

Single-File Labels and Patch Extent in SWE-bench Lite: A Secondary Analysis for Repository Retrieval

Mason Carter, Lucas Bennett, Owen Turner

Abstract

Reference patches are convenient labels for repository localization, but their construction can restrict the retrieval question a benchmark can answer. This secondary analysis examines 300 public SWE-bench Lite test records from 12 repositories. Every reference patch changes one file, so file-count variation cannot support the initially considered association with issue-description length. We retain that finding and conduct an explicitly exploratory analysis of patch extent, defined as added plus deleted lines. The median patch changes six lines, and the median issue contains 150 word-like tokens. An ordinary least squares model of log one plus edited lines on log-base-two issue length, with repository fixed effects, estimates a slope of -0.0411 and an HC1 95 percent interval from -0.1295 to 0.0473. This weak descriptive association does not estimate retrieval effectiveness. The article explains why a single changed-file label cannot represent every file useful for understanding a task and proposes richer evaluation records for lexically anchored graph retrieval.

Keywords: SWE-bench Lite; repository localization; reference patches; exploratory regression; benchmark scope; graph retrieval.

1. What a reference patch labels

A reference patch identifies an observed implementation of a repair. It does not automatically identify every file a developer inspected or every implementation that could resolve the issue. Repository retrieval studies can nevertheless use changed files as convenient relevance labels because they are available in a structured format. That choice makes the evaluation reproducible while imposing an important boundary on its interpretation. A retriever can recover the changed file without collecting sufficient explanatory context, and it can retrieve an unchanged file that is useful even though the label treats that item as irrelevant.

SWE-bench organizes real repository issues and corresponding changes into an issue-resolution benchmark (Jimenez et al., 2024). This article studies the public Lite test subset as a dataset, rather than executing its evaluation harness. The objective is to inspect what its patch labels reveal about variation in edit scope and whether description length is a useful descriptive correlate. LARGER is relevant because its lexical anchors and local structural expansion seek context around an initial search match (Hu et al., 2026). Understanding the labels used to evaluate such retrieval is a separate contribution from measuring the performance of the retrieval system itself.

The analysis began with file count as a possible outcome. Inspection showed that all 300 retained records have exactly one reference-patch file. A regression of file count on issue length would therefore have no meaningful outcome variation. The analysis was changed before this manuscript was written to examine edited-line extent, while preserving the original finding and documenting the adjustment. The resulting model is exploratory. It should not be presented as a preregistered test or as confirmation of a preferred relationship that was specified before the dataset was inspected.

2. Public snapshot and extraction rules

The source is the SWE-bench Lite test split maintained by Princeton NLP (Princeton NLP, 2026). The locally retained input comprises three machine-readable responses containing 100 records each, downloaded on 25 October 2026. The request URLs, response hashes, and extraction code are preserved. The full set contains 300 distinct instance identifiers and 12 repositories. The extraction checks that the relevant issue-description and patch fields are not marked as truncated. No generated patch, agent trace, repository graph, or test-execution outcome is included in the analyzed data. The separate test patch field is not used to define the outcome.

For each reference patch, file paths are extracted from the unified-diff headers beginning with the file-difference marker. Unique source-side paths define the changed-file count. Added lines begin with a plus sign after excluding file-header lines; deleted lines begin with a minus sign after the analogous exclusion. Edited lines are their sum. A replacement can therefore contribute both a deletion and an addition. This is an explicit measure of textual change, not a count of distinct semantic operations. Context lines are not included, and the number of diff hunks is retained as an additional descriptive field.

Issue length is the number of word-like tokens matched by a declared regular expression. The expression includes identifier-like strings and numbers, so the variable is not an estimate of English prose length alone. A long code example can increase it without adding a comparable amount of natural-language explanation. This matters for lexical retrieval. BM25 and related approaches operate on token and document representations whose construction affects the matching process (Robertson and Zaragoza, 2009). The present count is a reproducible text descriptor; it is neither a BM25 score nor a direct measure of the lexical evidence available for identifying the relevant implementation.

The public snapshot is a selected benchmark collection. Its repository mix and issue eligibility reflect benchmark construction rather than random sampling from all maintenance tasks. Reporting the exact snapshot and processed variables makes the calculation replayable without making the collection representative. The FAIR principles support preserving identifiers and reusable metadata for this purpose (Wilkinson et al., 2016). A subsequent analysis should use the recorded response hashes to distinguish an exact reproduction from an update using a changed online dataset. Dataset identity is part of the empirical specification, not a detail that can be inferred from a familiar benchmark name.

