Binoculars GUI User Guide

This manual details the operation of the Binoculars graphical interface for iterative editing and rescoring. The workflow is designed for rapid rewrite cycles using approximate impact feedback, with periodic full `Analyze` runs to obtain precise metrics, and regular use of `Clear Priors` to maintain a clear display.

1) Starting the GUI

From the repository root, execute:

```
./binoculars.sh --config fast --gui path/to/doc.md
```

Alternatively:

```
venv/bin/python binoculars.py --config fast --gui path/to/doc.md
```

2) Interface Layout

The main interface consists of several sections:

- Left pane: editable source text.
- Left gutter: line numbers and per-line contribution indicators.
- Right pane: markdown preview, with backgrounds reflecting score data.
- Right-pane footer: synonym panel, activated by clicking on words.
- Toolbar: `Analyze`, `Save`, `Undo`, `Clear Priors`, `Quit`.
- Status bar: displays current score, metrics, and workflow state.

Colour conventions after running Analyze:

- Red foreground lines or segments: lower-perplexity (`LOW`) contributors.
- Green foreground lines or segments: higher-perplexity (`HIGH`) contributors.
- Yellow foreground: your edits since the last Analyze.
- Faint background colours: previous analyzes or edits, retained for comparison.

3) Core Workflow Structure

The GUI separates editing into two phases.

1. Approximate iteration phase:

    - Edit text.
    - Request rewrites.

- Review approximate B impact in the rewrite popup.
- Apply options quickly.

2. Exact scoring phase:

- Run `Analyze` at intervals.
- Refresh exact `B`, observer/cross metrics, and heatmap labels.

Key points:

- Impact values in the rewrite popup are estimates.
- Exact B requires a full `Analyze` run.
- After edits or rewrites, the status bar marks B as stale until Analyze is run.

4) Recommended Iteration Loop

For efficient throughput, use this sequence:

1. Press `Analyze` on the current document.
2. Inspect red and green contributors.
3. Edit directly or request rewrites.
4. Apply one or more rewrite options (auto-analyze does not trigger).
5. Continue editing while monitoring the stale indicator.
6. Re-run `Analyze` when you need to confirm the exact score.
7. Use `Undo` to revert the last tracked change.
8. Press `Clear Priors` if overlays become visually distracting.
9. Repeat as needed.

Recommended cadence:

- Run `Analyze` every few rewrites (for example, every 3–8), or after a major section change.
- Run `Clear Priors` if faint backgrounds begin to obscure current hot spots.

5) Requesting Rewrites

A) Rewrite a Single Scored Red Segment

1. Ensure at least one successful `Analyze` run has completed.
2. Right-click directly on a red (`LOW`) segment.
3. Wait for the popup to generate and score options.
4. Review options `[1]`, `[2]`, `[3]`, each with an approximate B impact.
5. Select an option using the button or keyboard: `1`, `2`, or `3`.
6. The selected text replaces the segment in the left pane.

B) Rewrite a Highlighted Block (Multi-Line)

1. Highlight any block in the left pane (may include mixed LOW/HIGH/neutral lines).

2. Right-click while the selection is active.

3. The GUI rounds the selection to full lines for rewrite generation.

4. If the selection includes unscored text, only the scored portion is rewritten.

5. The popup displays three options and approximate B deltas.

6. Select an option using the button or 1/2/3.

Selection rewrite behaviour:

- Unchanged source lines are preserved, even if the model output omits or collapses them.
- Trailing blank lines from the selected source are retained in the replacement.

6) Rewrite Popup Details

Popup features:

- Scrollable option body for long rewrites.
- Status and progress lines during generation.
- Source label indicates the rewrite backend:
    - external LLM
    - internal fallback
    - internal model
- Quit or Esc cancels with no change.

Option order:

- Sorted by expected increase in B (most human-like first).

