Fousseynou DIAKITE L0

MASTER 2 MCI BRESIL

BIG DATA ANALYTIC: Data modeling and source code generation and ai

Article 2: Usage for Code Generation and Data Analysis

Large language models (LLMs) have been touted to enable increased productivity in many areas of today's work life. Scientific research as an area of work is no exception: the potential of LLM-based tools to assist in the daily work of scientists has become a highly discussed topic across disciplines. However, we are only at the very onset of this subject of study. It is still unclear how the potential of LLMs will materialise in research practice. With this study, we give first empirical evidence on the use of LLMs in the research process. We have investigated a set of use cases for LLM-based tools in scientific research, and conducted a first study to assess to which degree current tools are helpful. In this paper we report specifically on use cases related to software engineering, such as generating application code and developing scripts for data analytics.

With the public release of ChatGPT in November 2022, large language models (LLMs) attracted widespread public attention. LLMs are machine learning models, often based on neural networks following a pre-trained transformer architecture, which have many parameters and have been trained on a large corpus of training data. 1 At the time of writing, hundreds of billions of model parameters are not a rare occurrence, as is training data with a trillion tokens.

The potential of using LLMs in aiding the research process is currently a highly discussed topic. In an editorial, Susarla et al. 3 explore the potential of LLMs in information systems research. They explore research question formulation, data collection, data analysis, and writing—as tasks in which LLMs could add benefit. Kasneci et al.

Code generation: Matrix multiplication in Java, using multi-threading. A complex aspect of object-oriented programming but which can be solved in a succinct manner.

Data analysis: Given some data, generate Python code to analyze the data provided through different tasks and questions. 3.

Data visualization: Given some data, generate R code to visualize it in different ways.

CORRECTNESS	BENCHMARK	15%
EFFICIENCY	BENCHMARK	45%
COMPREHENSIBILITY	BENCHMARK	40%
OVERALL RATING	RATE	100%