

Bakkalaureatsarbeit im Rahmen der
Lehrveranstaltung 265 069
Projektpraktikum über

Plagiat – Menschenraub in der
Wissenschaft

Helga Vorfa (0227190) Martin Weik (0026006)

17.11.2008

Inhaltsverzeichnis

1	Einleitung	4
2	Plagiat – Menschenraub in der Wissenschaft	6
2.1	Wortherkunft	6
2.2	Geschichte	6
2.3	Plagiat – Eine Begriffsdefinition	7
2.4	Plagiatsformen	10
2.4.1	Totalplagiat	10
2.4.2	Teilplagiat	10
2.4.3	Autoplagiat	11
2.4.4	Übersetzungsplagiat	11
2.4.5	Ideenplagiat	12
2.4.6	Ghostwriting	12
3	Entstehung und Erkennung von Plagiaten	14
3.1	Entstehung	14
3.2	Erkennung	15
3.2.1	Erkennung durch Programme	15
3.3	Plagiaten vorbeugen	19
4	Problem „Plagiat“ in Österreich	21
4.1	Situation in Österreich	21
4.2	Rechtslage in Österreich	24
4.2.1	Urheberrechtsgesetz	24
4.2.2	Geschützte Werke	28
4.3	Universitätsgesetz	30
5	Plagiat in der Wirtschaft	33
5.1	Produktpiraterie	36
6	technischer Teil	38
6.1	Einleitung	38
6.1.1	LVA Beschreibung	38

Inhaltsverzeichnis

6.1.2	Aufgabenstellung	38
6.1.3	Ansatz	38
6.2	Projektstart	39
6.2.1	Vorbedingungen	39
6.2.2	aspHeute	39
6.3	Neuimplementierung in Java: YAPLAF	41
6.3.1	Texte einlesen und vorbereiten	42
6.3.2	Abgaben verarbeiten und Common Strings suchen	44
6.3.3	Definition von „MatchingString“	46
6.3.4	die Datenstruktur MatchingStringContainer	48
6.3.5	Ergebnisse ausgeben und abspeichern	50
6.4	Kritik	51
6.4.1	Arten von Kopien	51
6.4.2	Testfälle	52
6.5	Ausblick	56

1 Einleitung

Obwohl im universitären Bereich immer mehr auf Plagiate und deren Folgen aufmerksam gemacht wird, schrecken Viele nicht davor zurück, weiterhin zu plagieren. Unter Plagieren fallen nicht nur das Wort-wörtliche Abschreiben von fremden Arbeiten, sondern auch das sinngemäße Abschreiben und das Abschreiben von eigenen Arbeiten.

Kapitel zwei gibt einen Überblick über den Ursprung des Begriffs Plagiat, die fünf bekanntesten Plagiatsformen und das Problem Ghostwriting, welches als Sonderfall des Plagiats betrachtet werden kann.

Das Thema Plagiats-Software wird in Kapitel drei behandelt. Unter „Plagiats-Software“ versteht man Programme, die eingesetzt werden um Plagiaten auf die Spur zu kommen, indem Texte auf Ähnlichkeit mit anderen Texten überprüft werden. Aus Studien der Professorin Deborah Weber Wulf¹ hat sich herausgestellt, dass nicht jede Plagiats-Software, die momentan verwendet wird um Plagiate aufzuspüren, auch gut dafür geeignet ist. Das Problem liegt teilweise an der Tatsache, dass Plagiate immer schwieriger zu erkennen sind, aber auch an Mängeln der Software.

Die Rechtslage in Österreich wird in Kapitel vier beschrieben. Das Urheberrechtsgesetz stellt klare Anforderungen an die Verwendung von fremdem geistigem Eigentum für eigene Zwecke. Durch ausreichende Kenntnisse über richtiges Zitieren könnte die Anzahl der als Plagiate aufgedeckten Arbeiten minimiert werden.

Auch im wirtschaftlichen Bereich haben sich Produktpiraterie und Fälschung weit verbreitet. Diese etwas andere Form des Plagierens hat in den letzten Jahren sehr viel Schaden verursacht. Kapitel fünf gibt einen Überblick über das Ausmaß der Verbreitung von Produktfälschungen sowie über Produkte, die ein beliebtes Ziel von Fälschern sind, und der Risiken, die Fälschungen mit sich bringen.

Im Zuge dieser Arbeit wurde ein Programm erstellt, mit dem sogenannte Collusionstests², also Vergleiche von Texten innerhalb eines Sets, durchgeführt werden können und das LVA Leitern das Prüfen auf vermutlich kopierte Abgaben erleichtern soll. Damit soll auch bei den Studenten / Studentinnen präventiv ein Bewusstsein dafür geschaffen werden, dass auch das Abschreiben „kurzer“ Texte nach wie vor als Abschreiben zu bewerten ist. Die Erkenntnisse daraus sind in Kapitel sechs zusammengefasst.

¹http://plagiat.fhtw-berlin.de/ff/startseite/fremde_federn_finden

²<http://plagiat.fhtw-berlin.de/software/2008/collusion/>, Stand vom 17.11.2008

1 Einleitung

Quellenangaben

Alle Websites wurden zuletzt am 17.11.2008 besucht. Die Angaben im Literaturverzeichnis beziehen sich auf die letzte Änderung, soweit vorhanden.

Alle Gesetztexte stammen aus dem Rechtinformationssystem des österreichischen Bundeskanzleramts [Öst] und entsprechen dem aktuellen Stand.

2 Plagiat – Menschenraub in der Wissenschaft

2.1 Wortherkunft

Der Begriff „Plagiat“ wird vom lateinischen „plagium“ abgeleitet und bedeutet ursprünglich „Menschenraub“. Die erstmalige Verwendung ist im ersten Jahrhundert nach Christus belegt: Als der Schriftsteller Fidentius von seinem Kollegen Martial abgeschrieben hat, verglich dieser das mit Kindesentführung und prägte damit diesen Begriff. [Wul07, S. 2]

2.2 Geschichte

„Die Welt will betrogen sein“

Sebastian Brant [Bra04]

Bis in das 19. Jahrhundert war es üblich, fremde Werke als Eigene auszugeben. Die Urheberschaft als Beschreibung von geistigem Eigentum hatte sich, im Unterschied zum materiellen Eigentum, noch nicht entwickelt. Ein Buch zu stehlen war somit verboten, ein Buch abzuschreiben war jedoch erlaubt.

Man kann behaupten, dass es in der Natur des Menschen liegt, alles Neue nicht unmittelbar nach dessen Erscheinen zu akzeptieren. Uns fällt es leichter, aus dem Vorhanden und Vertrauten zu schöpfen. Dem zufolge können wir uns als perfekte Plagiatoren bezeichnen.

Man braucht nur an Sprichwörter oder Witze zu denken. Diese haben sich schnell verbreitet und haben Jahrzehnte, sogar Jahrhunderte, dank mündlicher Überlieferung überlebt. Der Schriftsteller Goethe war der Meinung, man solle das aufnehmen, was andere bereits gesagt oder gedacht haben.

„Ich verdanke meine Werke Tausenden von Dingen und Personen außer mir, die mir dazu das Material boten. Es kamen Narren und Weise, helle Köpfe und bornierte, Kindheit und Jugend wie das reife Alter: alle sagten mir, wie es ihnen zu Sinn sei, was sie dachten, was sie lebten und wirkten und welche Erfahrungen sie sich gesammelt, und ich hatte weiter nichts zu tun als zuzugreifen und das zu ernten, was andere für mich gesät hatten.“ [Kar01, S. 2]

In der Zeit des Barocks war es in der Musik üblich, durch das neu Gestalten von alten, vorhanden Werken neue Werke zu schaffen.

Der Autor Karl May hat seine Reiseerlebnisse, die er in seinen Büchern beschreibt, nicht selbst erlebt. Alle diese Ereignisse sind aus verschiedenen Büchern, die sich mit der Thematik befasst haben, abgeschrieben.

Carlos Castaneda, ein anderer Schriftsteller, erntete Lob und Unterstützung von angesehenen Wissenschaftlern und Professoren wie Herold Garfinkel für seine Bücher. Die angeblichen Erlebnisse des Schriftstellers waren jedoch eine Ausarbeitung anderer Bücher und Zeitschriften. Ein paar skeptische Wissenschaftler fanden damals heraus, dass Castaneda viele Fehler unterlaufen waren. Beispielsweise war das fehlerhafte Übersetzen der mexikanischen Ortsnamen ins Englische ein Hinweis darauf, dass der Autor seine Geschichten nicht selbst erlebt hatte. [EKW03, S. 78–79]

Die Lage im wissenschaftlichen Bereich war nicht bedeutend anders; auch hier wurde auf vorhandenes Wissen aufgebaut. Newton soll gesagt haben:

Wenn ich weiter als andere gesehen habe, dann nur deshalb, weil ich auf der Schulter von Giganten stand.[Kar01, S. 2]

Dabei wurde oft auf Quellenangaben „vergessen“, was gerade in Zeiten des Internets enorme Probleme aufwerfen kann.

Über den „Urheber“ zum „Plagiat“

Erst Ende des 19. Jahrhunderts, mit der Entstehung eines klaren Begriffs des Urhebers, konnte man davon ausgehend den Begriff „Plagiat“ definieren. Ab diesem Zeitpunkt war klar, dass nicht Alles allen gehört. Der Urheber ist derjenige, der ein Werk geschaffen hat und somit er als Einziger alle Rechte über sein Werk besitzt.

2.3 Plagiat – Eine Begriffsdefinition

Der Begriff „Plagiat“ lässt sich nicht leicht definieren. Es existieren sehr viele Definitionen, welche sich oft widersprechen und auch Fragen offen lassen.

Stefan Schaltenbrand gibt eine humoristische Definition von Plagiat [Wul07, S. 3]:

*„Aus einem Buch abschreiben = Plagiat;
aus zwei Büchern abschreiben = Essay;
aus drei = Kompilation;
aus vier = Dissertation“.*

Eine anschauliche Definition ist in der Seminararbeit „Multimedia und Electronic Publishing“ von Karsten Kutschera zu finden [Kut97, S. 14]:

„ Als Plagiat bezeichnet man allgemein die bewusste Aneignung fremden Geistesgutes. Plagiator ist derjenige, der ein fremdes Werk oder Teile eines fremden Werkes als sein eigenes Werk ausgibt und somit „geistigen Diebstahl“ begeht. Der Plagiator begeht eine zivilrechtlich unerlaubte und zum Schadenersatz an den Autor verpflichtende Handlung, die gleichzeitig auch noch strafbar ist. Der Plagiator ist also derjenige, der seinen Text wörtlich bei einem anderen Urheber abschreibt, ohne ihn korrekterweise zu zitieren, um anschließend dieses Werk als sein eigenes geistiges Eigentum wieder herauszugeben“

Universität und Fachhochschulen

Mittlerweile ist das Phänomen Plagiat im universitären Leben sehr verbreitet. Aufgrund des Fehlens einer allgemein geltenden Regelung haben Universitäten und auch andere wissenschaftliche Einrichtungen oft eine eigene Definition von Plagiat. Hier ein paar Beispiele:

- Technische Universität Wien–Industrial Software Institut
Das Institut INSO (Industrial Software) hat eine Definition von Plagiat und auch klare Konsequenzen im Falle eines Plagiats festgelegt. Diese werden Studenten auf der Institutsseite bekannt gemacht [is08]:

„ Das simple Abschreiben von Textpassagen ohne Quellenangabe wird als Plagiat bezeichnet. Darunter fallen:

- Kopieren von Textpassagen aus Büchern, Zeitschriften, Webseiten, von Kollegen ohne entsprechende Quellenangabe.*
- Kopieren von Programmcode, z.B. von früheren Abgaben.*

“

Und zu den Konsequenzen des Plagiiere[n]s [is08]

„ Handelt es sich bei einer Arbeit um ein Plagiat, so muss eine neue Arbeit verfasst werden. Wird ein Plagiat nach der Beurteilung aufgedeckt so muss die Arbeit laut § 74 Abs.2 Universitätsgesetz für nichtig erklärt werden.“

- Universität Wien [Wie04]

„ Ein Plagiat ist die bewusste und unrechtmäßige Übernahme von fremdem geistigem Eigentum; der/die AutorIn verwendet – ganz oder teilweise – fremde Werke in einem eigenen Werk, ohne die Quelle anzugeben.“

Konsequenzen des Plagiiere[n]s:

2 Plagiat – Menschenraub in der Wissenschaft

„Neben möglichen Schadenersatzforderungen des Urhebers bzw. der Urheberin gem. § 87 UrhG und strafrechtlichen Konsequenzen – bestimmte vorsätzliche Urheberrechtsverletzungen können gem. § 91 UrhG mit Freiheitsstrafe bis zu sechs Monaten oder Geldstrafe bis zu 360 Tagessätzen bestraft werden – kommen vor allem universitätsrechtliche Sanktionen in Betracht:

- Wird das Plagiat noch vor der Beurteilung der wissenschaftlichen Arbeit entdeckt, muss der/die Studierende gem. § 18 Abs 3 des studienrechtlichen Teils der Satzung der Universität Wien eine neue Arbeit verfassen.*
- Wird das Plagiat erst nach der Beurteilung entdeckt, ist ein Verfahren nach § 74 Abs 2 UG 2002 (Nichtigerklärung von Beurteilungen) und gegebenenfalls auch nach § 89 UG 2002 (Widerruf erschlichener akademischer Grade) einzuleiten.*

“

- **Wirtschaftsuniversität Wien [DP08]**

„Als Plagiat wird üblicherweise die bewusste unrechtmäßige Übernahme von fremdem geistigem Eigentum bezeichnet.

