Herausgeber Prof. Dr. Barbara Dörsam

Schriftreihe Bachelor-Resümee

Forschungsbereich **Digital Publishing**

Plagiatserkennung in Webauftritten

Entwurf für einen Ansatz zur Erkennung von dupliziertem Quellcode in Webauftritten

Ellen Kühling

Impressum

Hochschule der Medien

Nobelstrasse 10

70569 Stuttgart

www.hdm-stuttgart.de

0711 8923-0

Autor

Ellen Kühling

Betreuer

Prof. Dr. Barbara Dörsam

Datum

Juli 2022

Wirtschaftsingenieurwesen Medien

www.hdm-stuttgart.de/wing

Layout

Jochen Riegg

Fotos und Illustrationen

Innenteil: Ellen Kühling

Bachelor-Resümee

Plagiatserkennung in Webauftritten

Entwurf für einen Ansatz zur Erkennung von
dupliziertem Quellcode in Webauftritten

Ellen Kühling

Juli 2022

Die Autorin

Ellen Kühling studierte Druck- und Medientechnologie mit dem Schwerpunkt Digital Publishing an der Hochschule der Medien (HdM) in Stuttgart. Im Rahmen ihrer Bachelorarbeit untersuchte sie unterschiedliche Tools zur Erkennung von dupliziertem Quellcode. Auf den Ergebnissen basierend erstellte sie anschließend einen Entwurf für einen Ansatz zur Erkennung von dupliziertem Quellcode.

Inhaltsverzeichnis

1. Kurzfassung

In dieser Publikation werden der Ablauf und die Ergebnisse der Bachelorarbeit „Plagiatserkennung in Webauftritten“ vorgestellt. Das Ziel der Arbeit war es, einen Ansatz für die Plagiatserkennung in Webauftritten zu finden. Als Grundlage dafür diente die Analyse von Webauftritten, welche von Studierenden im Rahmen der Veranstaltung Web-Technologien an der Hochschule der Medien erstellt wurden. Für die Zielerreichung wurden verschiedene Tools hinsichtlich ihres Nutzens bezüglich der Plagiatserkennung in Webauftritten getestet und anschließend bewertet. Die Evaluation der Tools erfolgte anhand Testbeispielen, welche durch die Analyse der bestehenden Webauftritten und den daraus resultierenden Kriterien konstruiert wurden. Auf den Ergebnissen der Testdurchführungen basierend wurde ein Entwurf für einen Lösungsansatz für das Aufdecken potenzieller Plagiate erarbeitet.

2. Hintergrund

Um die Grundkenntnisse der Webentwicklung zu erlangen, belegen Studierende des Studiengangs Wirtschaftsingenieurwesen Medien an der Hochschule der Medien in Stuttgart die Veranstaltung Web-Technologien. Die Prüfungsleistung ist dabei die Umsetzung eines Webauftritts, welcher aus mehreren HTML- und den dazugehörigen CSS-Dateien besteht. Die Erfahrungen und Erkenntnisse der Prüfungsabgaben in den vergangenen Semestern gaben Anlass zur Vermutung, dass ein kleiner Anteil der von den Studierenden eingereichten Webauftritten nicht von den Studierenden selbst erstellt, sondern von Projekten der Mitstudierenden kopiert wurden.

Da die manuelle Suche nach Plagiaten in einer Menge von 500-600 Webauftritten zeitlich nicht umsetzbar ist, lag die Motivation für einen Lösungsansatz der Arbeit darin, die Überprüfung der Prüfungsleistungen der Studierenden hinsichtlich potenzieller Plagiate automatisiert zu unterstützen. Eine ideale Lösung für den Anwendungsfall für die Untersuchung der Webauftritte wäre somit ein Ergebnis in Form eines prozentualen Ähnlichkeitswerts, welcher einen Hinweis auf potenzielle Plagiate gibt und damit die Dozierenden der Vorlesung im Prüfungsprozess unterstützt.

3. Vorgehensweise

Der Arbeit wurden insgesamt 21 studentische Webauftritte zur Verfügung gestellt. Neben der theoretischen Einführung in die Plagiatserkennung stellte die Analyse der Projekte hinsichtlich der Vorgehensweise bei der Verschleierung von potenziellen Plagiaten die Grundlage für die Evaluation verschiedener Herangehensweisen zur Plagiatserkennung dar.

