Topic Modeling Based Code Review Hotspot Prediction

12/2/25

Tongwei Zhang
Zhaoyi Ge
Zhiao Wei

UNIVERSITY OF WATERLOO

Why file-level Hotspots matter?

- Code review is
  - Expensive and time-sensitive
  - Done through pull requests
- Agentic PRs: AI tools (Copilot, Cursor, Claude Code) generate **multi-file** changes
- Reviewer pain point:
  - Multi-file patch
  - Long discussion thread
- **Question:** "Which files should I look at first?"

Our Problem

- We want to help reviewers by predicting:
  - Which changed files are **review hotspots**
  - How attention is distributed across files in a PR
- Focus: **Agent-authored PRs (Agentic PRs)** from AIDev
- Goal:
  - Build an **"attention map"** over files
  - Use a **simple, interpretable model** (topic × file category)

Gaps in Existing Work

- Modern code review well studied, but:
  - No **shared, reproducible definition** of file-level "review hotspots"
  - No **transparent baselines** for hotspot prediction on public PRs
- Agentic PRs complicate things:
  - Patches span loosely coupled modules
  - Descriptions may not map to clear subsystem boundaries

Research Questions

1. How can we define review hotspots in agentic PRs?

Operationalize using review comments + following commits

2. What are the characteristics of review hotspots in agentic PRs?

Analyze patterns across files, code complexity and PR structure.

3. How accurately can we predict review hotspots from PR descriptions and diffs?

Evaluate with Hit@k, Precision@k, Recall@k, nDCG@k, etc.

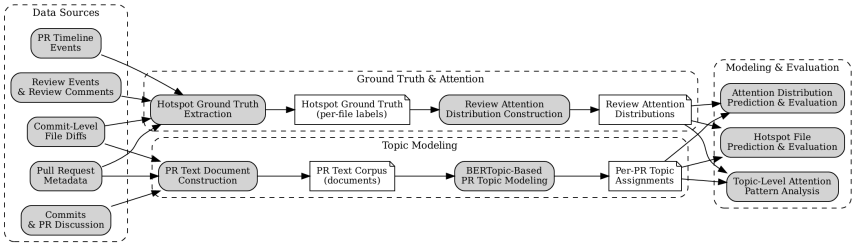
4. What topics drive hotspot predictions?

Identify key themes (auth, testing, etc.)

Datasets: AIDev

- AIDev:
  - 456k Agentic PRs across GitHub
  - Includes review comments, commits, metadata
- We use:
  - **AIDev-Pop**: PRs from popular repos (≥ 500 stars)
  - Intersection where we can:
    - Build file-level hotspots and attention
    - Build PR-level documents for topic modelling

Pipeline Overview



Ground Truth & Attention Distribution

For each PR:

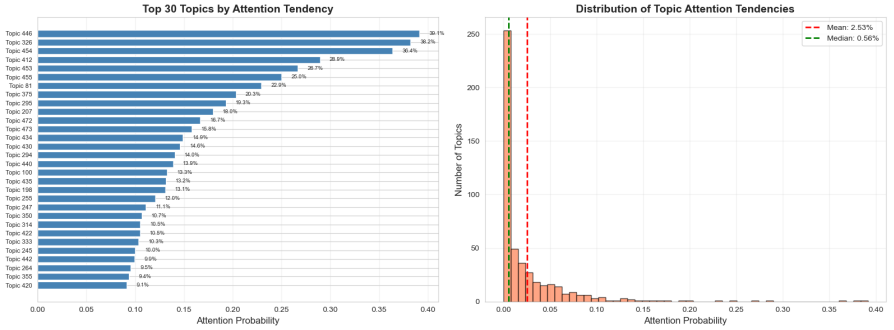
- Ground Truth:
  - Changed files
  - Files with in-line code review comment
  - Follow-up commit files
- Attention Distribution
  - Assign a positive attention score to files with review comments, and normalize
  - Assign a positive attention score to follow-up commit files, and normalize
  - If no such files, uniform distribution over changed files

PR Corpus and Topic Modeling

- PR Corpus
  - Title
  - Body
  - Commit messages

- BERTopic: A topic modeling technique that leverages transformers and c-TF-IDF to create dense clusters allowing for easily interpretable topics whilst keeping important words in the topic descriptions.
  - Gathered 465 Topics
  - Example: Topic 16 `['ocaml', 'rec', 'obj', 'mutable', '__show', 'print_endline', 'let', 'ml', 'repr', 'list_aux']`

RQ1: How can we define review hotspots?



Evaluation Setup

- **Data Partitioning**
  - **Total Dataset:** Merged PR Corpus (Topics) + Review Attention.
  - **Split Ratio:** 80% Training / 20% Testing.
    - *Note:* The code uses a fixed random seed (42) to ensure reproducibility.

- **Usage of Sets**
  - **80% Training Set:**
    - Used for **RQ2 and RQ4(Characteristics):** To learn the "Topic Attention Tendencies" and the "Topic x File Type" probability matrix.
    - We analyze *this* subset to understand what reviewers generally focus on.
  - **20% Test Set:**
    - Used for **RQ3 (Prediction):** To evaluate how well our learned rules generalize to unseen PRs.
    - Strictly held out during the learning phase to prevent data leakage.

