JPlag

Agathe Rzymkowski, Daniela Hammer

- verfügbar unter http://wwwipd.uka.de/jplag
- kostenfrei, Login über Account
- entwickelt von Guido Malpohl (1996)
- unterstützte Programmiersprachen: Java, C, C++, C# und Scheme
- ursprünglich für den Code-Vergleich entwickelt, kann inzwischen jedoch auch Textdateien vergleichen
- vergleicht nicht mit Dokumenten im Internet, sondern sucht nach Ähnlichkeiten in ausgewählten Texdokumenten untereinander
- liefert als Ausgabe einen Satz von HTML-Seiten
- grafische Darstellung der Ergebnisse
- richtet sich nach dem “Greedy String Tiling”-Algorithmus

Wie JPlag die Ähnlichkeit von zwei Programmen/Texten berechnet

JPlag versucht die maximalen nicht-überlappenden Token-Sequenzen zu finden. So können auch Duplikate gefunden werden, in denen Reihenfolgen von Methoden getauscht und/oder zusätzliche Aufrufe eingefügt wurden. Es verwendet aber noch weitere spezielle Token, welche die Semantik widerspiegeln (z. B. BEGINMETHOD).

JPlag arbeitet in zwei Phasen:

1. Alle zu vergleichenden Programm/Text-Paare werden geparst (oder gescannt, abhängig von der Eingangs-Sprache) und in Token-Strings konvertiert.

	Java source code	**Generated tokens**
1	`public class Count {`	BEGINCLASS
2	`public static void main(String[] args)`	VARDEF,BEGINMETHOD
3	`throws java.io.IOException {`	
4	`int count = 0;`	VARDEF,ASSIGN
5		
6	`while (System.in.read() != -1)`	APPLY,BEGINWHILE
7	`count++;`	ASSIGN,ENDWHILE
8	`System.out.println(count+" chars.");`	APPLY
9	`}`	ENDMETHOD
10	`}`	ENDCLASS

2. Diese Zeichenketten-Paare werden miteinander verglichen. Daraufhin wird die Ähnlichkeit jedes Paars bestimmt. Bei jedem dieser Vergleiche versucht JPlag, ein Token-Stream mit Teilstrings ("Kacheln") des anderen Programms gut wie möglich abzudecken. Der prozentuale Anteil der Token-Streams, die abgedeckt werden können, ist der Ähnlichkeitswert. Die entsprechenden Teilstrings werden durch die Schnittstelle visualisiert und beschrieben.
JPlag verwendet hierfür grundlegend den „Greedy String Tiling“-Algorithmus.