Theorie

Für den theoretischen Teil der Arbeit wurden zu Beginn die Grundlagen der textbasierten Plagiatserkennung sowie die Unterschiede der Plagiatserkennung in Quellcode herausgearbeitet. Anschließend wurden die Funktionsweisen der zu untersuchenden Tools, **DaisyDiff**, **JsDiff** und **JPlag** näher beschrieben sowie deren Algorithmen erklärt.

Alle ausgewählten Tools verfolgten den strukturorientierten Ansatz der Plagiatserkennung, d.h. es werden entweder Zeichenketten oder tokenisierte Segmente des Quellcodes verglichen. Die Dateien werden dabei in eine Token-Zeichenkette umgewandelt. Die Token stellen die lexikalischen Einheiten dar, welche die Bausteine des Programms repräsentieren. Die entstehende Kette aus Token bildet das Programm ab und kann nachfolgend mit einer anderen Tokenkette verglichen werden (Prechtelt, Malpohl, & Phlippsen, 2000).

Praxis

Im praktischen Teil der Arbeit wurden zunächst die Webauftritte der Studierenden hinsichtlich bestimmter Kriterien analysiert, woraus konstruierte Testbeispiele abgeleitet wurden. Aus der Analyse entstand demnach ein Kriterienkatalog, anhand welchem jeweils 8 Testbeispiele für HTML- und CSS-Dateien herausgearbeitet wurden. Mittels der konstruierten Testbeispiele konnten anschließend die ausgewählten Tools hinsichtlich ihrer Funktionsweise und in Bezug auf den Anwendungsfall der Webauftritte der Studierenden getestet werden.

Jedes Tool wurde dabei individuell untersucht und bezüglich vorher festgelegter Kriterien bewertet. Neben der Korrektheit der Testdurchführung spielten auch Kriterien wie Benutzerfreundlichkeit, Zuverlässigkeit und Erweiterbarkeit eine wichtige Rolle für den zu untersuchenden Anwendungsfall, weshalb sie in die Kriterienliste für die Bewertung mit aufgenommen wurden.

Die Erkenntnisse und Ergebnisse des praktischen Teils führten zum Schluss der Arbeit zur Erstellung eines Entwurfs für einen Ansatz zur Plagiatserkennung von dupliziertem Quellcode.

4. Ergebnisse

DaisyDiff

Das Tool DaisyDiff wurde in der ersten Version im Jahr 2008 entwickelt. Es vergleicht zwei HTML-Dateien miteinander und zeigt deren Unterschiede auf. Dabei wird die Baumstruktur der HTML-Syntax aufeinander abgebildet. Das Ergebnis liefert dabei eine zusätzliche HTML-Datei zurück, welche mithilfe einer bestimmten Syntax die Änderung kennzeichnet. Es werden bei jeder Änderung Span-Tags eingefügt, die die jeweilige Änderung hinsichtlich ihrer Funktion spezifizieren. Diese eingefügten Span-Tags wurden gezählt, um einen Hinweis auf potenzielle Plagiate zu erhalten.

Abbildung 1 Testdurchlaufs mit DaisyDiff – grafische Darstellung im Browser und im Quellcode

Zusammenfassung der Ergebnisse:

- eingefügte Span-Tags geben keine Hinweise auf mögliche Plagiate
- Sehr unübersichtliche und nicht nachvollziehbare Darstellung der Ergebnisse

JsDiff

Das npm-package JsDiff wird als JavaScript-Textdifferenzierungs-Implementierung beschrieben. Für den Anwendungsfall der zu vergleichenden Webauftritte war speziell die Methode diffCss des JsDiff-Pakets relevant. Mittels Longest-Common-Subsequence-Algorithmus werden zwei CSS-Dateien verglichen, deren Unterschiede erkannt und gekennzeichnet. Alle geänderten Objekte werden angezeigt und zusammengezählt.

