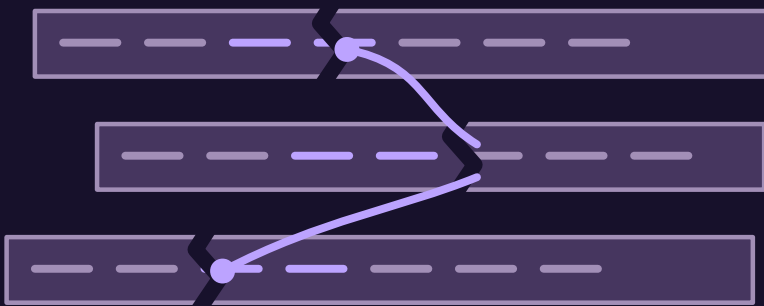




PRACTICAL AI

Debug Chunking and Context Selection

Find exactly where a rule or exception disappears.



Sanjaay Gowda T S

FREE SAMPLE / INTRODUCTION + CHAPTER 1 / EDITION 1.0.0

sanjaaygowdats.ai

About this free sample

Debug Chunking and Context Selection

Sanjaay Gowda T S

First edition 1.0.0, 2026-10-13

Copyright 2026 Sanjaay Gowda T S. This is a labelled free sample, not the complete ebook. You may share this sample unchanged. It contains the cover, sample permissions, contents, introduction, Chapter 1 and a closing product link. The remaining chapters, worked solutions and companion files are not included. The full ebook text, diagrams and artwork have separate single-reader terms; this sample permission does not authorize sharing the full purchased PDF or EPUB.

The companion source code, tests and fictional data are separately MIT licensed. You may adapt or share that project with its included licence notice. MIT does not apply to the ebook text or book artwork. The books and companion projects have no technical activation requirement or buyer-specific watermark.

This first edition was prepared with AI assistance. The case studies are fictional teaching examples. Any measured outputs come from the supplied examples, not production customer data. The teaching scope, assumptions and limitations are explained in the chapters. Independent human technical, editorial and assistive-technology reader review is pending. Automated tests and accessibility checks do not establish certification or suitability for every use.

Run examples in a separate working folder using the supplied synthetic data. Review inputs and calculations before adapting the techniques to real operations. Model software, service access, subscriptions, ongoing updates and consulting are not included.

Author website: sanjaaygowdats.ai

Ebook terms: sanjaaygowdats.ai/ebooks/license

Inside this free sample

Start here.....	4
1. Define the missing-context problem	5
Continue with the full guide	7

This free sample contains the introduction and Chapter 1 only. The closing link opens the complete guide. The companion project and worked solutions are supplied with the complete ebook.

Start here

The answer's missing exception may have disappeared long before generation. It could have been separated during chunking, ranked below the candidate cutoff, skipped by a context budget or silently truncated from a long passage. Changing the prompt cannot restore evidence that never reached it. This book builds a small laboratory for locating the earliest stage where useful context was lost.

The fictional example concerns battery-pack rentals. Its ordinary rule requires return before the rental period ends. An unexpected desk closure creates an exception. A separate paragraph explains damaged-pack handling. These invented policies are teaching data, not safety instructions or a real rental contract. The companion performs no model call and produces no customer answer. It traces words through chunks, lexical ranking and whole-chunk selection.

Use Python 3.10 or later and basic lists, dictionaries and functions. Open a terminal in the extracted project directory, where `main.py` and `example-input.json` are siblings, and run:

SHELL

```
python --version
python main.py --input example-input.json
python -m unittest discover -s tests -v
```

Windows users may use `py -3`; macOS or Linux users may need `python3`. No packages, API keys or services are required. `Expected-output.json` records the actual deterministic example output. The tests verify boundaries and coverage behavior, including deliberate partial-context failures. They do not demonstrate any language model's answer quality.

Your deliverable is an evidence-preserving design record: source paragraph identities, chunk spans, ranked candidates, selected context, skipped passages and annotated coverage. Word budgets are illustrative and must not be treated as model token limits. Full-paragraph coverage is a conservative local proxy for evidence availability, not a semantic judgment that an answer is correct.

1. Define the missing-context problem

Start with the question's conditions

The example asks whether a battery pack can be returned after the rental ends. A response that quotes only the ordinary deadline could miss the closure exception. A response that quotes only the exception could imply that all late returns are allowed. The evidence need therefore includes a relationship between two paragraphs rather than a single matching sentence.

