

ГУАП

КАФЕДРА №41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

доцент, канд. техн. наук
должность, уч. степень, звание

подпись, дата

Е.Л. Турнецкая
инициалы, фамилия

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №6

Применение ИИ в тестировании

по курсу: ИНЖЕНЕР ПО ТЕСТИРОВАНИЮ

РАБОТУ ВЫПОЛНИЛИ

СТУДЕНТ гр. №

2212

14.11.2024

подпись, дата

М.С. Мухин
В.Ю. Прокофьев
А.А. Ибрагимов
Г.С. Солдатенков
инициалы, фамилия

1. Цель работы:

Получение практических навыков применения искусственного интеллекта в тестировании программного обеспечения.

2. Описание нейросетевой модели:

В работе использовалась большая языковая модель Yandex.GPT, разработанная компанией Яндекс. Эта модель применяется для генерации текста, перевода, ответов на вопросы и выполнения других задач обработки естественного языка.

В сфере тестирования ИИ может использоваться для автоматизации создания тестовых данных, генерации тестовой документации, анализа результатов тестирования и даже для автоматизации части процесса тестирования.

3. Генерация тестовых данных:

Запрос: "Сгенерируй 10 тестовых email адресов для регистрации пользователей на сайте, используя домен @example.com. Адреса должны быть разнообразными, включать цифры, специальные символы и буквы верхнего и нижнего регистра."

Ответ Yandex.GPT:

Сгенерируй 10 тестовых email адресов для регистрации пользователей на сайте, используя домен @example.com. Адреса должны быть разнообразными, включать цифры, специальные символы и буквы верхнего и нижнего регистра.

1. User1A@example.com
2. User2B@example.com
3. User3_C@example.com
4. User4D98@example.com
5. User5_E_F@example.com
6. User6G7H@example.com
7. User7I_JKL@example.com
8. User8MNO@example.com
9. User9PQRST@example.com
10. User0UVWXYZ@example.com

 Copy answer

Комментарий: Yandex.GPT в течение 5 секунд успешно сгенерировал набор разнообразных email адресов, полностью соответствующих заданным критериям.

Запрос: "Дай определения тестирования методом «белого ящика», фаззинг-тестирования и попарного тестирования."


Ответ Yandex.GPT:

Дай определения тестирования методом «белого ящика», фаззинг-тестирования и попарного тестирования

Тестирование методом «белого ящика» — это метод тестирования программного обеспечения, при котором тестирующий имеет доступ к коду программы и может проверять его на наличие ошибок и уязвимостей. Этот метод также называется структурным тестированием, так как он основан на анализе структуры программы.

Фаззинг-тестирование — это автоматизированный метод тестирования программного обеспечения, который заключается в подаче случайных или специально сгенерированных данных в программу с целью вызвать неожиданные или нежелательные реакции. Фаззинг-тестирование помогает обнаружить уязвимости и ошибки в программе, которые могут быть не замечены при ручном тестировании.

Попарное тестирование — это метод проектирования тестовых сценариев, который основан на выборе пар входных параметров и проверке всех возможных комбинаций этих параметров. Попарное тестирование позволяет сократить количество тестов и обеспечить более эффективное покрытие кода.

 Copy answer

Комментарий: Yandex.GPT предоставил достаточно корректные определения, хотя ожидалось более подробное описание каждого из видов. Если мы сделаем запрос с конкретными критериями (объем информации, конкретная необходимая информация, комментарии и более углубленный ответ), Yandex.GPT выдал бы более подробный ответ. Для машины, которая умеет обучаться и сохраняет контекст предыдущих запросов очень не сложно справиться с такой задачей.

3.1 Генерация тестовых сценариев и отчетов:

Запрос: "Составь тестовый сценарий для проверки функциональности кнопки "Добавить в корзину" на странице товара в интернет-магазине."

Ответ Yandex.GPT:

Тестовый сценарий для проверки функциональности кнопки «Добавить в корзину» на странице товара в интернет-магазине

Цель: проверить, что кнопка «Добавить в корзину» корректно добавляет товар в корзину и отображает его количество.

Предусловия:

Пользователь авторизован в системе.

На странице товара отображается информация о названии, цене, изображении и описании товара.

В корзине пользователя нет товаров.

