Topic Modelling-Based Code Review Hotspot Prediction

Tongwei Zhang
University of Waterloo
Waterloo, Canada
t98zhang@uwaterloo.ca

Zhaoyi Ge
University of Waterloo
Waterloo, Canada
zhaoyi.ge@uwaterloo.ca

Zhiao Wei
University of Waterloo
Waterloo, Canada
zhiao.wei@uwaterloo.ca

ACM Reference Format:
Tongwei Zhang, Zhaoyi Ge, and Zhiao Wei. 2025. Topic Modelling-Based Code Review Hotspot Prediction. In . ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Problem Statement and Motivation

Software development increasingly happens through pull requests, and those conversations now often involve AI coding agents—systems like GitHub Copilot, Cursor, and Claude Code that propose patches, rework modules, and scaffold tests. As agent-authored pull requests (Agentic-PRs) become more common, reviewers face a practical bottleneck: in a multi-file patch with a long discussion thread, where should they look first? This project addresses that question directly. Given a pull request's title, description, and code diff, we predict which specific files will attract the most reviewer attention—the review *hotspots*. We use the AIDev dataset as our empirical foundation, which provides Agentic-PRs with diffs, review comments, and follow-up changes; we will report the exact snapshot/DOI used for reproducibility [3].

Code review is expensive and time-sensitive; reviewers must decide what to read and in what order, often under pressure [1]. With agent-produced patches, this problem gets harder because of stylistic variation, multi-module edits, and vague rationales. If a tool could reliably predict the top-k files likely to draw comments, teams could focus attention better, reduce delays, and improve code review quality. Our goal is to provide a small, evidence-based "attention map" that helps reviewers prioritize their first passes.

There is a clear gap: prior work has characterized code review processes and outcomes, but we lack a shared, reproducible way to define "review hotspots" at file granularity using observable evidence from public repositories [1]. We also lack transparent baseline models—simple enough to run across ecosystems, yet strong enough to beat heuristics based only on code churn—that can predict those hotspots from a PR's text and structure. Agentic-PRs make this harder: agent patches often bundle changes across loosely coupled modules, and their descriptions may not map to subsystem boundaries the way human-authored PRs do [3].

We address this gap by defining hotspots (for each repository) using two observable signals in AIDev: inline or file-scoped code review comments that belongs to specific files or lines, and follow-up commits during review that touch those files. From these signals, we build a **attention distribution** over changed files and identify the attention distribution. For future PR's, we can then predict review hotspots from the PR's description and diff context using topic-level signals. Specifically, we model each PR's text and diff hunks to obtain a mixture over latent topics, and use those topics to rank changed files by expected attention. This design connects to classic work on degree-of-interest and developer context—particularly Kersten and Murphy's Mylar, which infers developer attention from structured signals [2]—but applies it to the code review setting, where the "interest" comes from the public record of comments rather than IDE traces. In short, we bridge a classic concept to a contemporary, large-scale, agent-centric environment.

We will deliver two main contributions: First, we will release a reusable baseline for hotspot prediction, including ground truth, evaluation metrics (Hit@k, Precision@k, Recall@k, nDCG@k), and open scripts. Second, we create a pipeline that employs topic modeling on repositories and predict attention-needed files for each Agentic PR. This work aligns with the MSR agenda by providing both a novel, topic-driven prediction method and a large-sample empirical analysis of Agentic-PR review focus.

2 Datasets and Tools

We base our work on the AIDev dataset [3]. AIDev is a new large-scale dataset comprised of 456,535 Agentic PRs created by Autonomous Coding Agents on GitHub. Specifically, the dataset features 7122 Agentic PRs and 6628 human PRs from popular repositories with more than 500 stars. We believe the size of this dataset is satisfactory for our purpose. For each PR, the dataset contains all associated review comments and commits; each commit includes its message and the set

Conference'17, Washington, DC, USA
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

of modified files. We will leverage this rich dataset to build and evaluate our proposed topic modeling pipeline. We will implement the pipeline using an established topic modeling library, such as BERTopic or Gensim, chosen for its suitability in handling our specific dataset.

3 Research Questions

RQ1. How can we define review hotspots in agentic PRs?
This question establishes the methodological foundation for our study. We operationalize review hotspots using observable signals—review comments and follow-up commits—and develop a systematic approach to identify and label hotspot files from the AIDev dataset, creating reproducible ground truth for evaluation.