3. Descriptive results and an exploratory model

All 300 patches change one file. The median edited-line count is six, the mean is 10.13, and the maximum is 76. The median hunk count is one. The median issue length is 150 word-like tokens. These quantities describe the supplied reference solutions and issue texts. They do not measure the number of files necessary for a correct explanation, the number of lines a coding agent would need to inspect, or the difficulty of validating a repair. A one-line behavioral correction can require extensive understanding, while a longer mechanical edit can be straightforward.

The revised outcome is $\log(1 + \text{edited lines})$. The main explanatory variable is $\log_2(\text{issue tokens})$, so a one-unit increase corresponds to a doubling of the token count. We estimate a pooled ordinary least squares specification and a specification adding repository indicator variables. The latter compares the relationship after allowing each repository its own intercept. HC1 covariance estimates address heteroskedasticity in the working regression (White, 1980). They do not make observations independent within repositories or convert the selected benchmark into a probability sample. The interval is conditional on this modeling approach and should be interpreted accordingly.

Table 1. Completed exploratory regressions. The slope relates a doubling of issue-token count to $\log(1 + \text{reference-patch edited lines})$; confidence intervals use HC1 covariance.

Specification	Slope	95 percent interval	R squared
Pooled OLS	-0.0468	-0.1323 to 0.0388	0.0040
Repository fixed effects	-0.0411	-0.1295 to 0.0473	0.0708

The fixed-effects estimate is close to the pooled estimate, and both intervals include zero. The results do not precisely distinguish a positive from a negative linear association under these specifications. They also do not establish that description content is irrelevant to repair or retrieval. The independent variable summarizes length and discards the specific identifiers, error traces, and constraints that can make a description useful. A small coefficient for length is compatible with a strong role for particular words. The model's outcome is reference-patch extent, which is another distinct limitation on what the association can mean.

Repository fixed effects account for mean differences, not every dependency among issues from the same project. Cluster-robust inference provides a framework for considering such dependence, with particular caution when the number of clusters is limited (Cameron and Miller, 2015). There are only 12 repositories here, with unequal representation. We therefore retain the simple HC1 calculation as a declared descriptive analysis and do not claim broad population significance. A more ambitious inferential study would need a sampling argument and a justified covariance strategy. Adding repository indicators alone cannot supply either one.

4. Constant labels constrain localization claims

The constant file count does not make the benchmark useless. It means that one proposed use of the variable is unsupported. In this snapshot, a file-level top-k metric asks whether the selected list contains the one reference-patch file for each issue. It cannot demonstrate performance as a function of the number of reference files because that number does not vary. A study wishing to investigate coordinated edits across files would need another collection or a different outcome. That requirement follows from the observed label distribution, without assuming anything about a particular agent's behavior.

The distinction between edit support and inspection support is especially important for structural retrieval. A caller, shared interface, fixture, or configuration file can clarify the changed implementation without appearing in the reference patch. A graph-expanded result can therefore introduce useful context that a changed-file metric does not reward. It can also introduce irrelevant neighbors. Neither possibility can be resolved by relabeling every connected file as relevant. A stronger evaluation would obtain task-specific judgments or execution-linked evidence for the proposed inspection set. The gold patch remains one observable object among several needed to assess context adequacy.

This is the evidentiary boundary relevant to LARGER's lexical-to-structural transition. The benchmark audit does not show that local expansion helps or harms. It identifies what a test based only on changed-file labels can observe. The appropriate next experiment would hold an agent and its information budget fixed while comparing retrieval policies on outcomes that include sufficiency of the presented context. It would retain the original ranked list and the additional inspected files. No such comparison, graph construction, or downstream coding run was performed in the completed analysis.

5. Length, extent, and the danger of causal shortcuts

The association should not be interpreted as the effect of writing a longer issue. Issue length can respond to the nature of the bug, reporter experience, repository conventions, discussion history, and the availability of a reproducible example. Patch extent can respond to implementation style and the selected solution. A causal diagram can clarify how shared causes and selection influence the relationship (Greenland et al., 1999). The current model includes no intervention on issue descriptions and no design identifying that intervention's effect. Its role is to describe a feature of the available dataset and to expose why that feature is an incomplete proxy for retrieval difficulty.

The log transformation also limits intuitive interpretation. A coefficient of -0.0411 is a difference in the expected transformed outcome under the fitted projection. It should not be read directly as 0.041 fewer edited lines, and it does not imply that every doubled description produces a smaller patch. Predictions on the original scale would require attention to the transformation and residual distribution. We do not use the model to forecast an individual issue's edit count. Reporting the coefficient in its actual units avoids adding a seemingly practical conclusion that the model was not designed to support.