Ein Plagiat liegt vor wenn:

- 1. die Arbeit Anderer für die Eigene ausgibt*
- 2. Arbeiten (oder Teile davon) aus dem Internet herunterlädt und sie dann für die eigene ausgibt*
- 3. fremdsprachige Arbeiten (oder Teile davon) übersetzt und ohne Quellenangaben übernimmt*
- 4. Zitate verwendet, ohne die entsprechenden Quellen zu nennen.*
- 5. ein und dieselbe Arbeit (oder Teile davon) in verschiedenen Seminaren abgibt/einreicht.*

“

- Fachhochschulen [Fac08]

„ Wenn es sich bei einer Diplomarbeit nachweislich um ein Plagiat oder ein teilweises Plagiat einer anderen wissenschaftlichen Arbeit handelt, so ist gem. Akkreditierungsrichtlinien des FHR die Approbation der Diplomarbeit für ungültig zu erklären (vgl. AR 2005, II.E.2.c.). Da die einen Diplomstudiengang abschließende Diplomprüfung eine Gesamtprüfung ist, die sich aus der Anfertigung einer Diplomarbeit und der Ablegung einer kommissionellen Prüfung zusammensetzt, wäre daher auch die Diplomprüfung für ungültig zu erklären und der verliehene akademische Grad durch das für die Verleihung des akademischen Grades zuständige Organ abzuerkennen. Die Wiederholung der Diplomprüfung, deren Beurteilung für ungültig erklärt wurde, ist auf die Gesamtzahl der Wiederholungen anzurechnen; im Rahmen der ursprünglichen Diplomprüfung bereits erbrachte und gültige einzelne Prüfungsleistungen könnten anlässlich der Wiederholung der Diplomprüfung zutreffendenfalls Berücksichtigung finden.“

2.4 Plagiatsformen

Plagiate lassen sich in fünf Gruppen aufteilen. Jede davon hat bestimmte Eigenschaften, die sie von anderen Gruppen unterscheidet.

2.4.1 Totalplagiat

Die Wort-wörtliche Übernahme von Texten anderer Autoren ohne Quellenangabe wird als Totalplagiat bezeichnet. Der Plagiatör gibt fremde Texte als eigene aus. Diese Plagiate werden in der Regel sehr leicht gefunden.

Der Iraker A.K. Alsabti ist der bekannteste Fall davon. [EKW03, S.82] Er hat eine medizinische Prüfungsurkunde gefälscht, eine Verwandtschaft mit der königlichen Familie Jordaniens erfunden, sich in Amerikanischen Universitäten hineingeschwindelt und sich selbst einen Professortitel verliehen. Er hat sicherlich einige seiner sechzig veröffentlichten Aufsätze abgeschrieben.

2.4.2 Teilplagiat

Anders als beim Totalplagiat werden beim Teilplagiat die Texte aus Büchern oder anderen Arbeiten nicht 1:1, sondern nur teilweise übernommen. Hier werden Sätze aus verschiedenen Quellen übernommen, teilweise umgeschrieben und durch eigene Sätze ergänzt.

Typisch für diese Arbeiten ist ein Stilbruch z.B. zwischen Kapiteln, in vielen Fällen werden auch die Fehler anderer Arbeiten mit übernommen. Diese Form des Plagiiens

bringt für den Plagiator sehr viel Arbeit mit sich, weil es sich hier um eine Zusammenfassung aus mehreren Arbeiten ohne Quellenangabe handelt.

2.4.3 Autoplagiat

Unter dem Begriff Autoplagiat versteht man die Wiederverwertung eigener Arbeiten. Auf der Wirtschaftsuniversität Wien [DP08] ist genau definiert, dass das Einreichen von ein und derselben Arbeit in verschiedenen Seminaren als Plagiat zu behandeln ist. Diese Form des Plagiiens sollte jedoch als harmlos im Vergleich zu den anderen Plagiatsformen betrachtet werden, da sich der Plagiator zumindest einmal die Mühe gemacht hat, für ein Thema zu recherchieren und die Vorlage auch selbst verfasst hat.

2.4.4 Übersetzungsplagiat

Bei dieser Plagiatsform werden Texte, die in fremden Sprachen vorliegen, übersetzt und als eigene Leistung präsentiert. Dieses Plagiat ist am Schwierigsten zu finden. Auch durch den Einsatz von Software wird sich die momentane Lage nicht schnell verbessern. Sehr oft werden Übersetzungsdienste zur Hilfe gezogen. Ein gutes Beispiel hierfür ist Babel Fish. Es nimmt viel Arbeit ab und liefert selbst für lange Texte eine relativ korrekte Übersetzung.

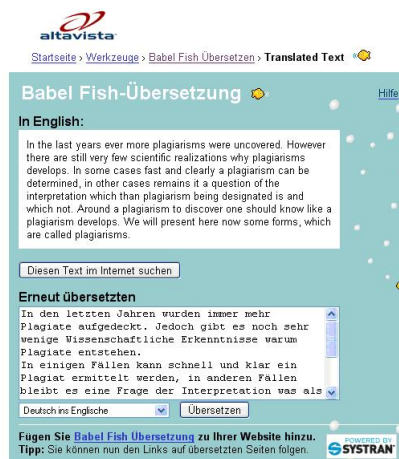


Abbildung 2.1: Babel Fish Uebersetzung
[Bf]

Diese Form des Plagiats ist sehr beliebt, weil vorhandene Arbeiten übernommen werden können und das Plagiat nur sehr schwer nachgewiesen werden kann.

2.4.5 Ideenplagiat

Das Ideenplagiat ist die wohl umstrittenste Form des Plagiiereus. Hier wird die Idee oder die Struktur einer anderen Arbeit übernommen. Der „Plagiator“ hält sich an die Strukturierung dieser anderen Arbeit, ohne sich selbst Gedanken über eine eigene Ausarbeitung zu machen.

Ob man diese Arbeiten als Plagiat einstufen kann, ist nicht ganz klar. Vieles spricht dafür, denn solange ein Autor sich nicht die Mühe macht, selbst eine Struktur für seine Arbeit zu entwerfen, dafür aber eine andere Arbeit benutzt, kann es nur ein Plagiat sein.

Es kann aber vorkommen, dass zwei Menschen unabhängig voneinander die selbe Idee haben. Dazu kommt noch die Tatsache, dass beim Ideenplagiat keine wörtliche Übernahme erfolgt. Es erweist sich als schwierig in diesem Fall ein Plagiat nachzuweisen.

2.4.6 Ghostwriting

Als Sonderfall eines Plagiats kann man das Ghostwriting betrachten. Hier schreibt man als Auftraggeber nicht selbst (ab), sondern lässt Andere gegen Bezahlung für sich schreiben. Diese Vorgangsweise ist an sich legal und in der Bellestrik nicht unüblich, in der Wissenschaft aber einem Plagiat gleich zu stellen, da eine fremde Leistung als die Eigene präsentiert wird.

Auf dieses Problem gehen die diversen Agenturen ein und empfehlen eine gründliche Überarbeitung nach Vertragsbeendigung. Die übergebenen Arbeiten sollen offiziell nur als „Hilfestellung“ für die eigene Arbeit dienen. Studenten die diese Hinweise nicht ernst nehmen bzw. diese bewusst umgehen, sollten damit rechnen, dieselbe Arbeit wieder zu schreiben oder negativ beurteilt zu werden.

Der Bestellvorgang ist meist einfach, die Agentur dient als Vermittler, somit ist die Anonymität gegenüber dem Schreiber gewährleistet. Es folgt eine Übersicht über einige Ghostwriting Anbieter:

Link	Beschreibung
www.drfranke.de/ index.php?id=10	Dr.Franke Consulting ist ein Text und Schreibbüro in Deutschland seit 1988 tätig. Das Team besteht aus 140 Mitarbeitern, es werden 150 Fachbereiche angeboten. Auf der Seite ist eine ausführliche Beschreibung über den Auftragsablauf und die Schweigepflicht zu finden
www.acad-write.com	Ein anderer Ghostwritinganbieter, der Hilfe bei Wissenschaftlichen Texten, Diplom- und Magisterarbeiten und Dissertationen anbietet. Preise pro Seite für weniger als 8 Arbeitstage sind 34,50 € Preise pro Seite für mehr als 8 Arbeitstage sind 26,50 € Eine genaue Aufteilung der angebotenen Fachbereiche ist nicht gleich zu finden. Für die Bestellung ist es erforderlich, ein Kontaktformular auszufüllen.
www.akademische-texte.de	Anbieter von Hausarbeiten, Seminararbeiten, Magisterarbeiten, Diplomarbeiten und Dissertationen. Je nach Fachgebiet liegt ein Seitenhonorar zwischen 26,50 € und 60,00 €. In der ABG wird ausdrücklich darauf hingewiesen, dass die Seiten nicht wortwörtlich übernommen werden dürfen, um später als eigene Arbeit abgegeben zu werden. Die wichtigsten Bereiche, die angeboten werden, sind: Wirtschaftswissenschaften; Politikwissenschaften; Pädagogik; Psychologie; Sprachwissenschaften
www.bombus-works.at	Bambus-Works Marketing and More bietet neben Preetexten und Trauersprüchen Hilfe bei der Verfassung von wissenschaftlichen Arbeiten Um mit der Agentur in Kontakt zu kommen muss man ein Formular ausfüllen. Keine Preisangabe vorhanden.

Tabelle 2.1: Ghostwriting Anbieter

3 Entstehung und Erkennung von Plagiaten

3.1 Entstehung

Plagiate sind nicht durch das Internet entstanden, auch Suchmaschinenanbieter wie Google oder Yahoo haben sie nicht erfunden. Aber man kann behaupten, dass das Internet und die diversen Suchmaschinen das Auffinden von Informationen wesentlich erleichtert haben: Informationen sind elektronisch verfügbar und können jederzeit vervielfältigt werden (Stichwort „copy & paste“). Die Versuchung ist daher sehr groß, Texte und Informationen ohne Quellenangabe zu übernehmen.

Sicher ist, dass Plagiate in den letzten Jahren mehr als zuvor unsere Aufmerksamkeit auf sich gezogen haben und so sind sie leider ein wichtiger Bestandteil unseres wirtschaftlichen und wissenschaftlichen Lebens geworden.

Es gibt keine genauen Daten, wie hoch die Anzahl der Plagiate ist, aber man kann davon ausgehen, dass nicht wenige Schüler oder Studenten beim Verfassen von Referaten oder Arbeiten auf „anonyme“ Quellen zurückgreifen. Hausarbeitenbörsen wie „www.hausarbeiten.de“ und Anbieter von Ghostwriting (siehe Abschnitt 2.4.6) sind Quellen, an die sich Schüler und Studenten oft wenden.

Plagiate entstehen nicht nur, wenn Schüler oder Studenten keine Quellen angeben, sondern auch, wenn Universitätsmitarbeiter auf fremde Arbeiten zugreifen und auf die Quellen vergessen oder Arbeiten der betreuten Studenten später als eigene herausgeben. Die Vorbildwirkung ist dadurch nicht mehr gewährleistet.

Warum Plagiat?

Es ist noch nicht ganz klar, warum plagiiert wird. Leistungsdruck und schlechtes Zeitmanagement können einige der Gründe sein. Der wichtigste Grund ist aber sicherlich die mangelnde Erfahrung im Recherchieren und im Zitieren.

Ohne einerseits zu wissen, wie man richtig recherchiert, werden Schüler unsicher und begehen schnell den Fehler, sich an fremden Arbeiten zu bedienen.

Ohne andererseits zu wissen, wie man richtig zitiert, können Studenten schnell den Fehler begehen, sich an Ghostwriting Anbieter zu wenden oder sich auch an fremden Werken zu bedienen ohne die Quelle anzugeben. Solange in diesem Bereich kein Lernprozess erfolgt, werden immer mehr Plagiate entstehen.

3.2 Erkennung

Zufall spielt mit

Lehrer und auch Professoren haben die Aufgabe, Arbeiten von Schülern und Studenten zu lesen und auf Korrektheit zu überprüfen. In vielen Fällen wird beim Korrekturlesen ein Plagiat ersichtlich, weil man sich dabei an die verschiedenen Stoffflecken einer Patchworkdecke erinnert fühlt.

Sobald in einem Teil einer Arbeit viele Rechtschreib- und Grammatikfehler vorhanden sind, in einem anderen Teil aber nicht, ist das ein deutlicher Hinweis auf ein Plagiat. Wechselnde Formatierung ist ein anderes Anzeichen dafür. Beispielsweise, wenn in einem Text oder innerhalb einer Passage gleichzeitig deutsche und auch französische Anführungszeichen zu finden sind, aber auch wechselnde Schriftart und Schriftgröße.