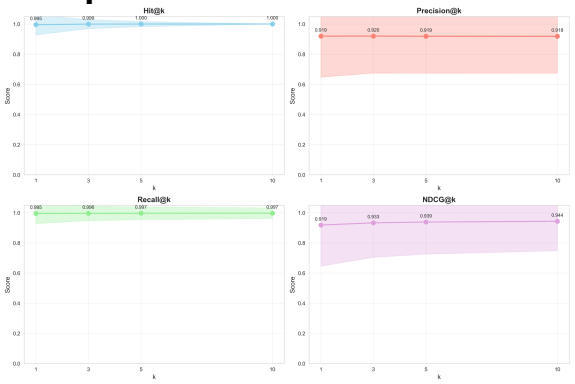
RQ2. What are the characteristics of review hotspots in agentic PRs?

- **Hypothesis:** Reviewer attention depends on both the *semantic context* (Topic) and the *functional role* (File Type) of the code.
- **File Categorization:** Files are classified into 5 types: `Source`, `Test`, `Docs`, `Config`, `Build`.
- **Model:** We estimate the conditional probability **P(Hotspot | Topic, FileType)**.
  - *Example:* For one topic, `Source` (e.g., `.js`, `.css`) files are hotspots, while `Config` files might be ignored.
  - *Example:* For another topic, `Test` files are the primary hotspots.

Hotspot files analysis step – to RQ3

- **Topic Inference:** For each PR, we have its **Dominant Topic**.
- **Rule Application:** We consult the "Topic x File Type" matrix (from RQ2) to see which file types are statistically "hot" for this topic.
  - *Example:* If Topic = "xx", then `Source` files are hot.
- **Prediction:** We scan all files in the PR.
  - If a file's type matches the "hot" types for the topic, it gets a score of **1.0** (Predicted Hotspot).
  - Otherwise, it gets a score of **0.0**.
- **Evaluation:** We compare this predicted list against the *actual* files that received comments (Ground Truth) using ranking metrics (Hit@k, NDCG, Precision, Recall).

RQ3. How accurately can we predict review hotspots from PR descriptions and diffs?



(1) High Hit Rate (Hit@k):

- **Hit@1 = 0.996:** The model successfully identifies at least one relevant hotspot file in the top-1 prediction 99.6% of the time.
- **Hit@10 = 1.000:** The top-10 predictions always contain relevant files.

(2) Strong Ranking Quality (NDCG@k):

- **NDCG@1 = 0.919:** Indicates highly relevant files are ranked at the very top.
- Improves to **0.944** at NDCG@10.

(3) Precision & Recall:

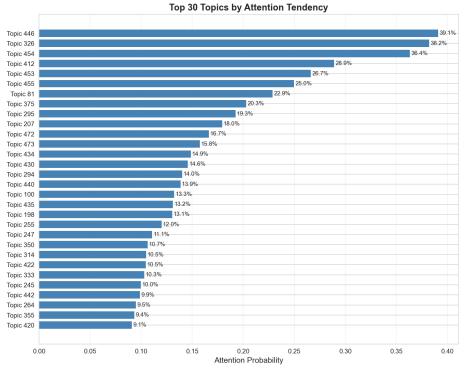
- **Precision@1 = 0.920:** The top predicted file is a true hotspot 92% of the time.
- **Recall@10 = 0.997:** The top 10 predictions capture almost all relevant files.

Topic order analysis step – to RQ4

- **Aggregation:** We grouped all Pull Requests in our training set by their **Dominant Topic** (assigned by BERTopic).
- **Metric Calculation:** For each topic, we calculated the **Attention Probability**:
  - *Formula:* **P(Attention | Topic) = Total Commented Files(from reviewer) / Total Changed Files(from PR)**
  - This measures the likelihood that any given file in a PR of this topic will receive a review comment.
- **Ranking:** We sorted all 484 topics by this probability to identify the "Top Hot Topics" vs. "Cold Topics".

RQ4. What topics drive hotspot predictions?

- **Topic Attention Tendency:** We identified specific topics that consistently attract higher reviewer attention.
- **Top "Hot" Topics:**
  - **Topic 446:** 39.1% Attention Probability (Highest)
  - **Topic 326:** 38.2% Attention Probability
  - **Topic 454:** 36.4% Attention Probability
- **Variance:** There is a significant variance in attention across topics. Some topics (like Topic 446) are nearly **40x** more likely to be commented on than the bottom topics (which have ~0% probability).



Threats to Validity

1. Construct Validity: (Measurement)

- **Proxy Metric**: We use *review comments* as a point for attention.
  - *Limitation 1*: Misses "silent reads" (careful reading without comments).
  - *Limitation 2*: Treats all comments equally (trivial nitpicks vs. critical errors).

2. Internal Validity: (Causality)

- **Classification**: Heuristic-based file mapping may misclassify non-standard structures.
- **Confounders**: We assume *topic* drives attention, but unmodeled factors exist:
  - PR size, author experience, and code complexity.

3. External Validity (Generalizability)

- **Data Source**: Open-source repositories.
- **Limitation**: May not generalize to **industrial environments** (e.g., strict compliance or security mandates).

UNIVERSITY OF WATERLOO

Thanks!