LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Plagiarism Detection Approaches for Simple Introductory Programming Assignments

Sven Strickroth

ABP Workshop 2021

2021-10-29

Motivation

- Programmieren lernen wird oft als große Hürde empfunden
- Nicht alle Studierenden lösen die (Haus-)Aufgaben selbst oder reichen Plagiate ein
- Plagiate und andere Formen des Schummelns werden oft als große Probleme genannt (Luxton-Reilly et al., 2018)

- Plagiate müssen schon von Anfang an bekämpft werden

- In „großen“ Kursen benötigt man Software-Support
  - Wie auswählen?
  - Welche Schnittstellen werden angeboten?
  - Eignen sich die Tools auch für kleinere Aufgaben?

State-of-the-Art

- Systematischer Literatur-Review von Novak et al., 2019
  - Definitionen von (akademischen) Plagiaten
  - eingesetzte Methoden/Tools
  - Verschleierungsmethoden
  - Evaluationsansätze für Methoden und benutzte Datensätze
  - TOP-5 Tools: JPlag, MOSS, Sherlock-Warwick, Plaggie, SIM-Grune
  - viele Tools nicht verfügbar

- Häufig „nur“ Vergleich der Robustheit, Performanz oder Features
- Erwähnenswert ist Modiba et al., 2016
  - Vergleich von False-Positives und False-Negatives auf realem Datensatz
  - Aber: keine klare Definition/Darstellung der Methode und Plagiate

- Forschungslücke: Vergleich verschiedener Ansätze auf echten Daten

Existierende Ansätze

Tool	Release	Algorithmus	Java-Unterstützung	Quellcode-Verfügbarkeit	Offline
JPlag	2.12.1 (2019)	Greedy String Tiling (token-based)	ja (<= 9)	ja, GPLv3	ja
Levensthein Distanz	(GATE)	Levensthein Distanz	beliebige Text-Dateien	ja, GPL	ja
Moss	n/a	Winnowing (token-based)	ja, auch ungültige Syntax	nein	nein
Plaggie	2006	Greedy String Tiling (token-based)	ja (<= 6), nur syntaktisch korrekt	ja, GPL	ja
Sherlock-Sydney	n/a	Winnowing (token-based)	beliebige Text-Dateien	ja, public domain	ja
SIM	3.0.2 (2017)	token-based	ja, auch ungültige Syntax	ja, BSD	ja

Features & Schnittstellen

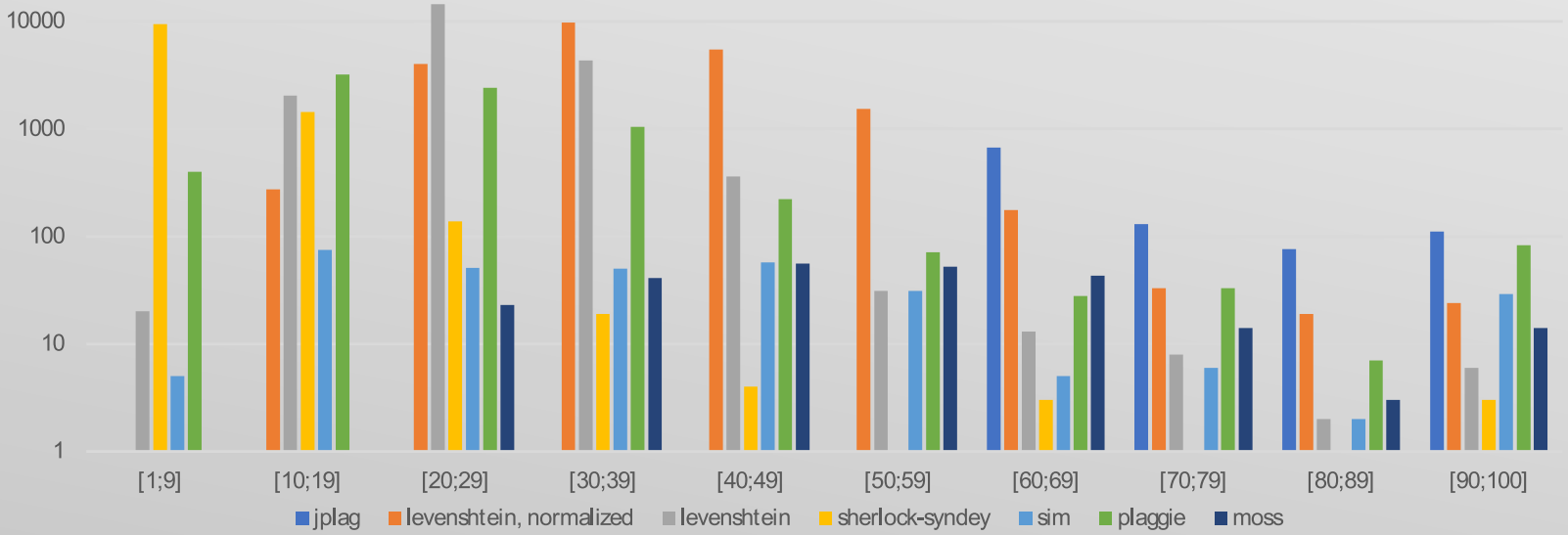
Tool	CLI (Sprache)	Java API	CSV	HTML Report	STDOUT	Standard-Threshold	Code-Vorlagen
JPlag	ja (Java)	ja	ja	ja	(ja)	max. 1000 Ähnlichkeiten in 20 Clustern	ja
Levensthein Distanz	(nein)	ja	ja	nein	(nein)	-	nein
Moss	ja	(ja)		ja	nein	250 Ähnlichkeiten	ja (automatisch)
Plaggie	Ja (Java)	ja	ja	ja	nein	>= 50% Ähnlichkeit	ja
Sherlock-Sydney	Ja (C)	nein	ja	nein	ja	>= 50% Ähnlichkeit	nein
SIM	ja (C, „nur“ Windows)	nein	ja	nein	ja	-	nein

