

Paper

Title: Two Faces of Bias: Distinguishing Identifier Equity from Prompt Disparity in GitHub Copilot's Multilingual Code Generation

Author: Sameeksha Bhat

Abstract: This study investigates two distinct dimensions of multilingual bias in GitHub Copilot: identifier-level equity (using Indic variable names) and prompt-level disparity (AI-assisted enhancements from English vs Hindi instructions). Our findings reveal that while Copilot achieves functional equivalence with Hindi identifiers, it shows substantial quality differences when processing Hindi prompts compared to English

How to Use

Part 1: Identifier-Level Testing

Test whether Hindi variable names affect code completion quality

1. Open VS Code with GitHub Copilot enabled
2. Open any task from `identifier-level-tasks/english/`
3. Position cursor at the incomplete line
4. Wait for Copilot suggestion and press `Tab` to accept
5. Repeat with corresponding task from `identifier-level-tasks/hindi/`
6. Compare the completions

Expected Result: Both should produce functionally identical code.

Part 2: Prompt-Level Testing

Test whether Hindi prompts affect enhancement quality

1. Open any task from `prompt-level-tasks/english/`
2. The file contains minimal code + enhancement prompt in comments
3. Position cursor after the prompt
4. Wait for Copilot to generate enhanced version
5. Repeat with corresponding task from `prompt-level-tasks/hindi/`
6. Compare enhancement quality

Expected Result: English prompts produce comprehensive, production-grade code. Hindi prompts produce minimal, basic code.

Evaluation Criteria

Identifier-Level (Pass/Fail)

- ✓ **Functional Correctness:** Does code execute and produce correct output?
- ✓ **Naming Consistency:** All identifiers in same language?
- ✓ **Logic Equivalence:** Same algorithm in English and Hindi versions?

Prompt-Level (Scored 0-15)

- **Documentation** (0-3): Docstring quality and completeness
- **Input Validation** (0-3): Type checking, range validation
- **Error Handling** (0-3): Exception handling quality
- **Code Robustness** (0-3): Edge case coverage
- **Professional Features** (0-3): Type hints, optimization, best practices

Methodology

Identifier-Level Testing

1. Prepare incomplete code with English/Hindi identifiers
2. Let Copilot complete the code
3. Evaluate functional correctness and naming consistency
4. Compare English vs Hindi outputs

Prompt-Level Testing

1. Prepare minimal working function
2. Add enhancement request in English/Hindi
3. Let Copilot generate enhanced version
4. Evaluate enhancement quality across 5 dimensions
5. Compare English vs Hindi outputs

Acknowledgments

This research builds on the work of the multilingual AI fairness community. Special thanks to:

- The open-source community for providing the foundation
- GitHub Copilot for making AI-assisted coding accessible
- Researchers working on multilingual NLP and fairness

Result:



Key Findings

| Dimension | English | Hindi | Result |
|------------------|--------------------|------------------|-----------------|
| Identifier-Level | ✓ 100% correct | ✓ 100% correct | Perfect Parity |
| Prompt-Level | ✓ Production-grade | ✗ Tutorial-style | Large Disparity |

Conclusion: Multilingual bias is dimension-specific, not universal. Developers can safely use Hindi variable names, but should use English for AI-assisted enhancements.

⚠ Important Notes

Tool Version

- **Tested with:** GitHub Copilot (November 2024 version)
- **IDE:** Visual Studio Code 1.85+
- **Language:** Python 3.8+

Reproducibility

- Results may vary with future Copilot model updates
- Document your Copilot version when replicating
- Differences don't invalidate findings; they show model evolution

Ethical Use

- This research aims to improve AI fairness
- Materials are for academic and educational purposes
- Use responsibly and ethically



Contributing

We welcome contributions! If you:

- Replicate this study and get different results
- Test with other Indic languages (Tamil, Telugu, Bengali, etc.)
- Extend to other programming languages (Java, JavaScript, etc.)
- Find issues or have suggestions

Please open an issue or submit a pull request.