Plagiatsfallen

Plagiatsfallen sind eine andere Möglichkeit, Plagiaten auf die Spur zu kommen. Sie sind absichtlich eingebaute Fehler, um Plagiatoren zu überführen. Fingierte Lexikonartikel sind ein gutes Beispiel für Plagiatsfallen. Es handelt sich dabei um frei erfundene Einträge, die eigentlich nicht existieren sollten. Die Telekom Austria beispielsweise verwendet nicht existierende Telefonnummer in Telefonbüchern. [Pra07]

3.2.1 Erkennung durch Programme

Universitäten und Fachhochschulen haben sich im Kampf gegen Plagiate mit Software zur automatischen Erkennung gewappnet.

An der Universität Wien wird seit Herbst 2006 die Plagiats-Software *MyDropBox* verwendet. Mit Hilfe dieser Plagiats-Software soll jede der jährlich 5000 eingereichten Arbeiten überprüft werden. Um diese Überprüfung durchführen zu können, müssen die Studierenden ab dem Wintersemester 2006/2007 ihre Arbeiten elektronisch abgeben. Masterarbeiten, Diplomarbeiten und Dissertationen werden auf einem Server gespeichert und mit anderen vorhandenen Arbeiten auf dem Server und aus dem Internet verglichen. Bei einem positiven Testergebnis müssen die Studenten bei der jetzigen Gesetzeslage nur ein anderes Thema suchen und die Arbeit wiederholen. [Haf06]

Auch an der Wirtschafts- Universität in Wien und an der Universität für Bodenkultur wird die Plagiats-Software *MyDropBox* verwendet. [fBW07]

Laut einem Interview mit Herrn Professor Dr. Johann Blieberger mit der „Österreichischen Computer Gesellschaft“ wird auch an der Technischen Universität Wien daran gearbeitet, ein geeignetes Produkt zu finden und einsetzen zu können. Momentan wird überlegt, das Softwarepaket *MyDropBox* trotz ein paar Schwächen zu verwenden. [Mal06, S. 14]

Marktübersicht (Stand März 2008)

- **Turnitin** [iL08] ist eine Software, die Plagiate aufspürt. Die Datenbank verfügt über 40 Millionen Arbeiten von verschiedenen Studenten, tausende Bücher und über 12 Milliarden Web Seiten.

Sobald ein Student seine Arbeit abgibt bzw. Professoren die Ausarbeitungen der Studierenden mit Turnitin überprüfen, werden diese Daten in die interne Datenbank von Turnitin gespeichert. Auf der Seite von *Turnitin* wird angegeben, dass täglich um die 100.000 Ausarbeitungen von Studenten hinzugefügt werden und dass *Turnitin* in 90 Ländern in über 7000 Institutionen Weltweit verwendet wird.



Abbildung 3.1: Turnitin

[iL08]

- **MyDropBox** [Haf06] ist ähnlich wie Turnitin ein weit verbreitetes Programm zur Aufdeckung von Plagiaten. Laut *MyDropBox* Seite ist diese Software „Weltweit führend“.

Eingereichte Arbeiten werden in sicheren Datenbanken auf Servers in Toronto und Dallas gespeichert und mit über 8 Billionen Quellen aus dem Internet, Zeitungsartikeln, Büchern und anderen vorhandenen Arbeiten verglichen. Der Prozess des Prüfens dauert nur einige Minuten. Auch bei *MyDropBox* können entweder Studenten oder Professoren Arbeiten abgeben.

3 Entstehung und Erkennung von Plagiaten



Abbildung 3.2: MyDropBox [Haf06]

- **Ephorus** [Ger08] ist eine Plagiatssoftware, die bei dem Test der „Plagiatsjägerin“ Deborah Weber Wulf im Jahre 2007 am Besten abgeschnitten hat.

Je nach Bedarf kann zwischen *Ephorus Internet* (Vergleiche mit Internet Quellen), *Ephorus Group* (Vergleiche zwischen Dokumenten innerhalb einer wissenschaftlichen Einrichtung), *Ephorus Database* (Alle eingereichten Dokumente werden in einer Datenbank gespeichert) und *Ephorus Integrated* gewählt werden. *Ephorus* kann man auch kostenlos testen, man muss sich allerdings vorher registrieren. Studenten und Schüler können sich bei <http://student.ephorus.com> anmelden und ihre Arbeiten einreichen. Auf der *Ephorus* Seite ist eine Anleitung über die Demoversion zu finden.

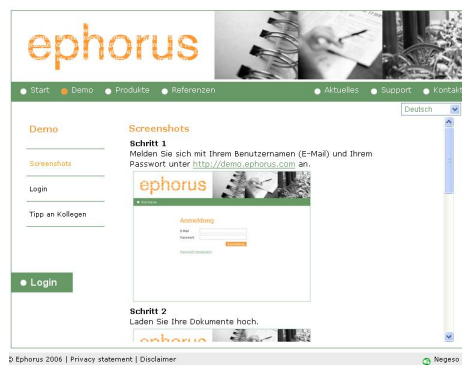


Abbildung 3.3: Ephorus
[Ger08]

Leistungsfähigkeit

Die „Plagiatsjägerin“ Debora Weber Wulff hat 2004 verschiedene Softwarepakete wie Turnitin, MyDropBox, PlagiarismFinder getestet. Ihre Meinung dazu:

3 Entstehung und Erkennung von Plagiaten

Platz	System	Punkte
1	Ephorus (www.ephorus.de)	42
2	Docoloc(www.docoloc.de)	35
3	Copyscape-Premium (www.copyscape.com)	34
4	PlagAware (www.plagaware.de)	34
5	Urkund (www.urkund.com)	34
6	Copyscape free (www.copyscape.com)	32
7	TextGuard (www.textguard.de)	29
8	ArticleChecker (www.articlechecker.com)	26
8	Turnitin (www.turnitin.com)	26
9	Picapica (www.picapica.net)	25
10	TextGuard (www.textguard.de)	17
11	DocCop (www.doccop.com)	12
12	iPlagiarismCheck (www.iplagiarismcheck.com)	12
13	StrikePlagiarism (www.strikeplagiarism.com)	0
14	CatchItFirst (www.catchitfirst.com)	0

Tabelle 3.1: Plagiats Software [Sie07]

„Die Technik kann den menschlichen Verstand nicht ersetzen. Ich habe verschiedene Programme getestet. Bei den meisten kann man genauso gut eine Münze werfen. Manche Systeme sind so beschränkt, dass sie korrekte Zitate für Plagiatsstellen halten. Auf der anderen Seite erkennen sie Artikel aus der Wikipedia nicht.“ [Hum05]

Im Jahr 2007 hat die Professorin Debora Weber Wulff wieder 14 Produkte getestet [Sie07]. Die Ergebnisse dieser Untersuchung sind auch dieses Mal wenig positiv ausgefallen. Keines der 14 Produkte schaffte es, über 50 von 60 möglichen Punkten zu erreichen.

Der Plagiatserkennungsdienst *CatchItFirst* schnitt am Schlechtesten ab. Von 20 Beispieltexen wurde kein Plagiat erkannt, das Ergebnis der 32 Stunden langen Untersuchung war bei allen Texten falsch.

Das System Turnitin hat bei dieser Untersuchung auch schlecht abgeschnitten. Von 60 möglichen Punkten hat das System nur 26 Punkte erreicht.

Nummer eins dieser Untersuchung war das System *Ephorus*. Es erreichte als einziges 42 Punkte.

Wieder einmal ist aus einer Untersuchung von Systemen zu Plagiatsfindung nichts Neues herausgekommen außer:

“Es gibt keine Software, mit der man Plagiaten zweifelsfrei auf die Spur kommt“ [Sie07]

Softwarepakete versagen sehr oft bei Übersetzungsplagiaten. Es ist schwierig, eine Arbeit als Plagiat zu identifizieren, die in einer anderen Fremdsprache veröffentlicht, dann in die deutsche Sprache übersetzt und als eigene Ausarbeitung eingereicht worden ist.

Der Einsatz von Softwarepaketen hat Vorteile und Nachteile. Kurzfristig gesehen helfen diese Produkte bei der Entlarvung von Plagiatoren. Langfristig betrachtet wird Plagiats-Software wahrscheinlich wenig an der Tatsache ändern, dass weiterhin plagiiert wird. Sie bleibt eine vorübergehende und kostspielige Lösung.

3.3 Plagiate vorbeugen

Oft kommt man durch Zufall darauf, dass es sich bei einer Arbeit um ein Plagiat handelt. Softwareprogramme zum Aufspüren von Plagiaten sind weit verbreitet und werden wie auch weiter oben schon erwähnt in vielen wissenschaftlichen Institutionen verwendet.

Nur: Wie kann man Plagiate vorbeugen?

Wichtig für die Zukunft ist, die Entstehung im Vorfeld durch geregelte Maßnahmen zu verhindern bzw. die Anzahl der Plagiate zu minimieren. Schüler und Studenten sollen dafür sensibilisiert werden, dass Plagieren mit Diebstahl gleichzusetzen ist. Durch Vorträge und einführende Lehrveranstaltungen können sie darauf aufmerksam gemacht werden und lernen, richtig zu zitieren und zu recherchieren. Sobald aus einer fremden Arbeit Passagen übernommen oder interpretiert werden, muss die Quelle angegeben werden.

Jede Universität bzw. wissenschaftliche Einrichtung kann einheitliche Richtlinien festlegen, die ein korrektes wissenschaftliches Verhalten und auch die Konsequenzen im Falle eines Fehlverhaltens beschreiben. Dazu empfehlen sich auch Maßnahmen, die das betreuende Personal entlasten und somit die Qualität der Beurteilung steigern. Weiters wäre es sinnvoll, soweit es möglich ist, neue Themen für wissenschaftliche Arbeiten zu finden, damit niemand in Versuchung kommt, vorhandene Arbeiten über ein bestimmtes Thema zu verwenden.

Die „Plagiatsjägerin“ Debora Weber Wulf bietet online einen Kurs an, der speziell für Lehrende gedacht ist, bietet aber darüber hinaus jedem Interessierten die Möglichkeit, sich über Plagiate zu informieren.

Lehrer können mit Hilfe dieser Seite besser verstehen, wie Plagiate entstehen, wie sie aufgedeckt werden können und was getan werden sollte um diesen vorzubeugen. Weiters können sich Lehrer selbst testen und herausfinden, wie gut sie Plagiate erkennen können. [Wul07, S. 1–58]

Der Weg zu einer Wissenschaft ohne Plagiate führt also über klare Konsequenzen, Bewusstseinsbildung, Qualität in der Lehre und Wissen über korrektes Arbeiten.

Richtiges Zitieren

Richtiges Zitieren ist ein wichtiger Punkt um Plagiate vorzubeugen. Bei allgemein bekannten Aussagen, beispielsweise: „Die Erde ist rund“, ist eine Quellenangabe nicht notwendig. Bei Aussagen, die nicht allgemein bekannt sind, etwa die Ergebnisse einer neuen Studie über die politische Orientierung einer Bevölkerungsgruppe, ist eine Quellenangabe erforderlich.

3 Entstehung und Erkennung von Plagiaten

Da sich jede wissenschaftliche Arbeit auf andere Arbeiten beziehen kann, ist es notwendig zu wissen, wie diese fremden Arbeiten in die eigene Arbeit eingebaut werden können. Zitate erlauben es, verschiedene Texte entweder Wort-wörtlich zu übernehmen oder diese mit eigenen Wörtern zu interpretieren.

Es gibt zwei Arten zu zitieren:

- Wort-wörtlich zitieren
Der übernommene Text ist identisch mit dem Original.
- Sinngemäß zitieren
Eine Paraphrase [Dri06] liegt vor, sobald ein Text aus dem Original in einer eigenen Arbeit in eigenen Worten interpretiert wird. Auch wenn es sich hier nicht um wörtliches Zitieren handelt, muss das Zitat gekennzeichnet sein.

Fremdsprachige Autoren müssen in der Originalsprache zitiert werden. Schwierigkeiten bereitet das Zitieren aus Internet-Quellen. Hier ist wichtig, die korrekte Internetadresse und den Autor anzugeben. Die letzten Änderungen, die auf der Seite durchgeführt worden sind bzw. das Datum wann diese Seite das letzte Mal besucht worden ist, sollte auch angegeben werden, damit klar ist, auf welche Version der Seite sich eine Arbeit bezogen hat.

4 Problem „Plagiat“ in Österreich

4.1 Situation in Österreich

Um die momentane Situation in Österreich beschreiben zu können, sollten genaue Studien über die Anzahl der jährlich aufgedeckten Plagiate vorliegen. Da diese nicht vorliegen bzw. die Meinungen über das Problem Plagiat sehr unterschiedlich sind, ist es fast unmöglich, eine exakte Beschreibung durchzuführen.

Die Österreichische HochschülerInnenschaft forderte 2006 die gesetzliche Festlegung des Begriffs Plagiat. Ihr Vorschlag war folgender:

„...eine Arbeit, bei der mehr als ein Viertel unzitirt aus einer anderen Quelle verwendet wird, als Vollplagiat gilt.“ [sH07]

Darüber hinaus forderte die österreichische HochschülerInnenschaft eine gesetzliche Regelung von Sanktionen im Falle eines Plagiats, von der Neuverfassung der Arbeit bis hin zur Aberkennung des Akademischen Grades.