RQ2. What are the characteristics of review hotspots in agentic PRs?
This question characterizes review hotspots empirically by analyzing comment and commit patterns across files in the AIDev dataset, examining how hotspots relate to file types, code complexity, and PR structure.

RQ3. How accurately can we predict review hotspots from PR descriptions and diffs?
This question evaluates the core predictive capability of our approach. We measure our model's ability to rank files by expected reviewer attention using Hit@k, Precision@k, Recall@k, and nDCG@k across different values of k.

RQ4. What topics drive hotspot predictions?
This question identifies which latent topics (e.g., authentication, testing, dependency management) correlate most strongly with hotspot files. We conduct case studies demonstrating how topic distributions align with actual review focus areas.

4 Methodology and Schedule

4.1 Data Collection and Attention Distribution

The AIDev dataset is publicly available online. We plan to primarily use a subset of the dataset, AIDev-Pop, which only contains PRs from popular repositories. We believe this helps improve the quality of review comments and increase the size of data for a single repository.

For each PR in a repository, our first step is to create a unified "document" that represents the complete discussion surrounding it. We construct a document corpus where each document corresponds to a single PR. To capture the full context, each document combines and links all available textual data including title, main body, commit messages and reviewer comments. Concurrently, we extract the list of all files modified in each PR. This list of files is linked directly to its corresponding PR document, which is essential for the correlation phase.

The next phase connects the abstract topics back to concrete files to build our attention distributions. We iterate through every PR in our dataset. For each PR, we take its generated topic distribution and attribute those topics to every file modified in that PR. In the end, we obtain a attention map which can be used to predict review hotspots for each file.

4.2 Baseline Construction and Evaluation

We first construct the baseline which consists of ground truth, evaluation metrics (Hit@k, Precision@k, Recall@k, nDCG@k). We construct the ground-truth based on the comments and code changes each file receives in a code review. As a final step, we evaluate our attention distribution using the above mentioned baseline.

4.3 Schedule and Use of AI

We break down the project into four milestones as described above (data collection, attention distribution creation, baseline creation and evaluation). We estimate each milestone takes roughly the same amount of time.

AI based tools might be used in data collection and topic modeling. We may use LLMs to perform data cleaning on the AIDev dataset. We may use embedding-based topic modeling tools to yield a more accurate distribution.

5 Threats to Validity and Mitigation Strategies

We acknowledge threats to validity following the standard framework [4].

5.1 Internal Validity

Definition of Review Hotspots. Our operationalization of review hotspots relies on two observable signals: inline or file-scoped review comments and follow-up commits during review. However, this definition may not perfectly capture the attention of the reviewer. Some critical files may receive few comments because reviewers implicitly trust them or defer discussion to offline channels. In contrast, files with many comments may reflect minor style disputes rather than substantive issues. To mitigate this threat, we will: (1) conduct manual validation on a random sample of 100 PRs, asking developers to independently label hotspot files and computing inter-rater agreement; (2) report case studies that qualitatively illustrate the correspondence between our hotspot labels and actual review focus.

5.2 External Validity

Dataset Representativeness. The AIDev dataset may not represent all agentic workflows. It may exhibit skewed distributions across project types (e.g., more open-source than enterprise applications) and PR complexity. We mitigate this by: (1) providing detailed dataset statistics; (2) performing stratified evaluation across different project categories; and (3) explicitly discussing the scope of our conclusions.

References

[1] Alberto Bacchelli and Christian Bird. Expectations, outcomes, and challenges of modern code review. In *Proceedings of the 2013 International Conference on Software Engineering*, ICSE '13, page 712–721. IEEE Press, 2013.

[2] Mik Kersten and Gail C. Murphy. Using task context to improve programmer productivity. In *Proceedings of the 14th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, SIGSOFT '06/FSE-14, page 1–11, New York, NY, USA, 2006. Association for Computing Machinery.

[3] Hao Li, Haoxiang Zhang, and Ahmed E. Hassan. The rise of ai teammates in software engineering (se) 3.0: How autonomous coding agents are reshaping software engineering, 2025.

[4] Claes Wohlin, Per Runeson, Martin Hst, Magnus C. Ohlsson, Bjrn Regnell, and Anders Wessln. *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated, 2012.