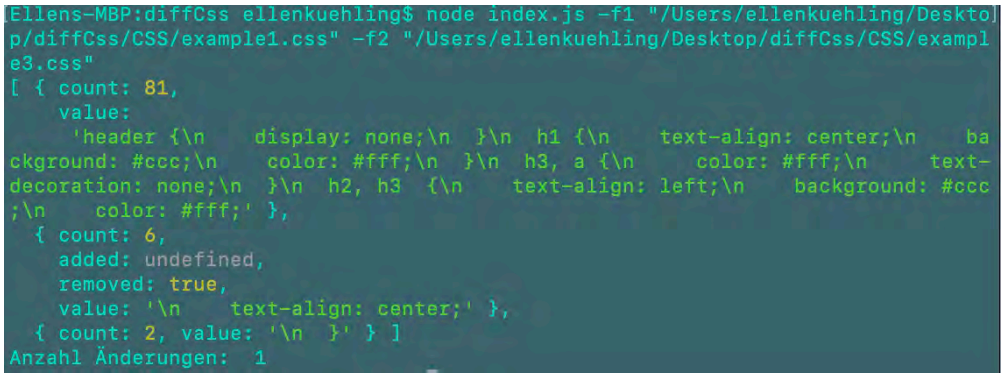

Abbildung 2 Ergebnis des Testdurchlaufs mit JsDiff

Zusammenfassung der Ergebnisse:

- Die Anzahl der Änderungen der zu vergleichenden Dateien gibt prinzipiell einen Hinweis auf potenzielle Plagiate, da die Gesamtanzahl der Änderungen bei ähnlichen Projekten gering ist.

- Größere sich unterscheidende Abschnitte werden jedoch lediglich als **eine** Änderung gezählt, wodurch das Ergebnis des Vergleichs verfälscht wird.

JPlag

JPlag wurde im Jahr 1996 als studentisches Forschungsprojekt an der Universität Karlsruhe gestartet. Die Entwicklung des Systems erfolgte dabei speziell für den Anwendungsfall studentischer Programmierabgaben der Sprache Java (Hage, Rademaker, & van Vugt, 2010). Für die Arbeit wurden Anpassungen durchgeführt, um die für den Vergleich relevanten Sprachen HTML und CSS testen zu können.

JPlag konvertiert die eingegebenen Dateien und Programme in Token-Strings, welche dann die Struktur des Programms repräsentieren. Dadurch wird jede Datei in eine Zeichenkette aus kanonischen Token umgewandelt, welche mittels dem Greedy-String-Tilling-Algorithmus verglichen werden (Hage, Rademaker, & van Vugt, 2010). Die Bildung der Token musste individuell festgelegt werden. Hierbei wurden in der Arbeit zwei unterschiedliche Ansätze verfolgt: 1) Vergleich der Token aus Eigenschaftsnamen, 2) Vergleich der Token aus Id-und Klassenbezeichnungen.

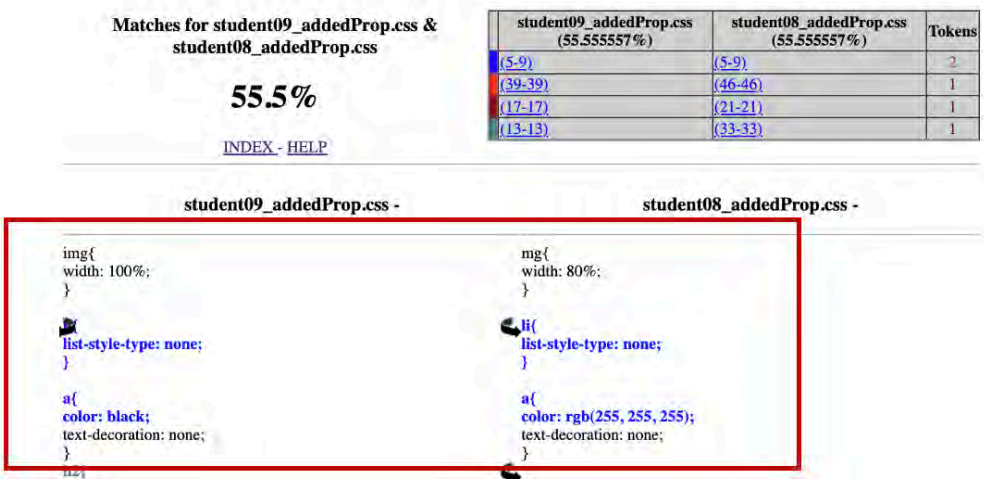

Abbildung 3 Ergebnis des Testdurchlaufs mit JPlag – Prozentualer Ähnlichkeitswert