Several measurement choices could change the association. Counting characters, natural-language tokens after removing code blocks, or unique identifiers would define different explanatory variables. Counting added lines only or changed syntax nodes would define different outcomes. Exploring those alternatives might be scientifically useful, but selecting among them after inspecting their favorable intervals would introduce another selection process. The present report retains one revised primary outcome and both declared specifications. It preserves the reason for the outcome change rather than presenting a long search as though it had yielded a single inevitable regression.

6. From a descriptive audit to a benchmark experiment

A future predictive analysis should use withholding that matches its intended transfer claim. Structured cross-validation supports keeping related observations together when shared structure can make random splits optimistic (Roberts et al., 2017). For repository tasks, a held-out project evaluates a different challenge from new issues in a familiar project. A temporal split introduces another distinction if earlier issues or fixes are available during development. These choices belong to a prospective prediction study. The regressions here are fitted descriptions of the full snapshot and contain no held-out performance result.

Model selection would require an additional boundary. Choosing features or transformations using a final test set changes what its reported error estimates. Nested evaluation addresses this distinction between tuning and assessing a selected procedure (Varma and Simon, 2006). A retrieval benchmark should apply the same principle to thresholds, expansion depth, and context budgets. It should also disclose what was learned from reference patches during development. A system that uses a patch-derived label at query time has access to information unavailable in normal issue resolution. The benchmark record should make that separation inspectable.

The minimum extension would add independently judged inspection context, a versioned repository snapshot, and an interaction trace connecting retrieval to a specified task. A file-label score could remain as a diagnostic alongside context sufficiency, resource consumption, and downstream verification. This would preserve the convenience of the existing labels while exposing their limitations. The extension should be evaluated on more than one patch-scope regime if it claims general repository exploration. It should not rely on issue length alone as a substitute for those richer distinctions.

7. Conclusion

The completed analysis establishes two narrow findings. First, all 300 public Lite reference patches in the retained snapshot change one file, preventing an informative file-count regression. Second, issue-token length has a small and imprecisely estimated linear association with transformed edited-line extent under pooled and repository-fixed-effect specifications. Neither finding measures a coding agent or a retriever. Their contribution is to clarify the measurement support available for evaluating repository localization. Lexically anchored graph exploration can then be tested against the questions its evidence actually addresses, with changed files, inspected context, and successful actions recorded as distinct objects.

References

- Hu, Yuntong, et al. "LARGER: Lexically Anchored Repository Graph Exploration and Retrieval." arXiv preprint arXiv:2605.16352 (2026).
- Jimenez, Carlos E., et al. "SWE-bench: Can Language Models Resolve Real-World GitHub Issues?" *International Conference on Learning Representations*, 2024. <https://arxiv.org/abs/2310.06770>.
- Princeton NLP. *SWE-bench Lite*. Hugging Face Datasets, public test split, JSON snapshot downloaded 25 Oct. 2026. https://huggingface.co/datasets/princeton-nlp/SWE-bench_Lite.
- White, Halbert. "A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity." *Econometrica*, vol. 48, no. 4, 1980, pp. 817-838. <https://doi.org/10.2307/1912934>.
- Cameron, A. Colin, and Douglas L. Miller. "A Practitioner's Guide to Cluster-Robust Inference." *Journal of Human Resources*, vol. 50, no. 2, 2015, pp. 317-372. <https://doi.org/10.3368/jhr.50.2.317>.
- Roberts, David R., et al. "Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure." *Ecography*, vol. 40, no. 8, 2017, pp. 913-929. <https://doi.org/10.1111/ecog.02881>.
- Varma, Sudhir and Richard Simon. "Bias in error estimation when using cross-validation for model selection." *BMC Bioinformatics*, vol. 7, no. 1, 2006. <https://doi.org/10.1186/1471-2105-7-91>.
- Robertson, Stephen, and Hugo Zaragoza. "The Probabilistic Relevance Framework: BM25 and Beyond." *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, 2009, pp. 333-389. <https://doi.org/10.1561/15000000019>.
- Greenland, Sander, Judea Pearl, and James M. Robins. "Causal Diagrams for Epidemiologic Research." *Epidemiology*, vol. 10, no. 1, 1999, pp. 37-48. <https://doi.org/10.1097/00001648-199901000-00008>.
- Wilkinson, Mark D., et al. "The FAIR Guiding Principles for Scientific Data Management and Stewardship." *Scientific Data*, vol. 3, 2016, article 160018. <https://doi.org/10.1038/sdata.2016.18>.