Write the complete evidence requirement before adjusting chunk size. In the companion, ordinary-return requires return-rule. Closure-exception requires both return-rule and return-exception. Damage-handling requires damage-rule. These annotations deliberately make the required bundles inspectable. They are authored teaching judgments, not automatically discovered facts.

The project reports all three annotations for the one question. Some, such as damage handling, may be secondary to the question. This makes coverage inspection rich enough to show what was omitted; it does not declare every annotation equally necessary for every answer. A real evaluation should associate required bundles with each particular question and distinguish essential from optional evidence.

Name the stages separately

Source text is divided into chunks. A lexical ranker gives positive-score chunks an order. Top-k limits the candidate list. A budget packer chooses whole chunks from those candidates. Finally, a coverage check compares selected word positions with annotated paragraphs. Every stage can lose material for a different reason and needs its own diagnostic fields.

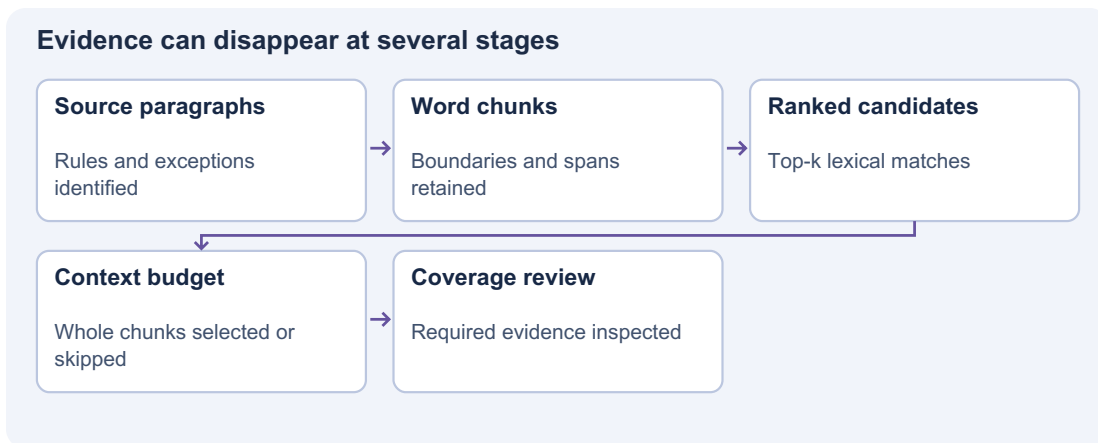


Figure 1. Source text passes through chunking, ranking and context packing before a reviewer sees it.

Figure 1 is a debugging path, not a promise that more context always improves results. Extra passages can introduce contradictions or distract a generator. The goal is enough authorized evidence for the question, preserved with its conditions. The companion helps locate omissions; semantic review determines whether the selected material actually supports the intended answer.

Avoid beginning with a magic chunk size

There is no universal word count that preserves every rule and exception. Short chunks can separate related clauses. Long chunks can dilute lexical overlap and consume the entire context budget. A boundary that suits a short policy may be wrong for a table, procedure or conversation. Choose an inspectable baseline and compare failures before adopting a more complicated strategy.

A useful debugging question is "Where did this required paragraph stop being available?" That question leads to source, chunk, rank and selection records. "Which chunk size is best?" is too broad without a task, corpus and metric. Keep the narrower question until the evidence points to a concrete repair.

Exercise: annotate a two-paragraph need

Read the ordinary return rule and closure exception. Explain why both belong in the closure-exception bundle. Write one question requiring only the ordinary rule and one requiring the exception. State how an answer based on a single paragraph could mislead in each direction.

Continue with the full guide

This free sample ends after the introduction and Chapter 1. The complete guide contains nine teaching chapters, worked exercise solutions, primary references and a separate Python companion project. The full PDF, reflowable EPUB and project files are supplied together.

Use the complete guide to work through the remaining exercises and inspect the supplied examples. The companion code and fictional data are separately MIT licensed; the complete ebook has the single-reader terms described on its product page.

The examples are teaching fixtures, not production customer results. Independent human technical, editorial and assistive-technology reader review remains pending. Software access, subscriptions, consulting and guaranteed results are not included.

[View the complete guide and purchase options.](#)

You may share this labelled free sample unchanged. Keep the full purchased PDF and EPUB private. Thank you for reading.