Шаги:

1. Перейти на страницу товара.
2. Нажать на кнопку «Добавить в корзину».
3. Проверить, что товар добавлен в корзину.
4. Проверить, что в корзине отображается корректное количество добавленных товаров (1 штука).
5. Вернуться на страницу товара и снова нажать на кнопку «Добавить в корзину».
6. Проверить, что количество товаров в корзине увеличилось до 2 штук.
7. Повторить шаги 5–6 несколько раз, чтобы убедиться, что каждый раз при нажатии на кнопку «Добавить в корзину» количество товаров увеличивается на 1 штуку.
8. Закрыть браузер и открыть его заново.
9. Авторизоваться в системе под тем же аккаунтом.
10. Перейти на страницу корзины.
11. Убедиться, что все товары, добавленные ранее, отображаются в корзине.
12. Удалить один из товаров из корзины.
13. Снова перейти на страницу товара и нажать на кнопку «Добавить в корзину».
14. Проверить, что после удаления товара из корзины, при повторном добавлении этого же товара, количество товаров в корзине увеличивается на 1 штуку.

Ожидаемый результат: кнопка «Добавить в корзину» работает корректно, товар добавляется в корзину, отображается его количество, при удалении товара из корзины и повторном добавлении, количество корректно изменяется.

Запрос: "Напиши отчет по дефекту, обнаруженному в веб-приложении: при попытке загрузки файла размером более 10 Мб, приложение выдает ошибку без информирования пользователя о причине"

Ответ Yandex.GPT:

Отчёт по дефекту

Описание дефекта:

При попытке загрузки файла размером более 10 МБ в веб-приложении происходит ошибка без информирования пользователя о причине.

Шаги для воспроизведения:

Перейти на страницу загрузки файлов в веб-приложении.

Выбрать файл размером более 10 МБ.

Нажать кнопку «Загрузить».

Наблюдать за появлением ошибки без сообщения о причине сбоя.

Ожидаемое поведение:

Веб-приложение должно корректно обрабатывать загрузку файлов размером до максимально допустимого размера, указанного в настройках системы, и информировать пользователя об успешной загрузке или указывать причину ошибки при возникновении проблем.

Фактическое поведение:

При загрузке файла размером более 10 МБ веб-приложение выдаёт ошибку без сообщения о её причине.

Влияние на пользователей:

Отсутствие информации о причинах ошибки может привести к потере данных и снижению доверия к веб-приложению.

Версии ПО и ОС:

Информация о версиях программного обеспечения и операционных систем не указана.

Временные рамки:

Ошибка возникает сразу после попытки загрузки файла размером более 10 МБ.

Рекомендации по исправлению:

Необходимо добавить проверку размера загружаемого файла и соответствующее сообщение об ошибке или превышении допустимого размера. Также рекомендуется провести тестирование с различными размерами файлов для подтверждения корректной работы функции загрузки.

Комментарий: Ответы были выданы буквально за 5-7 секунд и полностью соответствуют запросу, но данные ответы требуют проверки на корректность в рамках полного сценария и отчета. Ответы любого GPT нуждаются в проверке, и ответы Yandex.GPT не являются исключением, об этом нельзя забывать.

3.2 Теоретический вопрос:


Запрос: "Сформулируй теоретический вопрос по техникам тестирования с тремя вариантами ответа. Вопрос должен касаться выбора типа тестирования в зависимости от жизненного цикла ПО."

Ответ Yandex.GPT:

Сформулируй теоретический вопрос по техникам тестирования с тремя вариантами ответа. Вопрос должен касаться выбора типа тестирования в зависимости от жизненного цикла ПО

Какой тип тестирования лучше всего подходит для выявления дефектов на ранних стадиях разработки программного обеспечения?

1. Функциональное тестирование.
2. Регрессионное тестирование.
3. Тестирование на проникновение (пентест).

 Copy answer

Какой из этих ответов правильный?

Функциональное тестирование.

 Copy answer

Комментарий: Ответ снова был выдан очень быстро и соответствует запросу. Но для полной уверенности корректность ответа следует проверить самостоятельно.

3.3 Самостоятельный запрос:

Запрос: "Как можно использовать ИИ для автоматизации генерации тестовых данных на основе анализа требований к ПО?"

