for how to write a certain sentence or section
- Do not use LLMs for
  - structure your paper or work on your very first draft (e.g., bulletpoints) of a section
  - writing your (entire paper), the writing style tends to be not on point, also external requirements

Large Language Models (LLMs)

- Use LLMs for
  - checking spelling and grammar, get feedback
  - getting inspiration for how to write a certain sentence or section
- Do not use LLMs for
  - structure your paper or work on your very first draft (e.g., bulletpoints) of a section
  - writing your (entire paper), the writing style tends to be not on point, also external requirements
- If you don't want to give some company all your data, you can also self-host small models
  I like `ollama` with `lama3-8b`

Large Language Models (LLMs)

- Use LLMs for
  - checking spelling and grammar, get feedback
  - getting inspiration for how to write a certain sentence or section
- Do not use LLMs for
  - structure your paper or work on your very first draft (e.g., bulletpoints) of a section
  - writing your (entire paper), the writing style tends to be not on point, also external requirements
- If you don't want to give some company all your data, you can also self-host small models
  I like `ollama` with `lama3-8b`
- If we notice that your paper includes any text you did not understand or find long sections that just paraphrase sources you will not pass this course

LaTeX

- LaTeX composes visually nice results (visual appeal is important)

LaTeX

- LaTeX composes visually nice results (visual appeal is important)
- Often, you have to use templates anyway

LaTeX

- LaTeX composes visually nice results (visual appeal is important)
- Often, you have to use templates anyway
- Write one sentence per line

LaTeX

- LaTeX composes visually nice results (visual appeal is important)
- Often, you have to use templates anyway
- Write one sentence per line
  - Check length of sentences

LaTeX

- LaTeX composes visually nice results (visual appeal is important)
- Often, you have to use templates anyway
- Write one sentence per line
  - Check length of sentences
  - Easier reasoning about sentences (each sentence should have a purpose)

LaTeX

- LaTeX composes visually nice results (visual appeal is important)
- Often, you have to use templates anyway
- Write one sentence per line
  - Check length of sentences
  - Easier reasoning about sentences (each sentence should have a purpose)
  - Works well with version control

Git

- **Always use git** even if you are working alone

Git

- **Always use git** even if you are working alone
- Helps with collaboration

Git

- **Always use git** even if you are working alone
- Helps with collaboration
- Synchronizes across devices

Git

- **Always use git** even if you are working alone
- Helps with collaboration
- Synchronizes across devices
- Shows diffs

Git

- **Always use git** even if you are working alone
- Helps with collaboration
- Synchronizes across devices
- Shows diffs
- Makes it harder to loose data

Git

- **Always use git** even if you are working alone
- Helps with collaboration
- Synchronizes across devices
- Shows diffs
- Makes it harder to loose data
- Commit often, push often

Git

- **Always use git** even if you are working alone
- Helps with collaboration
- Synchronizes across devices
- Shows diffs
- Makes it harder to loose data
- Commit often, push often
- Check in all stuff necessary for building (text and figures) but not outputs

LaTeX is a Time Sink

- For quick iteration try to avoid LaTeX for initial draft(s)

LaTeX is a Time Sink

- For quick iteration try to avoid LaTeX for initial draft(s)
  - Text: Simple editor without much distraction (e.g. Word, Google Docs, Plain Editor)

LaTeX is a Time Sink

- For quick iteration try to avoid LaTeX for initial draft(s)
  - Text: Simple editor without much distraction (e.g. Word, Google Docs, Plain Editor)
  - Tables: Any software that can edit tables (Word, Excel, or alternatives)

LaTeX is a Time Sink

- For quick iteration try to avoid LaTeX for initial draft(s)
  - Text: Simple editor without much distraction (e.g. Word, Google Docs, Plain Editor)
  - Tables: Any software that can edit tables (Word, Excel, or alternatives)
  - Images: avoid perfectionism (excalidraw.com, Paper)

LaTeX is a Time Sink

- For quick iteration try to avoid LaTeX for initial draft(s)
  - Text: Simple editor without much distraction (e.g. Word, Google Docs, Plain Editor)
  - Tables: Any software that can edit tables (Word, Excel, or alternatives)
  - Images: avoid perfectionism (excalidraw.com, Paper)
- use `\usepackage{booktabs}` for tables

Citations with BibTeX

- Citing with BibTeX is easy

sample.bib

```
@Article{Abril07,
  author          = "Patricia S. Abril and...",
  title           = "The patent holder's...",
  journal         = "Communications of the ACM",
  volume          = "50",
  number          = "1",
  month           = jan,
  year            = "2007",
  pages           = "36--44",
  doi             = "10.1145/1188913.1188915",
  url             = "http://doi.acm.org/...",
}
```

main.tex

```
\section{Citations}
Some references~\cite{Abril07,Xy} ...
```

Citations with BibTeX

- Citing with BibTeX is easy
- Get correct `.bib` files from dblp.org

sample.bib

```
@Article{Abril07,
  author          = "Patricia S. Abril and...",
  title           = "The patent holder's...",
  journal         = "Communications of the ACM",
  volume          = "50",
  number          = "1",
  month           = jan,
  year            = "2007",
  pages           = "36--44",
  doi             = "10.1145/1188913.1188915",
  url             = "http://doi.acm.org/...",
}
```