Die Universität Wien hat im Jänner 2006 im Mitteilungsblatt [S. 1–14][Wie06] eine Richtlinie für gute wissenschaftliche Arbeit veröffentlicht. In dieser Mitteilung steht, wie gute wissenschaftliche Arbeit gewährleistet werden soll, wie mit wissenschaftlichem Fehlverhalten umzugehen ist und was die Konsequenzen sind.

Auszüge aus dem Mitteilungsblatt [S. 1–14][Wie06]:

Wissenschaftliches Fehlverhalten

”

§ 4 Wissenschaftliches Fehlverhalten von WissenschaftlerInnen

Ein wissenschaftliches Fehlverhalten liegt vor:

...

2. bei Verletzung geistigen Eigentums anderer Wissenschaftlerinnen oder Wissenschaftler. Sie liegt insbesondere vor bei

- *unbefugter Verwertung unter Anmaßung der Autorenschaft (Plagiat);*

4 Problem „Plagiat“ in Österreich

- *der Ausbeutung von fremden Forschungsansätzen und Ideen, insbesondere als Gutachterin oder als Gutachter (Ideendiebstahl);*
- *der Anmaßung oder Hinnahme unbegründeter wissenschaftlicher Autoren- oder Mitautorenschaft;*
- *der unbefugten Veröffentlichung und unbefugten Zugänglichmachung gegenüber Dritten, solange die Autorin oder der Autor das Werk, die Erkenntnis, die Hypothese, die Lehrmeinung oder den Forschungsansatz noch nicht veröffentlicht hat;*

“

Konsequenzen für Wissenschaftliches Fehlverhalten

Akademische Konsequenzen

”

§ 17 Akademische Konsequenzen

1. *Allfällige inneruniversitäre Konsequenzen wissenschaftlichen Fehlverhaltens werden von der Rektorin oder vom Rektor veranlasst.*
2. *Weiters kommt nach Maßgabe der einschlägigen gesetzlichen Bestimmungen die Aberkennung eines von der Universität Wien verliehenen akademischen Grades oder Titels in Betracht, wenn die Verleihung des Grades oder Titels durch vorsätzliches oder grob fahrlässiges wissenschaftliches Fehlverhalten erlangt wurde. Im Fall der Feststellung eines entsprechend gravierenden wissenschaftlichen Fehlverhaltens informiert die Rektorin oder der Rektor die für solche Maßnahmen zuständigen Universitätsorgane.*
3. *Andere Universitäten oder außeruniversitäre wissenschaftliche Einrichtungen und Vereinigungen werden über ein wissenschaftliches Fehlverhalten dann durch die Rektorin oder den Rektor informiert, wenn diese*

4 Problem „Plagiat“ in Österreich

Universitäten, Einrichtungen oder Vereinigungen davon unmittelbar berührt sind oder wenn die betroffene Wissenschaftlerin bzw. der betroffene Wissenschaftler eine leitende Stellung in der betreffenden Institution einnimmt oder in Entscheidungsgremien von Förderorganisationen oder dergleichen mitwirkt.

“

Zivilrechtliche Konsequenzen

”

§ 19 Zivilrechtliche Konsequenzen

Als zivilrechtliche Konsequenzen kommen bei wissenschaftlichem Fehlverhalten insbesondere Herausgabeansprüche (etwa im Hinblick auf entwendetes Material), Unterlassungsansprüche aus Urheberrecht, Patentrecht und Wettbewerbsrecht, Rückforderungsansprüche (etwa von Stipendien, Drittmitteln) oder Schadenersatzansprüche der Universität Wien oder Dritter in Betracht.“

Strafrechtliche Konsequenzen

”

§ 20 Strafrechtliche Konsequenzen

Ergibt sich der Verdacht einer von Amts wegen zu verfolgenden gerichtlich strafbaren Handlung, ist der Rektorin oder dem Rektor zwecks Erstattung einer Anzeige (§ 84 StPO) sofort Bericht zu erstatten. “

Am 17 März 2007 ist an der Technischen Universität Wien, Fakultät für Informatik, ein *Code of Ethics* veröffentlicht worden. Hier werden die ethischen Richtlinien, die jeder Angehörige der Technischen Universität Wien an der Fakultät für Informatik bei Verfassen einer wissenschaftlichen Arbeit beachten und einhalten sollte, beschrieben. [fITUW07, S. 1-3]

Bis jetzt fehlt zwar eine einheitliche Regelung, man merkt aber, dass in diese Richtung gearbeitet wird. Die veröffentlichten Regelungen der Universitäten werden wahrscheinlich das wissenschaftliche Fehlverhalten nicht so bald beseitigen, sie tragen aber dazu bei, wissenschaftliches Fehlverhalten einzuschränken.

Stefan Weber [Web07] hat sich in den letzten Jahren sehr intensiv mit dem Thema „Plagiat“ beschäftigt. Er ist der Meinung, dass die Anzahl der Plagiate an österreichischen Universitäten relativ hoch ist. Durch seine Recherchen wurden einige Plagiate

aufgedeckt. Das wohl bekannteste Beispiel ist die Diplomarbeit „Wickie und die starken Männer“.

In der Alpen-Adria Universität in Klagenfurt bezeichnete er die genannte Arbeit als Plagiat. Die „Ombudsstelle zur Sicherung der guten wissenschaftlichen Praxis“ der Universität in Klagenfurt stufte die Arbeit 2006 nach mehreren Gutachten auch als Plagiat ein. Die Autorin dieser Diplomarbeit war in der Zeit Universitätsassistentin und ihre Arbeit wurde damals vom zuständigen Professor mit einem *Sehr gut* beurteilt. Nach Aufdeckung des Plagiats wurde das Dienstverhältnis aufgelöst, die Autorin bekam jedoch die Möglichkeit, ihre Arbeit neu zu schreiben.

Die Alpen-Adria Universität hat daraufhin beschlossen, dass Dissertationsarbeiten online abgegeben werden müssen. Andere Arbeiten werden seit dem Vorfall noch gründlicher überprüft.

4.2 Rechtslage in Österreich

4.2.1 Urheberrechtsgesetz

Laut § 10 des UrhG ist der Urheber eines Werkes derjenige, der es geschaffen hat.

„ § 10.

1. *Urheber eines Werkes ist, wer es geschaffen hat.*
2. *In diesem Gesetz umfaßt der Ausdruck „Urheber“, wenn sich nicht aus dem Hinweis auf die Bestimmung des Absatzes 1 das Gegenteil ergibt, außer dem Schöpfer des Werkes auch die Personen, auf die das Urheberrecht nach seinem Tode übergegangen ist.*

“ [Öst]

Haben mehrere Personen ein Werk geschaffen, so sind laut § 11 des UrhG alle Miturheber.

„

§ 11

1. *Haben mehrere gemeinsam ein Werk geschaffen, bei dem die Ergebnisse ihres Schaffens eine untrennbare Einheit bilden, so steht das Urheberrecht allen Miturhebern gemeinschaftlich zu.*

4 Problem „Plagiat“ in Österreich

2. *Jeder Miturheber ist für sich berechtigt, Verletzungen des Urheberrechtes gerichtlich zu verfolgen. Zu einer Änderung oder Verwertung des Werkes bedarf es des Einverständnisses aller Miturheber. Verweigert ein Miturheber seine Einwilligung ohne ausreichenden Grund, so kann ihn jeder andere Miturheber auf deren Erteilung klagen. Hat der Beklagte im Inland keinen allgemeinen Gerichtsstand, so sind die Gerichte, in deren Sprengel der erste Wiener Gemeindebezirk liegt, zuständig.*
3. *Die Verbindung von Werken verschiedener Art - wie die eines Werkes der Tonkunst mit einem Sprachwerk oder einem Filmwerk - begründet an sich keine Miturheberschaft.*

“

[Öst]

Das Urheberrecht ist laut § 23 des UrhG vererblich jedoch nicht übertragbar.

”

§ 23.

1. *Das Urheberrecht ist vererblich; in Erfüllung einer auf den Todesfall getroffenen Anordnung kann es auch auf Sondernachfolger übertragen werden.*
2. *Wird die Verlassenschaft eines Miturhebers von niemand erworben und auch nicht als erbloses Gut vom Staat übernommen, so geht das Miturheberrecht auf die anderen Miturheber über. Dasselbe gilt im Falle des Verzichtes eines Miturhebers auf sein Urheberrecht, soweit dieser Verzicht wirkt.*
3. *Im Übrigen ist das Urheberrecht unübertragbar.*
4. *Geht das Urheberrecht auf mehrere Personen über, so sind auf sie die für Miturheber (§ 11) geltenden Vorschriften entsprechend anzuwenden.*

“ [Öst]

4 Problem „Plagiat“ in Österreich

Der Urheber eines Werkes hat als einziger das Recht zu entscheiden, wann und von wem sein Werk veröffentlicht werden soll. Nach § 15 des UrhG bestimmt auch hier alleine der Urheber, in welcher Menge und mit welchem Verfahren sein Werk vervielfältigt wird.

”

§ 15.

1. *Der Urheber hat das ausschließliche Recht, das Werk - gleichviel in welchem Verfahren, in welcher Menge und ob vorübergehend oder dauerhaft - zu vervielfältigen.*
2. *Eine Vervielfältigung liegt namentlich auch in dem Festhalten des Vortrages oder der Aufführung eines Werkes auf Mitteln zur wiederholbaren Wiedergabe für Gesicht oder Gehör (Bild- oder Schallträger), wie zum Beispiel auf Filmstreifen oder Schallplatten.*
3. *Solchen Schallträgern stehen der wiederholbaren Wiedergabe von Werken dienende Mittel gleich, die ohne Schallaufnahme durch Lochen, Stanzen, Anordnen von Stiften oder auf ähnliche Art hergestellt werden (Drehorgeln, Spieldosen u. dgl.).*
4. *Bei Plänen und Entwürfen zu Werken der bildenden Künste umfaßt das Vervielfältigungsrecht auch das ausschließliche Recht, das Werk danach auszuführen.*

“ [Öst]

Von großem Interesse im Hinblick auf Plagiate sind die im Urheberrechtsgesetz definierten Zitierregeln:

Zitatrecht

Zitieren aus urheberrechtlich geschützten Werken hängt auch mit der freien Werknützung zusammen. Ein eigenes Werk ist Voraussetzung für ein Zitat. Eine Arbeit sollte nicht ausschliesslich aus Zitaten bestehen, diese dienen wie auch im *Abschnitt 3.3* beschrieben nur als Unterstützung der eigenen Arbeit.

- Kleinzitat

4 Problem „Plagiat“ in Österreich

„ § 46.

Zulässig sind die Vervielfältigung und die Verbreitung sowie der öffentliche Vortrag, die Rundfunksendung und die öffentliche Zurverfügungstellung:

1. (1) *wenn einzelne Stellen eines veröffentlichten Sprachwerkes angeführt werden;*

”

- Großzitat

„ § 46.

Zulässig sind die Vervielfältigung und die Verbreitung sowie der öffentliche Vortrag, die Rundfunksendung und die öffentliche Zurverfügungstellung:

1. (2) *wenn einzelne Sprachwerke oder Werke der im § 2, Z. 3, bezeichneten Art nach ihrem Erscheinen in einem durch den Zweck gerechtfertigten Umfang in ein die Hauptsache bildendes wissenschaftliches Werk aufgenommen werden; ein Werk der im § 2, Z. 3, bezeichneten Art darf nur zur Erläuterung des Inhaltes aufgenommen werden.*

“

- Kunstzitat

„ § 54. (1)

- 3a. *einzelne erschienene Werke der bildenden Künste in einem die Hauptsache bildenden wissenschaftlichen Werk zu vervielfältigen, zu verbreiten und der Öffentlichkeit zur Verfügung zu stellen;*

“

- musikalisches Kleinzitat

„ § 52.

4 Problem „Plagiat“ in Österreich

Zulässig sind die Vervielfältigung und die Verbreitung sowie die öffentliche Aufführung, die Rundfunksendung und die öffentliche Zurverfügungstellung:

- 1. wenn einzelne Stellen eines erschienenen Werkes der Tonkunst in einem selbstständigen neuen Werke der Tonkunst angeführt werden;*
- 2. wenn einzelne Stellen eines veröffentlichten Werkes der Tonkunst in einer literarischen Arbeit angeführt werden;*
- 3. wenn einzelne erschienene Werke der Tonkunst in einem durch den Zweck gerechtfertigten Umfang in ein die Hauptsache bildendes wissenschaftliches Werk aufgenommen werden.*

“

§ 52. (3a) UrhG Es ist zulässig einzelne Stellen eines erschienenen Werkes der Tonkunst in einem selbstständigen neuen Werk der Tonkunst anzuführen.

4.2.2 Geschützte Werke

Werke der Literatur

„ § 2.

Werke der Literatur im Sinne dieses Gesetzes sind:

- 1. Sprachwerke aller Art einschließlich Computerprogrammen (§ 40a);*
- 2. Bühnenwerke, deren Ausdrucksmittel Gebärden und andere Körperbewegungen sind (choreographische und pantomimische Werke);*
- 3. Werke wissenschaftlicher oder belehrender Art, die in bildlichen Darstellungen in der Fläche oder im Raume bestehen, sofern sie nicht zu den Werken der bildenden Künste zählen.*

“

Schutzfristen Das Urheberrecht an Werken endet 70 Jahre nach dem Tod des letzten Miturhebers.

