

УДК 004.891.2

Ж.К. Онайбаев, Н.А. Горбунова, Е.А. Спирина
Карагандинский университет имени академика Е.А. Букетова,
Караганда, Казахстан

ИСПОЛЬЗОВАНИЕ ИНСТРУМЕНТА ИСКУССТВЕННОГО ИНТЕЛЛЕКТА COPILOT

Аннотация. Дан анализ использования инструмента искусственного интеллекта Copilot, который предоставляет пользователям Visual Studio, Neovim и JetBrains возможности автозаполнения кода.

Zh.K. Onaibaev, N.A. Gorbunova, E.A. Spirina
Karaganda University named after academician E.A. Buketov
Karaganda, Kazakhstan

USING THE COPILOT ARTIFICIAL INTELLIGENCE TOOL

Abstract. The article analyzes the use of the Copilot artificial intelligence tool, which provides users of Visual Studio, Neovim and JetBrains with code autocompleteness capabilities.

Copilot - это инструмент, разработанный GitHub и OpenAI для того, чтобы помочь пользователям Visual Studio, Neovim и JetBrains в написании кода. Впервые об этом было объявлено 29 июня 2021 года.

Copilot использует OpenAI, модифицированную версию GPT-3 (языковую модель, предназначенную для создания текста, похожего на человека), предназначенную для создания реального кода. Copilot обучается в общедоступных репозиториях GitHub любой лицензии [1].

Проект упоминается как партнер по программированию, извлекает контекст из кода, над которым вы работаете, предлагая целые строки или целые функции. Этот инструмент поможет быстро найти альтернативные способы решения проблем, написания тестов и изучения, новых API без поиска ответов на таких сайтах, как Stack Overflow [2].

Поскольку он основан на машинном обучении, он учится, когда вы его используете. По мере того, как вы печатаете, он адаптируется к тому, как вы пишете код, чтобы помочь вам выполнять свою работу быстрее. На мой взгляд, в Copilot интересно то, что он обычно генерирует исходный код, то есть код, который не дословно представлен в обучающих данных. Однако мы должны помнить, что он

совершенно неспособен писать творческий код. Креативность на данный момент, все еще в руках людей [3].

Данный проект является естественным развитием кодогенерации. Сначала команда GitHub начала с автодополнения кода, а сейчас к генеративным решениям на основе искусственного интеллекта, обученного на огромных объемах общедоступного кода. Он может работать только с тем, что уже существует - новые решения все равно потребуют людей. Точно так же, как Ruby on Rails когда-то сделал разработчиков более эффективными, автоматически генерируя множество стандартных каркасов из веб-проектов, Copilot должен экономить время разработчиков [2].

Проект также будет полезен для тестирования программного обеспечения. Тесты являются основой любого надежного проекта разработки программного обеспечения. Импортируйте пакет модульных тестов и позвольте Copilot предложить тесты, соответствующие вашему коду реализации. Предлагающий код никогда не следует использовать вслепую, особенно при обучении количества кода с GitHub, который может иметь переменное качество [3].

Стоит не забывать, что Copilot даже не пытается скомпилировать код, не проверяет, работает ли он, и не задумывается о том, действительно ли он выполняет то, что говорится в документации. Жаловаться на качество кода, написанного Copilot не надо. С точки зрения исследований в области машинного обучения и синтеза языков - это большой шаг вперед. Но мы также должны понимать, что код, который не работает, не тестирует крайние случаи, использует устаревшие методы, является многословным и создает технический долг, может быть большой проблемой.

Как показывает практика, чем меньше кода, тем легче его поддерживать и понимать. Обычно, если не знаем, как что-то сделать, мы гуглим. Например, нужно написать регулярное выражение, идем на Stack Overflow и находим решение. Или находим документацию. Читаем и создаем решение. Если пользоваться Copilot, нам бы выдало одно решение. Но первый метод дал нам более обширные знания. Обсуждения на Stack Overflow помогли понять проблемы, а также объяснили ограничения механизма регулярных выражений. Второй подход будет хуже как для опытных, так и для начинающих разработчиков.

Как мы видим данный проект - это предварительная версия очень новой технологии, включающий лучшие практики взаимодействия

человека с компьютером, разработки программного обеспечения, тестирования и многих других дисциплин.

Список использованных источников

1. Gershgorn, Dave. 29 June 2021. "GitHub and OpenAI launch a new AI tool that generates its own code".
2. The Verge. 7 July 2021. "GitHub's automatic coding tool rests on untested legal ground".
3. Jeremy Howard. 19 Jul 2021. "Is GitHub Copilot a blessing, or a curse?"

УДК 37.02

Н.А. Горбунова, А.А. Оралова

Карагандинский Университет им. Е.А. Букетова,
Караганда, Казахстан.

STREM-ТЕХНОЛОГИЯ КАК СРЕДСТВО РАЗВИТИЯ ТВОРЧЕСКОЙ ДЕЯТЕЛЬНОСТИ ОБУЧАЮЩИХСЯ

Аннотация. В данной статье анализируется инновационный подход к организации процесса образования, ориентированного на формирование научно-технической компетентности с помощью STREM технологий.

N.A.I. Gorbunova, A.A. Oralova

Karaganda University named after E.A. Buketov,
Karaganda, Kazakhstan.

STREM-TECHNOLOGY AS A MEANS OF DEVELOPING THE CREATIVE ACTIVITY OF STUDENTS

Abstract. This article explores a new approach to the organization of education, focused on the formation of the scientific and technical elite with the help of STREM technologies. The essence of STREM-learning is considered, the advantages and disadvantages of STREM-technology are defined.

Введение. Основная миссия современного образования – это подготовка конкурентно способных специалистов, способных к междисциплинарной, интегрированной деятельности, направленной на решение существующих в обществе проблем. Сегодня можно наблюдать противоречие между темпами технико-технологического