**Software Testing with Large Language Models: Survey, Landscape, and Vision**

Junjie Wang, Yuchao Huang, Chunyang Chen, Zhe Liu, Song Wang, Qing Wang

**Aim of the Research Study**

The aim of this paper is to systematically explore the emerging field of using Large Language Models (LLMs) in software testing. While LLMs like ChatGPT have gained popularity for their capabilities in generating code and documentation, their integration into software testing presents both opportunities and challenges. The study seeks to provide a comprehensive overview of existing research, identify current trends and limitations and propose a future vision for how LLMs can be meaningfully and responsibly adopted in testing activities.

**Methodology of the Study**

The authors conducted a detailed literature review of studies that focus on using LLMs across various stages of the software testing lifecycle. They organized the findings into a structured landscape covering different application scenarios such as test case generation, test oracle creation, bug localization, and test maintenance. Furthermore, they evaluated existing LLM-driven tools, documented their capabilities, and analyzed how LLMs are currently being integrated into testing workflows. The study also identifies key quality attributes, like correctness and reliability, which must be considered when using LLMs in testing tasks.

**Results of the Study**

The review reveals a rapid growth in research leveraging LLMs for software testing. The most common applications include generating test inputs, improving documentation-based test cases, and automating parts of regression testing. However, the paper also highlights significant limitations: current LLMs often generate incorrect or unverified content, struggle with domain-specific knowledge, and lack consistency in multi-step tasks. Moreover, many tools built on LLMs are still at the prototype stage, with limited real-world deployment.

**Implications for Research and Practice**

For researchers, this study lays out an important roadmap, calling for more rigorous benchmarks and various hybrid approaches that combine LLMs with verification tools. The paper suggests cautious but innovative adoption of LLMs in testing, particularly in areas where human oversight can guide and validate outputs. Ultimately, the work stresses the need for accountability, transparency, and trust when incorporating LLMs into safety- and quality-critical software development processes.