Werke der bildenden Kunst

“ § 3.

1. *Zu den Werken der bildenden Künste im Sinne dieses Gesetzes gehören auch die Werke der Lichtbildkunst (Lichtbildwerke), der Baukunst und der angewandten Kunst (des Kunstgewerbes).*
2. *Werke der Lichtbildkunst (Lichtbildwerke) sind durch ein photographisches oder durch ein der Photographie ähnliches Verfahren hergestellte Werke.*

“

Schutzfristen

„ § 74.

(6) Das Schutzrecht an Lichtbildern erlischt fünfzig Jahre nach der Aufnahme, wenn aber das Lichtbild vor dem Ablauf dieser Frist veröffentlicht wird, fünfzig Jahre nach der Veröffentlichung. Die Fristen sind nach § 64 zu berechnen.“

Werke der Filmkunst

„ § 4.

Unter Werken der Filmkunst (Filmwerke) versteht dieses Gesetz Laufbildwerke, wodurch die den Gegenstand des Werkes bildenden Vorgänge und Handlungen entweder bloß für das Gesicht oder gleichzeitig für Gesicht und Gehör zur Darstellung gebracht werden, ohne Rücksicht auf die Art des bei der Herstellung oder Aufführung des Werkes verwendeten Verfahrens.“

Schutzfristen

Filmwerke

„ § 62.

Das Urheberrecht an Filmwerken endet siebenzig Jahre nach dem Tod der zuletzt lebenden der folgenden Personen, und zwar des Hauptregisseurs sowie des Urhebers des Drehbuchs, der Dialoge und des für das Filmwerk besonders geschaffenen Werkes der Tonkunst.“

4.3 Universitätsgesetz

Das Universitätsgesetz hat keine genaue Beschreibung des Begriffs Plagiat und auch keine genau definierten Maßnahmen gegen Personen, die im wissenschaftlichen Bereich plagieren. Im § 74 (2) UG 2002 ist festgelegt wann eine Prüfung oder eine wissenschaftliche Arbeit für nichtig zu erklären ist:

„ § 74.

1. *Das für die studienrechtlichen Angelegenheiten zuständige Organ hat die Beurteilung einer Prüfung mit Bescheid für nichtig zu erklären, wenn die Anmeldung zu dieser Prüfung erschlichen wurde.*
2. *Überdies ist die Beurteilung einer Prüfung, einer wissenschaftlichen Arbeit oder einer künstlerischen Master- oder Diplomarbeit mit Bescheid für nichtig zu erklären, wenn diese Beurteilung, insbesondere durch die Verwendung unerlaubter Hilfsmittel, erschlichen wurde*
3. *Die Prüfung, deren Beurteilung für nichtig erklärt wurde, ist auf die Gesamtzahl der Wiederholungen anzurechnen.*
4. *Prüfungen, die außerhalb des Wirkungsbereiches einer Fortsetzungsmeldung abgelegt wurden, und Beurteilungen wissenschaftlicher Arbeiten sowie künstlerischer Master- und Diplomarbeiten, die außerhalb des Wirkungsbereiches einer Fortsetzungsmeldung erfolgten, sind absolut nichtig. Eine Anrechnung auf die Gesamtzahl der Wiederholungen erfolgt nicht.*

“

Im § 80 (2) UG 2002 ist vorgegeben, dass bei der Erstellung von Bachelorarbeiten, Masterarbeiten, Diplomarbeiten und auch bei Dissertationen die Bestimmungen des Urheberrechtsgesetzes zu beachten sind.

„ § 80.

1. *Im Bachelorstudium sind im Rahmen von Lehrveranstaltungen Bachelorarbeiten abzufassen. Nähere Bestimmungen über Bachelorarbeiten sind im jeweiligen Curriculum festzulegen.*

4 Problem „Plagiat“ in Österreich

2. *Bei der Bearbeitung des Themas und der Betreuung der Studierenden sind die Bestimmungen des Urheberrechtsgesetzes, BGBl. Nr. 111/1936, zu beachten.*

“

„ § 81.

1. *Im Diplom- oder Masterstudium ist eine Diplom- oder Masterarbeit abzufassen. In besonders berufsorientierten Studien ist es zulässig, im Curriculum anstelle der Diplom- oder Masterarbeit einen anderen gleichwertigen Nachweis vorzusehen. Die Abfassung als Klausurarbeit ist unzulässig. Nähere Bestimmungen über Betreuung und Beurteilung von Diplom- oder Masterarbeiten sind in der Satzung, nähere Bestimmungen über das Thema der Diplom- oder Masterarbeit sind im jeweiligen Curriculum festzulegen.*
2. *Die Aufgabenstellung der Diplom- oder Masterarbeit ist so zu wählen, dass für eine Studierende oder einen Studierenden die Bearbeitung innerhalb von sechs Monaten möglich und zumutbar ist.*
3. *Die gemeinsame Bearbeitung eines Themas durch mehrere Studierende ist zulässig, wenn die Leistungen der einzelnen Studierenden gesondert beurteilbar bleiben. Erfordert die Bearbeitung eines Themas die Verwendung der Geld- oder Sachmittel von Einrichtungen der Universität, so ist die Vergabe nur zulässig, wenn die Leiterin oder der Leiter dieser Einrichtung über die beabsichtigte Vergabe informiert wurde und diese nicht binnen eines Monats wegen einer wesentlichen Beeinträchtigung des Lehr- und Forschungsbetriebes untersagt hat.*
4. *§ 80 Abs. 2 gilt auch für Diplom- und Masterarbeiten.*

“

Wird ein akademischer Grad zu unrecht verliehen, so werden im § 89 UG die Folgen für solche Handlungen beschrieben:

4 Problem „Plagiat“ in Österreich

„ § 89.

Der Verleihungsbescheid ist vom für die studienrechtlichen Angelegenheiten zuständigen Organ aufzuheben und einzuziehen, wenn sich nachträglich ergibt, dass der akademische Grad insbesondere durch gefälschte Zeugnisse erschlichen worden ist.“

5 Plagiat in der Wirtschaft

Plagiate sind auch in der Wirtschaft nicht fremd, Produkt- und Markenfälschung sind hier sehr verbreitet.

„Als Produktpiraterie bzw. Produktfälschung wird das Nachahmen oder Fälschen von Produkten bezeichnet, also die gezielte Verletzung von Markenrechten, Patentrechten, Urheberrechten und sonstigen gewerblichen Schutzrechten und deren illegale Nutzung“ [fFs07]

Plagiate werden unter demselben Namen oder einem anderen Namen verkauft. Ziel des Plagiiens sind meistens erfolgreiche Konsumgüter wie: Louis Vuitton Taschen, Schuhe, Kosmetikprodukte, Spielzeug, Lebensmittel, Porzellan bis zu High-Tech Produkte wie MP3 Player. Auch die Automobilindustrie und sogar die Pharmaindustrie sind von Produktfälschungen betroffen.

Das Bundesministerium für Finanzen in Österreich hat die Warengruppen der beliebtesten Fälschungen im ersten Quartal 2007 veröffentlicht. Aus Abbildung 5.1 kann man sehen, dass Medikamente mit 6878 gefälschten Artikeln an ersten Stelle stehen, gefolgt von Computern mit 3968 Artikeln. An dritter Stelle kommen Konsumgüter wie Uhren, Schmuck, Schmuck, Bekleidungszubehör und andere Bekleidung. Für das erste Quartal des Jahres 2007 wurden 180 Fälle von gefälschten Medikamenten gefunden und 312 Fälle von gefälschten Computern. [fFs07, S. 1]

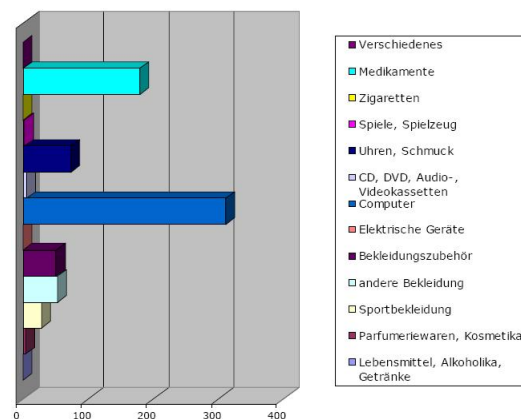


Abbildung 5.1: Anzahl der Fälle - Aufteilung der Fälschungen nach Warengruppen [fFs07]

5 Plagiat in der Wirtschaft

Auf der Homepage der Europäischen Kommission für Steuern und Zollwesen [Com07] ist eine Aufteilung nach Warengruppen für die ganze Europäische Union zu finden. Aus Abbildung 5.2 kann man ersehen, dass an erster Stelle Zigaretten in Höhe von 73.920.446 Stück stehen, an zweiter Stelle sind CDs und DVDs mit 15.080.161 Stück gefolgt von Kleidern und anderen Konsumgütern in Höhe von 14.361.867 Stück.

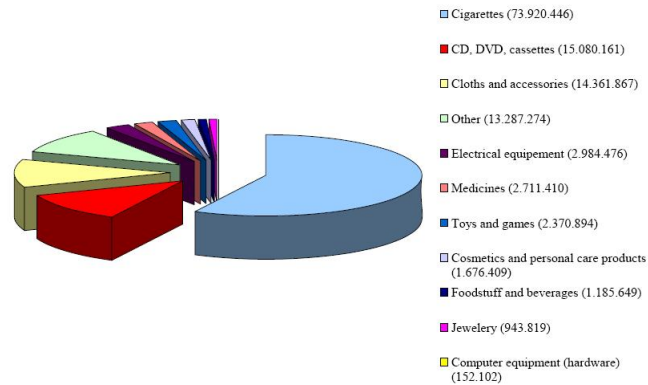


Abbildung 5.2: Fälschungen nach Warengruppen innerhalb der EU [Com07]

Die Europäische Kommission [Com07] hat im Jahresbericht 2006 eine Statistik veröffentlicht, wie sich die Anzahl der Fälschungen in den Jahren 1999 bis 2006 entwickelt hat. (Abbildung 5.3) Man kann beobachten, dass sich die Anzahl der Fälschungen in den Jahren 1999 bis 2002 nicht sehr stark geändert hat. Anders schaut die Situation von 2003 bis 2006 aus. Hier kann man einen deutlichen Anstieg an Fälschungen erkennen.

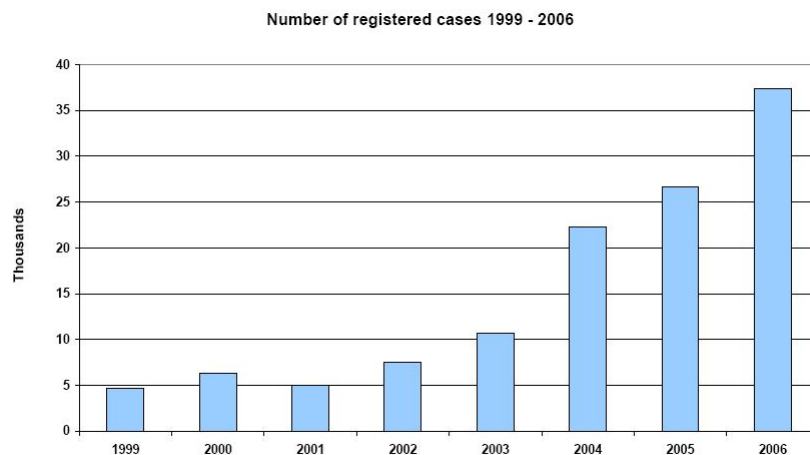


Abbildung 5.3: registrierte Fälle 1996-2006 [Com07]

Erfolgreiche Produkte werden nicht ohne Grund kopiert. Für solche Produkte existiert eine hohe Nachfrage. Plagiatoren verdienen auf Kosten des ursprünglichen Herstellers. Sie benötigen keine Marktforschung mehr, weil schon bekannt ist, welche Produkte beliebt sind. Produktfälscher ersparen sich die mühevollen Arbeit anderer Unternehmen und übernehmen sie mit dem Ziel persönlicher Bereicherung. Das gute Image anderer Unternehmen wird hier stark missbraucht.

Gefälschte Waren sind anhand ihres Aussehens von Konsumenten schwer zu erkennen. China ist eines der Problemländer im Bereich der Produkt- und Markenfälschung.

In China werden verschiedenste Fälschungen billig in Fabriken hergestellt. Die gefälschten Produkte kommen nach Europa und sind zu niedrigsten Preisen erhältlich. Viele von uns tragen durch den Kauf gefälschter Produkte zur Steigerung der Produktion bei.

China ist jedoch nicht das einzige Problemland, es ist dicht gefolgt von Indien, Vietnam, Thailand, Russland und der Türkei. In Asien ist nicht nur die gesetzliche Lage anders, da kein richtiger Rechtsschutz vorhanden ist, sondern auch die Kultur. Man kann behaupten, dass dort Kopieren nichts Unehrenhaftes ist.