7) Synonym Finder and One-Level Undo

The synonym finder operates as follows. Left-click a word in the left pane. After a brief delay, the synonym panel appears, displaying up to nine options arranged in three columns. To select a synonym, click the button 1..9 in the panel. The replacement is inserted into the left pane and highlighted in yellow to indicate an edit.

Synonym sources are prioritized in this order: the local fallback list, WordNet (if available in the environment), and the Datamuse API as a fallback.

One-level Undo is supported. The command Undo reverts exactly one tracked operation. Supported tracked operations include selected-block delete (Delete or Backspace), synonym replacement, single red-segment rewrite replacement, and highlighted-block rewrite replacement. Undo is intentionally limited to one level. If the text changes after the tracked operation, Undo is invalidated for safety.

8) Exact vs Approximate Metrics

Approximate values in the popup are designed for quickly ranking options. They do not recompute the full document cross term.

Exact values in the status bar are updated only by `Analyze`. These include exact `B`, `Observer logPPL`, `Performer logPPL`, `Cross logXPPL`, and scored `Last Line`.

9) Prior Overlays and `Clear Priors`

Priors are faint background traces of previous analyzes and edits. They provide visual context for "before vs after".

Clearing priors is recommended after several iteration cycles when old context becomes distracting, or before a final review pass if only the latest heatmap should be visible.

The `Clear Priors` command removes faint prior backgrounds. It does not alter the current text and does not run Analyze.

10) Preview Pane Selection Mirroring

When a block is highlighted on the left, the right preview mirrors the same selected block. Within the selected block, LOW lines display a red-tinted background, HIGH lines use a green-tinted background, and neutral lines use a neutral highlight style.

This approach makes it easier to inspect local context before requesting rewrites.

11) Saving and Output Files

`Save` creates timestamped files alongside the original source file.

- `[[HTML_BLOCK_0]]_edited_YYYYMMDDHHMM.md`
- If a filename conflict occurs, a numeric suffix is added.

Tip: Save snapshots at key points - such as after each Analyze-confirmed change. When you save, a modal popup displays the destination filename while the write operation completes.

12) Optional External Rewrite Backend (`Config.Binoculars.Llm.Json`)

If an external backend is configured and available, rewrite generation can use any OpenAI-compatible endpoint. If the configuration is missing, disabled, or unreachable, the GUI defaults to the internal performer model.

Typical fields include:

- `llm.enabled`
- `llm.endpoint_url`
- `llm.request_path`
- `llm.model`
- `llm.api_key` or `llm.api_key_env`
- `llm.timeout_s`, `llm.max_tokens`, `llm.temperature`, `llm.top_p`
- Context controls (`context_chars_each_side`, `context_paragraphs_each_side`, `context_window_max_chars`)
- `llm.extra_headers`, `llm.extra_body`

13) Troubleshooting

Rewrite menu will not open:

- Run `Analyze` at least once before attempting to open the menu.
- Make sure no analysis or rewrite operation is currently running.

Rewrite generation is slow:

- The first run may require model warmup.
- Latency from an external endpoint can be significant.
- If needed, reduce context fields in `config.binoculars.llm.json`.

Undo button is disabled or Undo fails:

- Undo supports only one tracked operation at a time.
- If the document changes after that operation, Undo is invalidated.

B score does not match popup deltas:

- Popup deltas are approximate.
- Run a full `Analyze` for exact values.

Status-bar messages disappear:

- Non-analysis workflow messages are temporary; metrics are restored automatically.
- `Analyzing...` stays visible until analysis finishes.

Visual clutter is high:

- Use `Clear Priors` and continue the iteration process.

Practical summary:

- Use rewrites, synonym swaps, and direct edits for rapid local iteration.
- Rely on popup numbers for relative ranking.
- Use `Analyze` as the precise checkpoint.
- Use one-level `Undo` to quickly revert the last tracked change.
- Use `Clear Priors` for a periodic visual reset.