Methode

- Untersuchung von zwei Aufgaben:
  - Berechnungsaufgabe (LoC: ~25, main Methode, Variablen, Berechnungen, 209 Abgaben, 60 bekannte Plagiate mit 63 Paaren): Berechnung der Zinsen eines Ratenkredits und wann es einen Break-Even-Punkt gibt. Eingabe über args[].
  - main-Methode mit einem “if else-if else”-Konstrukt oder verschachteltem if-Anweisungen zur Unterscheidung 3 Fälle und Rückgabe vorgegebener Strings basierend auf “arg[0]”. LoC: ~20, 616 Abgaben).
- Suche nach Plagiaten mit allen Tools mit Standardparametern (außer, dass alle möglichen Plagiate ausgegeben werden sollten >= 1% Ähnlichkeit)
- Vergleich der Anzahl der gefundenen möglichen Plagiate und des Inter-Rater-Agreements (bei >= 80% Ähnlichkeiten)

Ergebnisse: Aufgabe 1

Tool	No. results	No. def. thr.	No. ≥ 80 %	Max %	Mean %	Median %
JPlag	1000 (63)	40 (26)	190 (59)	100 (100)	3.2 (93.8)	0 (100)
LV	21528 (63)	21528 (63)	8 (8)	100 (100)	25.8 (52.3)	0 (46)
LVN	21528 (63)	21528 (63)	43 (36)	100 (100)	36.4 (81.7)	0 (65)
Moss	250 (63)	250 (63)	17 (14)	99 (99)	0.6 (62)	0 (58)
Plaggie	7647 (63)	227 (53)	90 (46)	100 (100)	8 (80.6)	0 (100)
Sherlock	11217 (63)	174 (24)	4 (4)	100 (100)	2.9 (22.3)	0 (17)
SIM	317 (63)	317 (63)	31 (27)	100 (100)	0.6 (51.2)	0 (99)



Ergebnisse: Aufgabe 2

Tool	No. results	No. def. thr.	No. ≥ 80 %	Max %	Mean %	Median %
JPlag	1000	288	1000	100	0.5	0
LV	189420	189420	31355	100	61.9	2
LVN	189420	189420	79165	100	73	3
Moss	250	250	15	99	0.1	0
Plaggie	99326	75431	52209	100	38.5	0
Sherlock	164968	81883	1051	100	19.4	0
SIM	978	978	290	100	0.3	0

Inter-Rater Übereinstimmung

	Plaggie	Moss	SIM	Sherlock	LVN	LV	Human
JPlag	0.641	0.144	0.279	0.041	0.358	0.080	0.464
Plaggie		0.242	0.429	0.063	0.480	0.142	0.600
Moss			0.583	0.190	0.366	0.400	0.350
SIM				0.228	0.594	0.410	0.574
Sherlock					0.170	0.667	0.119
LVN						0.274	0.678
LV							0.225

Tab. 4: Inter-rater agreement on assignment 1 for ≥ 80 % classification

	Plaggie	MOSS	SIM	Sherlock	LVN	LV
JPlag	0.027	0.000	0.024	0.019	0.015	0.017
Plaggie		0.000	0.007	0.016	0.681	0.472
Moss			0.039	0.000	0.000	0.001
SIM				0.022	0.004	0.008
Sherlock					0.011	0.024
LVN						0.431

Diskussion & Zusammenfassung

- Vergleich und Vorstellung verfügbarer Erkennungstools

- Generelle Probleme (Ground-Truth):
  - Kennt man wirklich alle Plagiate?
  - Wann ist ein gemeldetes Plagiat wirklich ein Plagiat?
- Aufgabe 1: fast alle Tools haben alle bekannten Plagiate gefunden, aber
  - Durchschnittsähnlichkeit dabei variierte stark
  - alle Tools meldeten max. Ähnlichkeit mit ~100%

    → kein einfaches Skalierungsproblem
- Aufgabe 2: viel zu viele gemeldete Ähnlichkeiten, auch bei Levenshtein!
- Inter-Rater-Agreement nicht wirklich vorhanden
- Weitere Untersuchungen notwendig: ggf. Kombinationen von Tools oder Berücksichtigung der Historie oder Fehlern besser geeignet

- Allgemein: Tools sind nur ein Ansatz, ggf. besser Studierende im Zweifel zu eigenen Lösungen befragen (nach Verständnis)

Prof. Dr. Sven Strickroth

Ludwig-Maximilians-Universität München
Institut für Informatik
Lehr- und Forschungseinheit für
Programmier- und Modellierungssprachen
Oettingenstraße 67
80538 München

Telefon: +49-89-2180-9300
sven.strickroth@ifi.lmu.de
https://www.tel.ifi.lmu.de