Ответ Yandex.GPT:

Искусственный интеллект может автоматизировать генерацию тестовых данных на основе анализа требований к ПО следующим образом:

1. **Анализ требований.** ИИ может анализировать текстовые описания требований и извлекать из них информацию о типах данных, которые должны быть включены в тесты. Это могут быть, например, диапазоны значений, допустимые и недопустимые значения, а также специфические требования к формату данных.
2. **Генерация тестовых сценариев.** На основе извлечённой информации ИИ может генерировать тестовые сценарии, включая создание примеров входных данных для проверки различных аспектов функциональности ПО.
3. **Создание тестовых данных.** Используя алгоритмы генерации данных, ИИ может создавать наборы тестовых данных, соответствующие требованиям. Например, для тестирования функций работы с числами ИИ может сгенерировать набор чисел в заданных диапазонах, а для тестирования текстовых полей — создать разнообразные текстовые строки.
4. **Проверка на соответствие.** Сгенерированные данные могут быть проверены на соответствие требованиям и стандартам качества, что позволяет обеспечить их корректность и эффективность при тестировании ПО.
5. **Обновление и оптимизация.** В процессе тестирования ИИ может собирать обратную связь о качестве сгенерированных данных и использовать её для обновления алгоритмов генерации, чтобы улучшить качество тестов и повысить эффективность процесса тестирования.

Таким образом, использование ИИ для автоматизации генерации тестовых данных позволяет ускорить процесс разработки, повысить качество тестирования и обеспечить более надёжное функционирование программного обеспечения.

Комментарий: Ответ GPT достаточно корректный, включает в себя краткое описание. Данный запрос демонстрирует потенциальные возможности ИИ в тестировании.

4. Результаты оценки корректности и объяснение:

Корректность ответов Yandex.GPT варьировалась. Простые запросы (генерация email адресов, некоторые определения) были выполнены быстро и качественно. Более сложные запросы (тестовые сценарии, отчеты,

теоретические вопросы) потребовали дополнительной проверки (необходимо было уточнить правильный ответ на вопрос с тремя вариантами ответа). В некоторых случаях ответы были неполными. Это связано с ограничением модели и необходимостью точной формулировки запроса. В целом, Yandex.GPT отлично справился со своей задачей.

5. Вывод:

Практическая работа продемонстрировала как полезность, так и ограничения использования Yandex.GPT в автоматизации задач тестирования программного обеспечения. Результаты показали, что модель эффективно справляется с генерацией простых тестовых данных, таких как email-адреса, и предоставлением кратких определений терминов из области тестирования. Это подтверждает потенциал ИИ для автоматизации рутинных и повторяющихся задач, что может существенно сократить время и ресурсы, затрачиваемые на тестирование. Генерация простых тестовых данных, например, для нагрузочного тестирования, может быть значительно ускорена с помощью Yandex.GPT.

Однако, при переходе к более сложным задачам, таким как создание подробных тестовых сценариев, отчетов о дефектах или формулирование глубоких теоретических вопросов, эффективность Yandex.GPT значительно снижается. Ответы модели часто требуют значительной доработки, включая редактирование, дополнение и проверку на корректность. Это связано с несколькими факторами: не всегда точным пониманием контекста запроса, недостатком глубинного понимания тонкостей тестирования и неспособностью самостоятельно проверять логическую цельность сгенерированной информации.

Полученные в ходе работы знания и навыки включают не только практическое применение Yandex.GPT для генерации тестовой документации, но и критическое оценивание результатов, полученных от ИИ. Мы научились формулировать запросы более точно и эффективно, а также оценивать достоверность и полноту ответов модели. Понимание ограничений Yandex.GPT является ключевым фактором успешного использования ИИ в тестировании, поскольку оно позволяет более адекватно оценивать его возможности и планировать работу с учетом необходимости ручной проверки и доработки результатов.

Основной проблемой, возникшей в ходе работы, было достижение высокой точности и полноты ответов на сложные вопросы. Решение этой проблемы требует не только более детальной формулировки запросов, но и использования дополнительных техник. Для дальнейшего улучшения эффективности необходимо более глубокое изучение возможностей модели, а также освоение специфических техник работы с большими языковыми

моделями в контексте задач тестирования ПО. В будущем важно исследовать возможности использования более специализированных моделей ИИ, обученных на данных из сферы тестирования.

6. Список использованных источников:

1. <https://practicum.yandex.ru/blog/neyroset-yandex-kak-polzovatsya/>