Bimodality & Naturalness: 
*LLMS! LLMS!! LLMS!!*

When Stochastic Parrots Write Code.....

Premkumar Devanbu
DECAL Laboratory
Computer Science

UC DAVIS

Thanks to NSF (Twice)
IARPA
Sandia Nat’l Labs
Humboldt Foundation
**FACT:** Codex, GPT-x, etc are now widely used to generate code.

- How much are people *using* this generated code? Does it help?

- How *good* is this code?
Does Codex help coders In “Vivo”?

n=410, survey, Github Devs;
30% code generated;
Helps productivity
74% “quick check”
..but…Non-func reqmnts?
Hard to control?

n=24; controlled study
+interview; @ Univ.
CoPilot not much help,
Many defects,
hard to Grok code,
..but subjects like it anyway!
Do LLMs help coders in “Vivo”?

**n=10k; Telemetry**
- 3% code generated
- 6% “Iteration” time reduction
- >30% suggestion acceptance

**n=2.6K; Survey + Telemetry**
- 23%-28% suggestion acceptance
- Acceptance rate correlates with self-reported productivity.
Personal take on Code LLMs

• Developers like them, Use them.

• Not clear they always fully understand the code they’re using, and what the “PSP” is for this.

• Prediction: In an astonishingly short time, every computer: laptops, mobile phones, toasters, microwaves, air-traffic control, nuclear power plants, cruise missiles…

Will be running code generated by an LLM!!!
AI-generated code will be running everywhere!!

Do LLMs generate buggy code?
Large Language Models and Simple, Stupid Bugs

Kevin Jesse  
*UC Davis*  
Davis, USA  
krjesse@ucdavis.edu

Toufique Ahmed  
*UC Davis*  
Davis, USA  
tfahmed@ucdavis.edu

Premkumar T. Devanbu  
*UC Davis*  
Davis, USA  
ptdevanbu@ucdavis.edu

Emily Morgan  
*UC Davis*  
Davis, USA  
eimorgan@ucdavis.edu
Methodology

- Simple, Stupid, Bugs 4 Java… One line bug fixes from 1000 projects. *(SS stubs4J, Karampatsis & Sutton 2020)*
- Go back in history, and find when they were injected (by human dev)
- Try the 🐝 with the prefix, and see…

……..Does 🐝 produce the 🦟 Or the 📕
All samples in dataset used were fixed before LLM training data was gathered.
Result

Codex produces fixed code

Codex produces buggy code TWICE as often

Something else
Result

Manual Review, 401 samples

- No Match: 60.25%
- Bug: 26.34%
- Patch: 13.41%
- Incorrect: 54.39%
- Unsure: 0.90%
- Patch: 2.85%
- Bug: 2.10%
Also looked at...

- When CoPilot generates Simple, Stupid Bugs, *were* they “stickier”?

- Good programmers Comment. Do Comments induce CoPilot not repeat human errors?
Take Aways…

• Programmers like LLM-based plugins.
• LLMs often recapitulate human errors.
• …when they do, these errors may be “sticky”.
• …but, we can improve their performance with comments.
• (Worry:) Devs use LLM-generated code without full review.