main.tex

```
\section{Citations}
Some references~\cite{Abril07,Xy} ...
```

Citations with BibTeX

- Citing with BibTeX is easy
- Get correct `.bib` files from dblp.org
  - Be sure that you cite the correct version of the paper

sample.bib

```
@Article{Abril07,
  author          = "Patricia S. Abril and...",
  title           = "The patent holder's...",
  journal         = "Communications of the ACM",
  volume          = "50",
  number          = "1",
  month           = jan,
  year            = "2007",
  pages           = "36--44",
  doi             = "10.1145/1188913.1188915",
  url             = "http://doi.acm.org/...",
}
```

main.tex

```
\section{Citations}
Some references~\cite{Abril07,Xy} ...
```

Citations with BibTeX

- Citing with BibTeX is easy
- Get correct `.bib` files from dblp.org
  - Be sure that you cite the correct version of the paper
  - Arxiv is one of the worst sources, try to find a source from a journal or conference

sample.bib

```
@Article{Abril07,
  author          = "Patricia S. Abril and...",
  title           = "The patent holder's...",
  journal         = "Communications of the ACM",
  volume          = "50",
  number          = "1",
  month           = jan,
  year            = "2007",
  pages           = "36--44",
  doi             = "10.1145/1188913.1188915",
  url             = "http://doi.acm.org/...",
}
```

main.tex

```
\section{Citations}
Some references~\cite{Abril07,Xy} ...
```

Citations with BibTeX

- Citing with BibTeX is easy
- Get correct `.bib` files from dblp.org
  - Be sure that you cite the correct version of the paper
  - Arxiv is one of the worst sources, try to find a source from a journal or conference
  - Especially on google scholar you will often find bad BibTeX files

sample.bib

```
@Article{Abril07,
  author          = "Patricia S. Abril and...",
  title           = "The patent holder's...",
  journal         = "Communications of the ACM",
  volume          = "50",
  number          = "1",
  month           = jan,
  year            = "2007",
  pages           = "36--44",
  doi             = "10.1145/1188913.1188915",
  url             = "http://doi.acm.org/...",
}
```

main.tex

```
\section{Citations}
Some references~\cite{Abril07,Xy} ...
```

Citations with BibTeX

- Citing with BibTeX is easy
- Get correct `.bib` files from dblp.org
  - Be sure that you cite the correct version of the paper
  - Arxiv is one of the worst sources, try to find a source from a journal or conference
  - Especially on google scholar you will often find bad BibTeX files
- Before submission: make sure any warnings are gone and all citations render!

sample.bib

```
@Article{Abril07,
  author          = "Patricia S. Abril and...",
  title           = "The patent holder's...",
  journal         = "Communications of the ACM",
  volume          = "50",
  number          = "1",
  month           = jan,
  year            = "2007",
  pages           = "36--44",
  doi             = "10.1145/1188913.1188915",
  url             = "http://doi.acm.org/...",
}
```

main.tex

```
\section{Citations}
Some references~\cite{Abril07,Xy} ...
```

Final Touches

- New terms should be written once in *italics* and then explained

Final Touches

- New terms should be written once in *italics* and then explained
- Make sure each figure is referenced in text

Final Touches

- New terms should be written once in *italics* and then explained
- Make sure each figure is referenced in text
- Position figures when you are finished with the rest

Final Touches

- New terms should be written once in *italics* and then explained
- Make sure each figure is referenced in text
- Position figures when you are finished with the rest
- Optimize line breaks

Final Touches

- New terms should be written once in *italics* and then explained
- Make sure each figure is referenced in text
- Position figures when you are finished with the rest
- Optimize line breaks
- Check references

Final Touches

- New terms should be written once in *italics* and then explained
- Make sure each figure is referenced in text
- Position figures when you are finished with the rest
- Optimize line breaks
- Check references
- Spell check

Final Touches

- New terms should be written once in *italics* and then explained
- Make sure each figure is referenced in text
- Position figures when you are finished with the rest
- Optimize line breaks
- Check references
- Spell check
- **Use your favorite LLVM for spell and grammar checks**

Further References

- Deirdre Nansen McCloskey, **Economical Writing**,Third Edition, 2019
- Joseph Williams, **Style: Toward Clarity and Grace**, Univ. of Chicago Press, 1990
- Justin Zobel, **Writing for Computer Science**, Springer, Third Edition, 2014
- Larry McEnerney, **The Craft of Writing Effectively** [link]
- Lorenz Froihofer, **Tips for scientific writing (for Germans)** [link]
- Lorenz Froihofer, **How to write a computer science paper** [link]

Summary

- Good writing is hard but important

Summary

- Good writing is hard but important
- Each sentence needs to contribute to your goal

Summary

- Good writing is hard but important
- Each sentence needs to contribute to your goal
- Revise more than you write

Summary

- Good writing is hard but important
- Each sentence needs to contribute to your goal
- Revise more than you write
- Write clear and concise

Summary

- Good writing is hard but important
- Each sentence needs to contribute to your goal
- Revise more than you write
- Write clear and concise
- A perfect text is one where you cannot omit anything