

Praktikum Smart Data Analytics

Sommersemester 2024

3. Übungsblatt

Automatisierte Erstellung von Playwright UI Tests mit Local LLM für die Cadenza Web App [Abgabedeadline:01.07.2024]

In diesem Übungsblatt lernen Sie, wie man mithilfe eines lokalen Language Models (LLM) automatisiert UI Tests für eine Webanwendung (Disy Cadenza) mit Playwright generiert.

UI Tests (User Interface Tests) sind eine entscheidende Komponente im Softwareentwicklungsprozess, da sie sicherstellen, dass die Benutzeroberfläche einer Anwendung wie erwartet funktioniert. UI Tests überprüfen, ob verschiedene Elemente der Benutzeroberfläche korrekt angezeigt werden und ob Benutzerinteraktionen, wie das Klicken auf Buttons oder das Ausfüllen von Formularen, ordnungsgemäß funktionieren.

Die Erstellung und Wartung von UI Tests kann zeitaufwendig und fehleranfällig sein, insbesondere wenn die Anwendung regelmäßig aktualisiert wird. Hier kommt die **Automatisierung von UI Tests** ins Spiel, die diesen Prozess erheblich erleichtert und beschleunigt. Mit Werkzeugen wie **Playwright** können Entwickler automatisierte UI Tests erstellen, die wiederholbar und zuverlässig sind.

Language Models (LLMs) wie GPT-4 bieten eine innovative Möglichkeit, den Prozess der Testautomatisierung weiter zu optimieren. Durch die Verwendung von LLMs können UI Testskripte automatisch generiert werden, basierend auf einfachen Beschreibungen der Benutzerinteraktionen. Dies reduziert den manuellen Aufwand und ermöglicht es Entwicklern, sich auf die Verbesserung der Anwendung zu konzentrieren, anstatt auf das Schreiben von Tests.

Disy Informationssysteme GmbH ist ein führender Anbieter von Softwarelösungen für Datenanalyse, GIS und Berichtswesen. Die **Cadenza Web App** ist ein zentrales Produkt von Disy und ermöglicht Nutzern eine umfassende Datenanalyse und -visualisierung. Cadenza bietet Funktionen zur Erstellung und Verwaltung von Berichten, zur Visualisierung von Geodaten und zur Durchführung komplexer Datenanalysen. Weitere Informationen finden Sie auf der [Disy-Website](#).

Aufgabe 1: Einführung in Playwright und Cadenza

a) Verstehen Sie Playwright:

- Recherchieren Sie, was Playwright ist und welche Vorteile es im Vergleich zu anderen UI-Testframeworks bietet.
- Installieren Sie Playwright in Ihrem Arbeitsumfeld. Dokumentieren Sie den Installationsprozess und stellen Sie sicher, dass die Installation erfolgreich war.

b) Erste Schritte:

- Erstellen Sie ein einfaches Playwright-Skript, das eine Website Ihrer Wahl öffnet, einen bestimmten Text auf der Seite überprüft und einen Screenshot der Seite speichert. Dokumentieren Sie Ihre Schritte und fügen Sie den Quellcode Ihres Skripts bei.

c) Verstehen Sie die Cadenza Web App:

- Recherchieren Sie die Hauptfunktionen und Benutzeroberfläche der Cadenza Web App. Erstellen Sie eine kurze Dokumentation über die wichtigsten Funktionen, die für Ihre UI-Tests relevant sein könnten.
- Navigieren Sie durch die Cadenza Web App und machen Sie sich mit den verschiedenen Modulen und Funktionen vertraut. Dokumentieren Sie die wichtigsten Bereiche und Aktionen, die in Ihren Tests abgedeckt werden sollen.

Aufgabe 2: Generieren von Tests mit Local LLM

a) Einrichtung des Local LLM:

- Installieren und konfigurieren Sie ein lokales Language Model, das für die automatische Codegenerierung geeignet ist. Dokumentieren Sie den Installations- und Konfigurationsprozess.

b) Automatisierte Testgenerierung:

- Erstellen Sie ein Python-Skript, das mithilfe des lokalen LLMs Playwright-Tests für die Cadenza Web App generiert. Das Skript sollte in der Lage sein, basierend auf einer gegebenen Beschreibung der Benutzerinteraktionen, entsprechende Playwright-Skripte zu erstellen.
- Beispielbeschreibung: "Öffne die Startseite der Cadenza Web App, klicke auf den Login-Button, gib im Login-Formular die E-Mail-Adresse und das Passwort ein, und klicke auf 'Anmelden'".
- Lassen Sie das LLM ein entsprechendes Playwright-Skript generieren. Führen Sie das generierte Skript aus und dokumentieren Sie die Ergebnisse.

Aufgabe 3: Sammlung von Finetuning-Daten für das LLM

a) Datenaufbereitung:

- Sammeln und bereiten Sie Daten auf, die das lokale LLM für zukünftiges Finetuning verwenden kann. Dies kann Interaktionsprotokolle, generierte Skripte, Fehlerberichte und Optimierungsschritte umfassen.
- Dokumentieren Sie, wie die Daten gesammelt wurden, und stellen Sie sicher, dass sie strukturiert und annotiert sind.

b) Erstellung eines Finetuning-Datensatzes:

- Erstellen Sie einen strukturierten Datensatz, der verwendet werden kann, um das LLM weiter zu trainieren. Dieser Datensatz sollte Eingabeaufforderungen, generierte Playwright-Skripte und zugehörige Metadaten umfassen.
- Beschreiben Sie das Format des Datensatzes und wie er für das Finetuning verwendet werden kann.

Aufgabe 4: Finetuning des Local LLM

a) Finetuning-Vorbereitung:

- Bereiten Sie das lokale LLM für das Finetuning vor, indem Sie den zuvor erstellten Datensatz verwenden. Stellen Sie sicher, dass alle erforderlichen Schritte und Einstellungen dokumentiert sind.

b) Durchführung des Finetunings:

- Führen Sie das Finetuning des lokalen LLM mit dem vorbereiteten Datensatz durch. Dokumentieren Sie den Prozess und die dabei auftretenden Herausforderungen.

c) Evaluierung der verbesserten Leistung:

- Evaluieren Sie die Leistung des finetuneten LLM, indem Sie neue Playwright-Tests generieren und deren Qualität und Effizienz überprüfen.
- Vergleichen Sie die Ergebnisse mit den vorherigen Generationen und dokumentieren Sie die Verbesserungen und möglichen weiteren Optimierungspotenziale.

Aufgabe 5: Validierung und Optimierung der generierten Tests

a) Validierung der Tests:

- Entwickeln Sie eigene Methoden zur Validierung der generierten Playwright-Tests. Überlegen Sie, welche Aspekte (z.B. Browser-Kompatibilität, Robustheit, Zuverlässigkeit) überprüft werden müssen und wie Sie dies umsetzen können.

Literatur/Quellen

- LLM finetune: <https://www.lakera.ai/blog/llm-fine-tuning-guide>
- Disy & Cadenza: <https://www.disy.net/de/produkte/cadenza/datenanalyse-software/>
- Playwright: <https://playwright.dev>
- Andere Quellen, die bei der Sammlung von Kontextinformationen nützlich sein können:
 - <https://github.com/vladikoff/playwright-ai>
 - <https://github.com/mayt/BrowserGPT>
 -