Zusammenfassung der Ergebnisse:

- JPlag ermöglicht den Vergleich einer größeren Anzahl von Dateien
- Das Ergebnis wird wie gewünscht als prozentualer Ähnlichkeitswert ausgegeben

Ergebnisse

Tabelle 1 Bewertung der Tools

Kriterium	DaisyDiff	JsDiff	JPlag
Korrektheit			
Zuverlässigkeit			
Benutzerfreundlichkeit			
Erweiterbarkeit			

 Gut / Vorhanden Ausreichend Mangelhaft / Nicht vorhanden

Vorstellung des Entwurfs

Nach der Auswertung der Testdurchführung wurde ein Lösungsvorschlag beschrieben, der Dozierende der Veranstaltung Web-Technologien im tatsächlichen Anwendungsfall gewinnbringend unterstützen sollte. Das Tool JPlag brachte den höchsten Nutzen für das Vergleichen der Dateien, da es durch die individuelle Festlegung der Token-Bildung gut einsetzbar ist und bezüglich der Korrektheit sehr gute Ergebnisse geliefert hat. Aus den beschriebenen Gründen wurde der Anwendungsfall des Vergleichens einer größeren Anzahl von Webauftritten mittels JPlag getestet und analysiert.

Dabei sind folgende Einschränkungen aufgetreten:

- Hohe Ähnlichkeitswerte bei zu kurzen Dateien
- Schema der Benennung der Klassen- und Ids wird oftmals aus den Übungen der Veranstaltung übernommen
- Ungeordnete Reihenfolge der Ähnlichkeitswerte
- Unübersichtliche Darstellung des Quellcodes bei mehreren Dateien innerhalb eines Ordners

Um den aufgeführten Einschränkungen entgegenzuwirken, wurden für den Entwurf für einen Ansatz für die Plagiatserkennung der Webauftritte Lösungsansätze herausgearbeitet, deren Einsatz eine bessere und benutzerfreundlichere Verwendung des Tools ermöglicht. Die Ansätze sehen vor, dass die zu untersuchenden Dateien gefiltert werden, wodurch zu kurze Dateien vom Vergleich ausgenommen werden können. Zusätzlich sollen Standardbezeichnungen und Standardformatierungen nicht als Token gespeichert werden. Die Darstellung des Ergebnisses des Vergleichs soll in geordneter Reihenfolge erfolgen. Schließlich soll es für die Dozierenden die Möglichkeit geben, die untersuchten Dateien direkt einsehen zu können.