„Wer große Meister kopiert erweist Ihnen Ehre.“ Konfuzius

In den letzten Jahren hat sich die Lage verschlechtert. Plagiatoren haben ihr Angebot erweitert. Durch das Plagiiere von No-Name Produkten haben sie die Möglichkeit, mehr Umsatz zu machen. Den Konsumenten fällt beim Kauf gefälschter Produkte nicht auf, ob es sich um eine Fälschung handelt oder um ein Original, denn bei den meisten No-Name Produkten weiß niemand, dass ein Original existiert. Man muss noch einmal hinzufügen, dass in China bzw. Asien noch kein richtiger Rechtsschutz existiert.

Zwar wurde in China der Rechtsschutz seit dem WTO-Beitritt 2001 verbessert, doch die Umsetzung der Gesetze ist in der Praxis immer noch ein schwieriges Thema. Die Ausbildung der Beamten und Fachkräfte ist eine der wichtigsten Aufgaben, die sich China vorgenommen hat, um den Schutz des geistigen Eigentums so gut wie möglich zu gewährleisten.

Die Europäische Union hat aufgrund der immer größer werdenden Anzahl an Fälschung und deren Verbreitung reagiert. In Österreich haben die Zollbeamten nun die Möglichkeit, Fälschungen aus Drittländern frühzeitig aus dem Verkehr zu ziehen um so zu verhindern, dass gefälschte Produkte in den Österreichischen und Europäischen Markt gelangen. Diese Möglichkeit ist rechtlich durch das Produktpirateriegesetz gesichert. In Abbildung 5.4 folgt eine Übersicht der Produktpiraterie-Aufgriffe der Österreichischen Zollbehörden [fFs07] im ersten Quartal 2007. An erster Stelle ist Indien, gefolgt von China und Hongkong.

5 Plagiat in der Wirtschaft

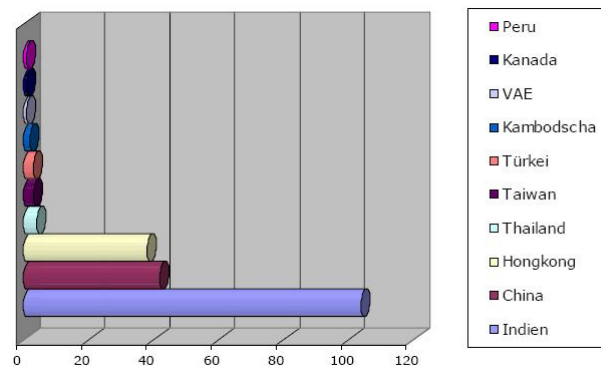


Abbildung 5.4: Anzahl der Fälle - Herkunftsland gefälschter Artikel in Österreich [fFs07]

Von der Europäischen Kommission [Com07] sind für das Jahr 2006 Statistiken für die Europäische Union veröffentlicht worden. Die Lage in der Europäischen Union kann man aus Abbildung 5.5 erfahren, hier ist China die Hauptquelle an gefälschten Produkten, 79 % der Fälschungen kommen von dort.

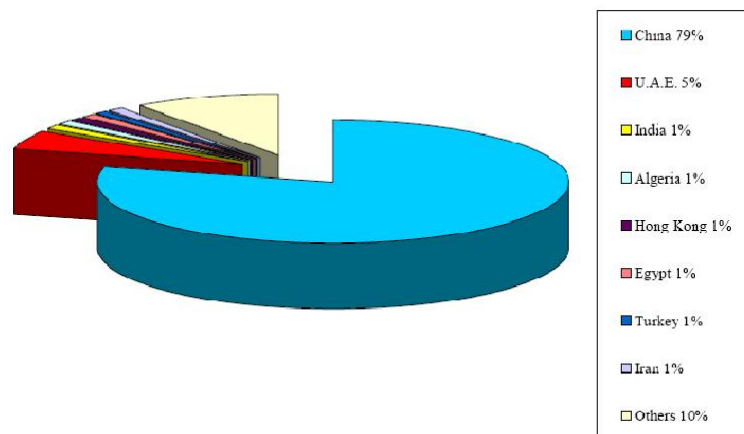


Abbildung 5.5: Herkunftsland gefälschter Artikel in der EU [Com07]

5.1 Produktpiraterie

Um sich vor Produktpiraterie schützen zu können muss ein Unternehmen zuerst feststellen, dass die eigenen Produkte gefälscht worden sind. Eingehende Beschwerden der Kunden können hier eine große Hilfe sein. Unternehmen können durch ausreichende Marktbeobachtungen feststellen ob ihre Produkte gefälscht werden oder nicht. Ein gefälschtes,

Produkt kann das Image eines renommierten Unternehmens zerstören. Dies kann zu Kundenverlust führen und als Folge noch mehr finanzielle Verluste mit sich bringen. Jedes Unternehmen sollte seine Produktentwicklung geheim halten und Produkte durch Kennzeichnungen sichern, so dass es für die Fälscher schwierig ist, diese Produkte nachzuahmen.

Seit 1986 existiert der Verein „Aktion Plagiarius“ [Bus07]. Dieser Verein verleiht jedes Jahr eine negative Auszeichnung. Die Auszeichnung, ein schwarzer Zwerg mit einer goldenen Nase, geht jedes Jahr an die besten Plagiatoren des Jahres. Die Verleihung soll die Verbraucher und die Medien sensibilisieren und sie auf die Lage aufmerksam machen. Ein paar Fakten zur Fälschungen [Bus07]:

”

- *10 Prozent des Welthandels sind Fälschungen und Nachahmungen*
- *Weltweiter volkswirtschaftlicher Schaden pro Jahr: EUR 200-300 Mrd. (Deutschland: EUR 29 Mrd.)*
- *Weltweiter Verlust von Arbeitsplätzen pro Jahr: 200.000 (Deutschland: 70.000)...*

“

Im Museum Plagiarius in Solingen werden seit 1. April 2007 rund 250 Produkte als Original und Fälschung ausgestellt. Das Plagieren bzw. der Ideendiebstahl kostet in Europa sehr viele Arbeitsplätze. Unternehmen, deren Produkte gefälscht werden, sind gezwungen, Sparmaßnahmen einzuführen um die durch Fälschung verursachten Verluste zu minimieren.

Aber das ist noch nicht genug. Die Gesundheit der KonsumentInnen ist durch Produktfälschung sehr gefährdet, vor allem weil die Fälscher es immer mehr auf die Pharmaindustrie und Lebensmittel abgesehen haben. Ein großes Problem verursacht der Verkauf von gefälschten Medikamenten über das Internet. Die Konsumenten müssen darauf achten, was sie kaufen, denn oft werden sie von Betrügern durch billige Preise angelockt. Der Kauf solcher Medikamente bringt ein sehr großes Risiko mit sich, denn diese Plagiate sind qualitativ mangelhaft und können die Gesundheit der Konsumenten gefährden. Auf der Homepage des Bundesministeriums für Finanzen [fFs08] steht ausführlich beschrieben, worauf die Konsumenten und Konsumentinnen beim Internet Shopping achten sollten.

6 technischer Teil

6.1 Einleitung

6.1.1 LVA Beschreibung

In der VU Daten- und Informatikrecht beantworten jedes Semester bis zu 500 Studentinnen und Studenten drei Fragestellungen kurz mit jeweils ca. 300 Wörtern in TUWEL, der E-Learning Plattform der technischen Universität Wien. Gruppenarbeit ist dabei nicht erlaubt, jede/r Student/in hat die Antworten alleine zu verfassen. Allerdings verleiten die „anonyme“, rein elektronische Abgabe und die technisch einfache Möglichkeit (Stichwort copy & paste) dazu, identische oder sehr ähnliche Antworten zu geben.

6.1.2 Aufgabenstellung

Die Abgaben werden von der LVA Leitung gelesen und einzeln bewertet. Oft ergibt sich dabei schon beim Durchlesen ein Gefühl dafür, wenn verbotenerweise zusammengearbeitet wurde. Bei bis zu 500 Texten mit gleicher Thematik ist dieses allerdings eher auf Zufall begründet.

Ein technisches Hilfsmittel, das verdächtige Abgaben automatisiert erkennt, ist gefragt. Im Vordergrund soll dabei Genauigkeit stehen, Geschwindigkeit spielt eine untergeordnete Rolle.

6.1.3 Ansatz

Ziel war ein Programm, das die Abgaben aus TUWEL entgegennimmt, paarweise vergleicht und verdächtige Abgaben ausgibt. Als Einstiegspunkt in die Thematik wurde von unserem Betreuer Dipl. Ing. Michael Tesar ein fertiger Algorithmus ¹ vorgeschlagen, der entsprechend adaptiert werden sollte.

Zusammenfassend bestand die Aufgabe darin, die Abgaben aus einer Datei in einem XML Format einzulesen, das Kreuzprodukt daraus zu bilden, auf jede Paarung einen Algorithmus anzuwenden, die Ergebnisse auszuwerten und bei Verdacht auszugeben.

¹aspHeute: <http://www.aspheute.com/artikel/20030910.htm>

PPR / Plagiate finden - Algorithmen Test

http://user.phil-fak.uni-duesseldorf.de/~loebner/lehre/hausarb/plagiate.htm zu finden.'"/>

Abbildung 6.1: Eingabeseite des ersten Prototypen in PHP

6.2 Projektstart

6.2.1 Vorbedingungen

Da die Suche nach Kopien paarweise erfolgt, wurde die technische Lösung auf zeichenweises Vergleichen beschränkt. Damit erübrigte sich die Implementierung einer Datenbank und es konnte auch angenommen werden, dass Leerzeichen und Zeilenschaltungen irrelevant sind.

Als Endergebnis wurde die Anzahl der übereinstimmenden Zeichen festgelegt.

6.2.2 aspHeute

erste Tests

Die Idee von aspHeute [Wil06] wurde aufgegriffen und auf Basis von PHP und Java nachgestellt, um ein Gefühl für das Vergleichen von Texten zu bekommen. (Abb. 6.1)

Dabei wurde für Tests vorrangig auf kurze und dokumentierte Texte von [Löb06] zurückgegriffen.

Kritik

Es stellten sich jedoch bald Kritikpunkte an diesem Verfahren heraus (Listing 6.1):

- Aufgrund der exzessiven Verwendung von Rekursion — jeder Vergleich von zwei nicht übereinstimmenden Buchstaben führt zu einem neuen Methodenaufruf — traten bei Texten schnell Probleme mit dem Speicherverbrauch im Stack auf.
- Die „gierige“ Vorgangsweise versucht möglichst schnell, identische Buchstaben zu finden. Dabei wird in den Texten aber immer nur vorwärts gesucht und z.B. vertauschte Sätze bleiben unberücksichtigt.

6 technischer Teil

- Zurückgegeben wird nur eine einzige Zahl, die die Anzahl an Übereinstimmungen repräsentiert. Das lässt keine Rückschlüsse auf den Ort der kopierten Stellen zu.

Listing 6.1: Rekursive Funktion in Algorithmus aspHeute

```
1  /**
2   * Rekursive Methode zum Finden übereinstimmender Buchstanketten.
3   * Testphase, ohne Gewähr auf Funktion!
4   *
5   * Paramter:
6   * - Startpunkt in Text 1
7   * - Startpunkt in Text 2
8   * - straight hat keine Bedeutung
9   *
10  * self::$offeneCalls misst die Größe des Stacks.
11  */
12  function stepLCS($t1offSet, $t2offSet, $straight = false) {
13      self::$offeneCalls++;
14      echo 'Aufruf_Nr.' . ($this->aufrufe++) . 't1:' . $t1offSet . 't2:' .
          $t2offSet . 'Calls:' . self::$offeneCalls;
15
16      if (($t1offSet >= $this->t1length || $t2offSet >= $this->t2length)) {
17          echo 'draussen<br />';
18          self::$offeneCalls--;
19          return 0;
20      } else {
21          if (!isset($this->table[$t1offSet][$t2offSet])) { // || $straight) {
22              echo ' <br />';
23              //echo ' nicht besucht';
24              if ($this->t1text {
25                  $t1offSet }
26                  == $this->t2text {
27                      $t2offSet }) {
28                  //echo ' Zeichen ident<br />';
29                  //echo ' Zeichen: ' . $this->t1text{$t1offSet} . '<br />';
30                  $this->table[$t1offSet][$t2offSet] = 1 + $this->stepLCS(
                      ($t1offSet + 1, $t2offSet + 1, true);
31              } else {
32                  //echo ' Zeichen nicht ident<br />';
33                  $this->table[$t1offSet][$t2offSet] = max($this->stepLCS(
                      ($t1offSet + 1, $t2offSet), $this->stepLCS(
                      $t1offSet, $t2offSet + 1));
34              }
35              self::$offeneCalls--;
36              return $this->table[$t1offSet][$t2offSet];
37          } else {
38              $self->offeneCalls--;
39              return 0;
40          }
41      }
42  }
```

Jeder einzelne war ein Ausschlusskriterium für den angestrebten Einsatz und führte zu einer Neuimplementierung in Java auf Basis des „longest common string“ [Wik08].

6.3 Neuimplementierung in Java: YAPLAF

Die Neuimplementierung hatte drei Ziele:

- Es sollten tatsächlich **alle** übereinstimmenden Stellen (common Strings) gefunden werden.
- Die Informationen über gefundene Stellen sollten erhalten bleiben.
- Der Speicherverbrauch sollte möglichst klein gehalten werden.

