Code review pomocí velkého jazykového modelu

Valdemar

Duben 2025

Abstrakt

Tato práce se zabývá využitím velkých jazykových modelů (LLM) při procesu kontroly zdrojového kódu, tzv. *code review*, ve vývoji softwaru. Cílem je prozkoumat možnosti, přínosy a limity těchto modelů v reálném vývojářském workflow a navrhnout experimenty, které ověří jejich efektivitu ve srovnání s lidskými recenzenty.

1 Úvod do tématu

V současném softwarovém vývoji představuje *code review* nedílnou součást procesu zajišťující kvalitu zdrojového kódu. Slouží nejen k odhalování chyb, ale i k předávání znalostí mezi členy týmu, udržování konzistentního stylu kódu a zvyšování celkové udržovatelnosti systému. Tato praxe je klíčová zejména ve větších týmech a projektech s dlouhodobým vývojem.

V posledních letech se do vývojářského procesu stále více zapojují nástroje založené na umělé inteligenci. Jedním z nejvýraznějších pokroků v této oblasti jsou tzv. *velké jazykové modely* (LLM – Large Language Models), jako je ChatGPT, Claude, Code Llama nebo GitHub Copilot. Tyto modely dokáží porozumět strukturovanému i nestrukturovanému textu a generovat smysluplné odpovědi, komentáře nebo návrhy na základě vstupních dat.

Otázkou tedy je, do jaké míry lze tyto nástroje využít pro automatizaci nebo podporu code review. Může LLM odhalit stejné chyby jako zkušený programátor? Je jeho návrh na refaktoring použitelný v reálném prostředí? A především – může takový model plnohodnotně doplnit, nebo dokonce nahradit lidského recenzenta?

Tato seminární práce si klade za cíl popsat současný stav výzkumu v této oblasti, formulovat výzkumné otázky a navrhnout experiment, který pomůže zodpovědět, jak efektivní je využití LLM při provádění code review.

2 State-of-the-art

- Github Copilot, Claude, CodeRabbit, ChatGPT
- Články
- Výzkum pasivity AI

3 Výzkumné otázky

- Je LLM efektivnější než člověk při code review?
- Má nekritická pasivnost vliv na kvalitu code review?
- Má jazyk vliv na code review?
- (Umí LLM rozpoznat a navrhnout efektivní refaktoring v porovnání s běžnými nástroji (např. SonarQube, linters)?)
- (Jak dobře si LLM poradí s review kódu v méně běžných jazycích (např. Rust, Haskell, Elixir)?)
- (Je výhodnější provádět code review člověkem a pak ho doplnit LLM, nebo naopak?)

4 Návrh experimentu

- příprava prostředí(code base, code na přidání)
- jak jsem použil nástroje
- jak budu hodnotit výstupy od LLM

5 Výsledky a diskuze

- výsledky review
- opovězení na otázky
- jaké jsou dopady na týmovou práci

6 Závěr

- shrnutí zjištění
- moje omezení (no money na chat premium, a nedělám v týmu se seniorem který by mi dal dobrý cr a tak)