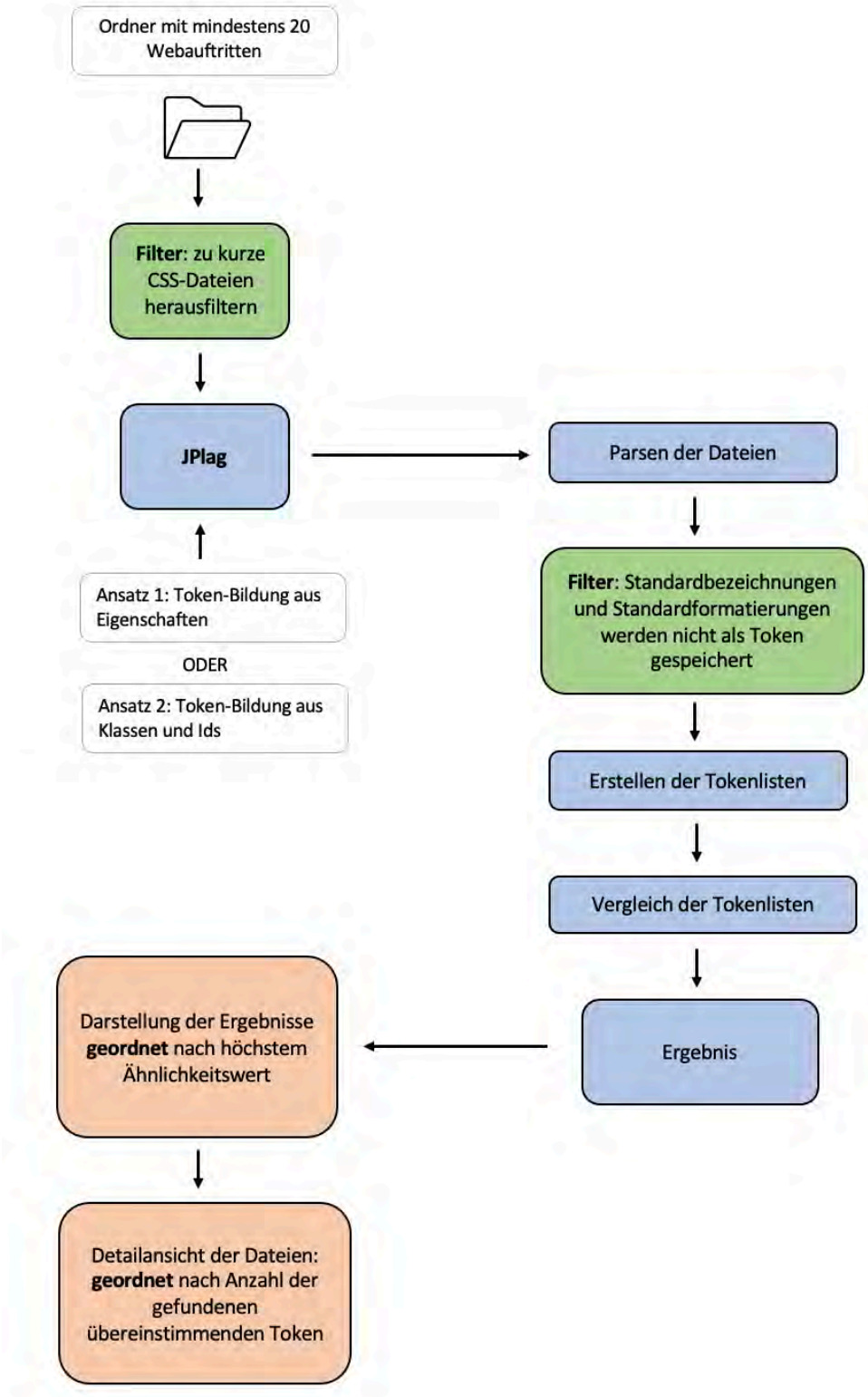

Abbildung 4 Entwurf für einen Ansatz für die Plagiatserkennung der Webauftritte

5. Zusammenfassung und Fazit

Die Durchführung der Tests hat ergeben, dass prinzipiell alle in der Arbeit untersuchten Tools Hinweise auf potenzielle Plagiate geben können. DaisyDiff landete dabei in der Bewertung auf dem letzten Platz, da nicht alle Ähnlichkeiten erkannt werden und die Benutzung nicht intuitiv geschieht. JsDiff zeigt die Änderungen der zu vergleichenden Dateien an, wodurch man lediglich einen Hinweis über die Anzahl der Änderungen bekommt. Aus diesen Gründen sind beide Tools für die Verwendung der Überprüfung der Webauftritte im Rahmen der Veranstaltung Web-Technologien nicht relevant.

JPlag hingegen ist durch seine Erweiterbarkeit und die Möglichkeit der individuellen Festlegung der Regeln für die Token-Bildung gut einsetzbar. Das Tool unterstützt die Überprüfung und gibt Hinweise auf potenzielle Plagiate. Zudem bietet es die Möglichkeit der Anpassung auf spezielle Aufgabenstellungen. Die Rückgabe des Ergebnisses zeigt eine Übersicht aller untersuchten Ordnern und Dateien an. Daher kann JPlag unter den vorgestellten Bedingungen unterstützend zur automatisierten Überprüfung der Webauftritte eingesetzt werden.

In jedem Fall müssen verdächtige Projekte zusätzlich vom Dozierenden überprüft werden, da ein Tool nicht in der Lage ist, Plagiate zu beweisen. Das Tool kann lediglich Ähnlichkeiten feststellen und den Prüfungsprozess automatisiert unterstützen.

6. Referenzen

[1] Hage, J., Rademaker, P., & van Vugt, N. (2010). *A comparison of plagiarism detection tools.* Utrecht: Utrecht University.

[2] Prechelt, L., Malpohl, G., & Phlippsen, M. (2000). *JPlag: Finding plagiarisms among a set of programs.* Karlsruhe: Universität Karlsruhe.