Nach einer Idee von Herrn Dipl. Ing. Michael Tesar wurde das Projekt

„Yet Another PLAGiarism Finder (kurz YAPLAF)“

benannt. Betrachtet wird Version 0.9.2 der Software.

Der Workflow sieht vor, dass die Abgaben aus einer XML Datei eingelesen, verarbeitet und verdächtige Werte in einer komma-separierten (csv) Datei ausgegeben werden. Die aktuell bearbeiteten Abgaben werden sowohl auf der Konsole als auch in eine Datei ausgegeben (Listing 6.3). Die Konfiguration des Programms wird in einer Datei im Java Properties Format im Java Classpath vorgenommen. Listing 6.2 zeigt eine beispielhafte Konfiguration:

Listing 6.2: Die Konfigurationsdatei YAPLAF.conf

```

1  ### YAPLAF Konfigurationsdatei
2
3  # Name der XML Datei, die eingelesen werden soll
4  YAPLAF_XML_filename = submission_ss07.xml
5
6  # Prozentsatz des LCS vom kürzeren Text,
7  # ab dem der Vergleich in Ausgabedatei abgelegt werden soll
8  YAPLAF_Prozent_Exportschwelle = 20
9
10 # Dateiname für Ausgabe – %n ist Platzhalter für eine fortlaufende Zahl
11 CSVWriter_export_filename = ausgabe%n.csv
12 # Anzahl der Einträge in einer Datei
13 CSVWriter_zeilen_Datei = 10000
14
15 # Mindestlänge für Strings, um abgelegt zu werden
16 LCSVektor_minStringlaenge = 7

```

Listing 6.3: Die Ausgabe von YAPLAF auf der Konsole und in die log-Datei

```

1  Datei submission_ss07.xml erfolgreich geöffnet
2  Es existieren 379 Abgaben.
3  Es werden 71631 Vergleiche durchgeführt.
4  Setze LCSSchranke auf 7
5  aktuelle Zeit: 16:16:09 CET
6
7  Durchlauf 1 von 71631
8  Vergleiche 000001 mit 000002: 126 Länge1: 1698 (7%) Länge2: 1482 (8%)

```

6 technischer Teil

```
9 Durchschnittliche Laufzeit: 156ms
10 Ende voraussichtlich um 19:22:23 CET
11 EinfuegeOperationen: 27 0 0
12
13 Durchlauf 2 von 71631
14 Vergleiche 000001 mit 000003: 265 Länge1: 1698 (15%) Länge2: 1382 (19%)
15 Durchschnittliche Laufzeit: 133ms
16 Ende voraussichtlich um 18:54:56 CET
17 EinfuegeOperationen: 48 0 0
```

6.3.1 Texte einlesen und vorbereiten

XML von Moodle

Das hinter TUWEL stehende Moodle ist in der Lage, Backups für einzelne Abgaben in XML zu erstellen, die dem Programm wiederum als Eingangsdaten dienen. Zu diesem Zweck wurde ein SAX Parser erstellt. Listing 6.4 zeigt die drei wesentlichen Methoden `startElement`, `endElement` und `characters` und wie daraus Objekte der Klasse `Abgabe` zur weiteren Verwendung erstellt werden.

Listing 6.4: die 3 wesentlichen Methoden des SAX Parsers

```
1 public void startElement(String namespaceURI, String localName,
2 String qualifiedName, Attributes atts) throws SAXException {
3     if (!this.isUser) {
4         if (qualifiedName.equals(this.userStr)) {
5             this.count++;
6             this.isUser = true;
7             this.temp = new Abgabe();
8             // System.out.println("Benutzer start");
9         }
10    } else if (qualifiedName.equals(this.vornameStr)) {
11        this.isVorname = true;
12    } else if (qualifiedName.equals(this.nachnameStr)) {
13        this.isNachname = true;
14    } else if (qualifiedName.equals(this.matrikelStr)) {
15        this.isMatrikelNr = true;
16    } else if (qualifiedName.equals(this.dataStr)) {
17        this.isStudentdata = true;
18        puffer = "";
19    } else if (qualifiedName.equals(this.abgabezeitStr)) {
20        this.isAbgabezeit = true;
21    }
22 }
23
24 public void endElement(String namespaceURI, String localName,
25 String qualifiedName) throws SAXException {
26     if (this.isUser) {
27         if (qualifiedName.equals(this.userStr)) {
28             this.isUser = false;
29             //Leere Eintraege duerfen nicht uebernommen werden.
30             if (temp.getMatrikelNr() != 0 && temp.getAbgabeText().length()
31                 > 0)
```

6 technischer Teil

```
31         abg.add(temp);
32         // System.out.println("Benutzer ende");
33     } else if (qualifiedName.equals(this.vornameStr)) {
34         this.isVorname = false;
35     } else if (qualifiedName.equals(this.nachnameStr)) {
36         this.isNachname = false;
37     } else if (qualifiedName.equals(this.matrikelStr)) {
38         this.isMatrikelNr = false;
39     } else if (qualifiedName.equals(this.dataStr)) {
40         temp.setStudentdata(this.puffer);
41         this.isStudentdata = false;
42     } else if (qualifiedName.equals(this.abgabezeitStr)) {
43         this.isAbgabezeit = false;
44     }
45 }
46
47
48 public void characters(char[] ch, int start, int length)
49     throws SAXException {
50     if (this.isUser) {
51         if (this.isMatrikelNr) {
52             String tempString = new String(ch, start, length);
53             try {
54                 temp.setMatrikelnr(new Integer(tempString));
55             } catch (Exception e) {
56                 System.out.println("TUWELSaxHelper: _Matrikelnummer_"
57                     + tempString + " _ist _keine _Ganzzahl.");
58                 temp.setMatrikelnr(0);
59             }
60         } else if (this.isVorname) {
61             temp.setVorname(new String(ch, start, length));
62         } else if (this.isNachname) {
63             temp.setNachname(new String(ch, start, length));
64         } else if (this.isStudentdata) {
65             this.puffer += new String(ch, start, length);
66         } else if (this.isAbgabezeit) {
67             temp.setAbgabezeit(new String(ch, start, length));
68         }
69     }
70 }
```

Leerzeichen, Umlaute

Die eingelesenen Abgaben werden danach von verbliebenen XML Tags und XML Entities, Leerzeichen und Zeilenschaltungen befreit. Umlaute werden durch ASCII Zeichen ersetzt und Grossbuchstaben werden in Kleinbuchstaben umgewandelt.

Das entfernen von XML Tags kann notwendig sein, wenn der in Moodle integrierte WYSIWYG Editor verwendet worden ist.

6.3.2 Abgaben verarbeiten und Common Strings suchen

Anschliessend werden die Abgaben paarweise der Komponente zum Finden der Common Strings übergeben und der Algorithmus gestartet (Listing 6.5).

Listing 6.5: die doppelte for Schleife vergleicht die Texte paarweise

```

1 // Fuer jede Kombination von zwei Abgaben...
2 for (int i = 0; i < abgaben.size(); i++) {
3     // "Quelltext" übergeben
4     lcs.setSrc(abgaben.get(i).getAbgabeText());
5     for (int j = i + 1; j < abgaben.size(); j++) {
6         // "Zieltext" übergeben
7         lcs.setDst(abgaben.get(j).getAbgabeText());

```

Basierend auf dem Ansatz des „longest common substring“ [Wik08] wurde ein Algorithmus entwickelt, der nicht nur **die** längste gemeinsame Zeichenkette sucht, sondern **alle** gemeinsamen Zeichenketten („common Strings“), die einem einfachen Kriterium — der Mindestlänge — genügen.

Dazu wird ein 2-dimensionales Array, dessen erste Dimension („Zeile“) der Länge eines Textes und die zweite Dimension („Höhe“) 2 entspricht, gebildet und mit 0 initialisiert. Die Höhe von 2 ist dadurch möglich, weil bei den zeichenweisen Vergleichen im Fall einer Übereinstimmung nur auf Werte der Zeichen davor, also dem diminuierten Wert des Index der zweiten Dimension zugegriffen werden muss. Der Zähler der zweiten Dimension kann normal über eine for-Schleife hoch zählen, der Index wird über die Operation Modulo 2 errechnet.

Mit 2 for Schleifen wird dann über die einzelnen Buchstaben der Texte iteriert. Wenn zwei Buchstaben identisch sind, wird über die aktuellen Iteratoren aus dem Array (der Index in der 2. Dimension ist Modulo 2) der Vergleichswert der in den Strings unmittelbar davor liegenden Buchstaben gelesen, um 1 erhöht und wiederum in dem aktuellen Feld gespeichert. Andernfalls wird das aktuelle Feld nicht verändert, da es schon mit 0 initialisiert worden ist.

Fallbeispiel

Ein Fallbeispiel soll zur Veranschaulichung dienen: Es werden dazu die zwei kurzen Texte „Ein Hippi im Mississippi“ und „Ein Mississippihippi“ verwendet.

Abbildung 6.2 zeigt das diagonale „Fortpflanzen“ der übereinstimmenden Textstellen. Die Buchstaben der Texte sind auf den Achsen unter und neben den Indices aufgetragen. Die Quadrate deuten die nachfolgende Generierung eines MatchingStrings und eine Einfügeoperation an. (siehe auch Listing 6.6)

Listing 6.6: die Zeichenvergleiche in der Klasse LCSVektor

```

1 // Array mit 2 * Laenge von dst mit 0 initialisieren
2 // uber modulo 2 kann dann auf die aktuelle und vorige Zeile zugegriffen
3 // werden -> spart Speicher
4 this.table = new int[2][this.dstlength + 1];

```

6 technischer Teil

	A	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21
B	e	i	n	h	i	p	p	i	e	i	m	m	i	s	s	i	s	s	i	p	p	i	
00	e	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
01	i	0	2	0	0	1	0	0	1	0	2	0	0	1	0	0	1	0	0	1	0	0	1
02	n	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
03	m	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
04	i	0	1	0	0	1	0	0	1	0	1	0	0	2	0	0	1	0	0	1	0	0	1
05	s	0	0	0	0	0	0	0	0	0	0	0	0	0	3	1	0	2	1	0	0	0	0
06	s	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4	0	1	3	0	0	0	0
07	i	0	1	0	0	1	0	0	1	0	1	0	0	1	0	0	5	0	0	4	0	0	1
08	s	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	6	1	0	0	0	0
09	s	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0	1	7	0	0	0	0
10	i	0	1	0	0	1	0	0	1	0	1	0	0	1	0	0	4	0	0	8	0	0	1
11	p	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	9	1	0	0
12	p	0	0	0	0	0	1	3	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0
13	i	0	1	0	0	1	0	0	4	0	1	0	0	1	0	0	1	0	0	1	0	0	11
14	h	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	i	0	1	0	0	2	0	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1
16	p	0	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0
17	p	0	0	0	0	0	1	4	0	0	0	0	0	0	0	0	0	0	0	1	3	0	0
18	i	0	1	0	0	1	0	0	5	0	1	0	0	1	0	0	1	0	0	1	0	0	4
19	e	1	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Abbildung 6.2: die Matrix der Übereinstimmungen

```

5      for (int i = 0; i < 2; i++) {
6          for (int j = 0; j <= this.dstlength; j++) {
7              this.table[i][j] = 0;
8          }
9      }
10     // Buchstaben fuer Buchstaben vergleichen
11     for (int i = 0; i < this.srclength; i++) {
12         for (int j = 0; j < this.dstlength; j++) {
13             // Fortschrittanzeige
14             if ((zaehlVergleiche++ % modErwartetVergleiche) == 0) {
15                 System.out.print(".");
16             }
17             // den Vorgängervergleichswert aus der Tabelle holen
18             vorgaenger = this.table[i % 2][j];
19             // Wenn die Buchstaben uebereinstimmen, Feld um 1 hochzaehlen
20             if (src.charAt(i) == dst.charAt(j)) {
21                 aktuell = vorgaenger + 1;
22                 // aktuellen Wert in die Tabelle schreiben
23                 this.table[(i + 1) % 2][j + 1] = aktuell;
24             }
25             // Ansonsten pruefen, ob der gefundene Substring eingefuegt
26             // werden soll und den aktuellen Wert zuruecksetzen!!!
27             else {
28                 aktuell = 0;
29                 // Wenn der Vorgaenger groesser als der gesetzte
30                 // Schwellwert
31                 // ist
32                 if (vorgaenger > this.schranke) {
33                     zaehlImFeld++;
34                     // in den MatchingStringContainer einfuegen:
35                     // Startpunkt1, Startpunkt2, Laenge

```

„Start in“ „A“	„B“	Länge	toString
0	0	3	ein
15	4	4	issi
12	7	4	issi
11	3	11	mississippi
4	10	4	ippi
18	15	4	ippi
3	14	6	hippie

Tabelle 6.1: Aus Abbildung 6.2 generierte MatchingStrings

```

35         container.insert(new MatchingString(i -
36             vorgaenger, j - vorgaenger, vorgaenger));
37     }
38 }
39 // Am Ende jeder "Zeile" die letzte "Spalte" ueberpruefen
40 if (aktuell > this.schranke) {
41     zaehlSpalte++;
42     container.insert(new MatchingString(i - vorgaenger, dstlength -
43         vorgaenger - 1, vorgaenger + 1));
44 }
45 // Die letzte "Zeile" durchlaufen und auf common Strings pruefen
46 int i = this.srclength % 2;
47 for (int j = 1; j < this.dstlength; j++) {
48     vorgaenger = this.table[i][j];
49     if (vorgaenger > this.schranke) {
50         zaehlZeile++;
51         container.insert(new MatchingString(srclength - vorgaenger, j -
52             vorgaenger, vorgaenger));
53     }
54 }

```

6.3.3 Definition von „MatchingString“

Für die Übereinstimmungen wurde der Begriff „MatchingString“ geprägt. Ein MatchingString repräsentiert dabei identische Textstellen in **zwei** Texten.

Ein MatchingString ist durch Startpunkte in zwei Texten (Positionen als Ganzzahl) und der Länge (Ganzzahl) definiert. Tabelle 6.1 zeigt alle gefundenen common Strings, die eine Mindestlänge von 3 Buchstaben haben und nun als MatchingStrings dargestellt werden.

Die Werte eines MatchingStrings können nach der Instanzierung nicht mehr verändert werden (Konstruktor in Listing 6.7). Die aktuelle Implementierung kann zu einer linearen Liste (ohne Iteratoren) verkettet werden und hat eine eigene Methode zum Entfernen aus dieser Liste, um das Löschen aus einem MatchingStringContainer zu beschleunigen

6 technischer Teil

(Listing 6.8). Sie verfügt auch über Methoden, die Überschneidungen und die Lage zu einem anderen MatchingString – jeweils in einem Text!!! – bestimmen können (Listing 6.9)

Listing 6.7: Der Konstruktor der Klasse MatchingString

```
1 public MatchingString(int start1, int start2, int length) {
2     assert start1 >= 0: "start1_ist_kleiner_0_(Grund_liegt_außerhalb_des_
3         Konstruktors!!!)";
4     this.start[0] = start1;
5
6     assert start2 >= 0: "start2_ist_kleiner_0_(Grund_liegt_außerhalb_des_
7         Konstruktors!!!)";
8     this.start[1] = start2;
9
10    assert length > 0: "length_ist_nicht_größer_0_(Grund_liegt_außerhalb_des_
11        Konstruktors!!!)";
12    this.length = length;
13
14    this.end[0] = this.start[0] + this.length; //calculate end0
15    this.end[1] = this.start[1] + this.length; //calculate end1
16
17    // this object is not part of a linear list
18    this.next = null;
19    this.prev = null;
20
21    this.active = true; //this object is set to active
22 }
```

Listing 6.8: Die Methode unbind() der Klasse MatchingString zum lösen aus der linearen Liste

```
1 void unbind() {
2     //Wenn dieser MatchingString Teil einer Verketteten List ist
3     if(this.active) {
4         //die Zeiger der unmittelbaren Nachbarn ändern
5         if (this.prev != null) this.prev.next = this.next;
6         if (this.next != null) this.next.prev = this.prev;
7         this.active = false;
8     }
9 }
```

Listing 6.9: Die Methode overlap(MatchingString other, int text) der Klasse MatchingString

```
1 public boolean overlap(MatchingString other, int text) {
2     //Testet, ob sich zwei MatchingStrings in Text text überschneiden
3     //Fall 1: Wenn other.end[text] in this liegt
4     if(other.end[text] > this.start[text] && other.end[text] <= this.end[text]) {
5         return true;
6     }
7     //Fall 2: Wenn other.start[text] in this liegt
```

MatchingString			Ergebnis	drops	Begründung
0	0	3	true	–	keine Überschneidung
15	4	4	true	–	keine Überschneidung
12	7	4	false	–	Überschneidung in Text A und kein zusätzlicher Nutzen
11	3	11	true	15, 4, 4	Überschneidung in beiden Texten aber größere Gesamtlänge
4	10	4	false	–	Überschneidung in Text B und kein zusätzlicher Nutzen
18	15	4	false	–	Überschneidung in Text A und kein zusätzlicher Nutzen
3	14	6	true	–	keine Überschneidung

Tabelle 6.2: Die Einfügeoperationen nach Tabelle 6.1

```

8         if (other.start[text] >= this.start[text] && other.start[text] < this.end[text])
9             {
10                 return true;
11             }
12             // Fall 3: Wenn this komplett in other liegt
13             if (other.start[text] <= this.start[text] && other.end[text] >= this.end[text])
14                 {
15                     return true;
16                 }
17             // Andernfalls gibt es keine Überschneidung
18             return false;
19     }

```

6.3.4 die Datenstruktur MatchingStringContainer

Beim Suchen nach gemeinsamen Strings werden tatsächlich alle Übereinstimmungen mit der erforderlichen Länge gefunden, unabhängig davon, ob sich Übereinstimmungen mit anderen Übereinstimmungen überlappen. Das könnte ohne Filtermechanismus beim Summieren der Länge der einzelnen Übereinstimmungen am Ende des Durchlaufs einen Wert ergeben, der größer ist als die eigentliche Textlänge. Diese Tatsache lässt sich anhand Abbildung 6.2 nachvollziehen: die MatchingStrings A:4, B:10, 4 und A:3, B:14, 6 überschneiden sich.

MatchingStrings müssen also nicht nur gespeichert, neu gefundene MatchingString dürfen auch nur bedingt eingefügt werden. Es gilt die Bedingung, dass ein MatchingString dann eingefügt wird, wenn er eines dieser Kriterien erfüllt:

- er überschneidet sich mit keinem vorhandenen MatchingString
- seine Länge ist größer als die Summe der Längen aller MatchingStrings, mit denen er sich überlappt

Dieses Verhalten soll zu einer Maximierung der Längen der im MatchingStringContainer gespeicherten MatchingStrings führen. Tabelle 6.2 veranschaulicht das Verhalten.

Einfügeoperation

Wenn ein MatchingString E an den MatchingStringContainer übergeben wird, berechnet dieser zuerst die Länge aller MatchingStrings im Container, die sich mit E überschneiden (Listing 6.10). Sollte dieser Wert kleiner sein, als die Länge von E, werden alle überlappenden MatchingStrings aus dem Container entfernt und E eingefügt. Umgekehrt wird E verworfen und der Inhalt des Containers nicht verändert (Listing 6.11).

Nach allen Einfügeoperationen sind diese MatchingStrings im Container gespeichert:

- 0, 0, 3: ein
- 11, 3, 11: mississippi
- 3, 14, 6: hippie

Listing 6.10: Die Methode calculateOverlappingStringLength() der Klasse MatchingStringContainer

```

1  private void calculateOverlappingStringLength() {
2      // Variablen zur Kalkulation zurücksetzen
3      this.calculatedOverlappingStringLength[0] = 0;
4      this.calculatedOverlappingStringLength[1] = 0;
5      this.calculatedOverlappingStringLengthUnique = 0;
6      // Bool'sche Variable um festzulegen, ob ein MatchingString in eine
7      // temporäre Liste aufgenommen wird.
8      boolean isOverlapping = false;
9      // Starten mit dem Kopf der Liste
10     this.matchingStringCurrent = this.linlist;
11     // Solange ein nächstes Element existiert
12     while (this.matchingStringCurrent != null) {
13         // Wenn sich die MatchingStrings in Text 0 überlappen
14         if (this.matchingStringCurrent.overlap(this.matchingStringElement, 0))
15         {
16             this.calculatedOverlappingStringLength[0] += this.
17                 matchingStringCurrent.length;
18             isOverlapping = true; //
19         }
20         // Wenn sich die MatchingStrings in Text 1 überlappen
21         if (this.matchingStringCurrent.overlap(this.matchingStringElement, 1))
22         {
23             this.calculatedOverlappingStringLength[1] += this.
24                 matchingStringCurrent.length;
25             isOverlapping = true;
26         }
27         // wenn isOverlapping wahr ist, wird this.matchingStringCurrent die
28         // temporäre Liste eingefügt.
29         if (isOverlapping) {
30             this.temp.add(this.matchingStringCurrent);
31             this.calculatedOverlappingStringLengthUnique += this.
32                 matchingStringCurrent.length;
33             isOverlapping = false;
34         }
35         // Zeiger auf das nächste Element setzen

```

```

31         this.matchingStringCurrent = this.matchingStringCurrent.next;
32     }
33     return;
34 }

```

Listing 6.11: Die Methode insertOperation() der Klasse MatchingStringContainer

```

1     private void insertOperation() {
2         //Für jeden MatchingString in der temporären Liste
3         while (!this.temp.isEmpty()) {
4             //oberstes Element aus der Liste holen
5             this.matchingStringCurrent = this.temp.poll();
6             //Sollte das aktuelle Element der Kopf der Liste sein
7             //ist eine Sonderbehandlung notwendig
8             if (this.matchingStringCurrent == this.linlist) {
9                 this.linlist = this.matchingStringCurrent.next;
10            }
11            this.history += "###-MatchingString_geloescht_###\r\n" + this.
                matchingStringCurrent.toString();
12            //Element aus der Liste löschen
13            this.matchingStringCurrent.unbind();
14        }
15        // den einzufügenden MatchingString bearbeiten
16        if (this.linlist != null) {
17            this.matchingStringElement.next = this.linlist;
18            this.linlist.prev = this.matchingStringElement;
19        }
20        this.linlist = this.matchingStringElement;
21    }

```

Es hat sich bei Tests herausgestellt, dass der MatchingStringContainer die zeitkritische Komponente des Programms ist. Die aktuelle Implementierung der Klasse MatchingStringContainer speichert MatchingStrings unsortiert in einer linearen Liste. Es ist deshalb notwendig, bei jeder Suche nach Überschneidungen, alle vorhandenen MatchingStrings zu überprüfen. Eine Implementierung mit 2 sortierten Listen – jeweils eine pro Text – könnte hier die Suche früher abbrechen und zu einer Verbesserung der Performance führen. Alternativ wäre auch eine eigene Datenstruktur ähnlich einer Hashmap mit „Einstiegspunkten“ in die Liste in regelmäßigen Abständen denkbar.

6.3.5 Ergebnisse ausgeben und abspeichern

Die Ausgabe erfolgt als Datei mit kommaseparierten Werten (vgl. csv: comma separated values) ausgegeben. Jeder Vergleich von zwei Texten wird in einer eigenen Zeile ausgegeben. Um die Menge an Datensätzen einzuschränken, kann in der Konfigurationsdatei (Listing 6.2) ein Prozentwert (Summe der übereinstimmenden Zeichen in Relation zur Textlänge) gesetzt werden, ab dem Vergleiche festgehalten werden sollen.

Pro Datensatz werden diese Felder ausgegeben:

- Daten von Student 1: Matrikelnummer, Name und Länge von Text 1.
- Quotient aus der Summe der übereinstimmenden Zeichen zur Länge von Text 1.

- Daten von Student 2: Matrikelnummer, Name und Länge von Text 2.
- Quotient aus der Summe der übereinstimmenden Zeichen zur Länge von Text 2.
- Die Summe der übereinstimmenden Zeichen (für beide Text ident).
Ein großer Wert kann ein Hinweis auf eine Kopie sein.
- Die Anzahl der gefundenen MatchingStrings.
Hier ist ein kleiner Wert ein Hinweis.

Aus der Kombination von Quotient, der Summe und der Anzahl der MatchingStrings lassen sich Rückschlüsse auf eine vermutliche Kopie ziehen.

6.4 Kritik

Der Einsatz von YAPLAF in der Praxis hat gezeigt, dass alleine das Vorhandensein eines Programms das Bewusstsein zum korrekten Arbeiten schaffen kann.

Wenn bei der ersten Abgabe im Semester 1:1 Kopien noch „normal“ sind, so werden diese gegen Ende des Semesters immer seltener und die Erkennungsrate sinkt. Auch wenn YAPLAF hinreichend funktioniert, ist eine manuelle Analyse leider aber nach wie vor notwendig.

Das Erkennen von Kopien fällt einem Menschen vergleichsweise leicht. Man hat ein „Gefühl“ dafür, einen Text schon einmal so ähnlich gelesen zu haben. Einem Computerprogramm dieses „Gefühl“ anzulernen, ist sehr schwer bis unmöglich.

Trotzdem ist das Ziel dieser Software eine Klassifizierung, um dem Lehrenden eine einfache Übersicht zu geben und auf vermutlich kopierte Abgaben hinzuweisen.

Es soll nun anhand von Beispielen hinterfragt werden, ob YAPLAF die gestellten Aufgaben erfüllen kann.

6.4.1 Arten von Kopien

1:1 Kopie

Der vergleichsweise einfache Fall einer direkten Kopie wird von YAPLAF per Definition als 1 MatchingString mit 100% Übereinstimmung erkannt.

Vertauschen oder verändern einzelner Worte

Durch das Verwenden von Synonymen oder das sinnerhaltende Vertauschen der Reihenfolge einzelner Wörter oder Sätze kann der Text umgestaltet werden.

Auch diesen Fall soll YAPLAF hinreichend gut erkennen, da die Reihenfolge der Wörter und Sätze in den Texten keine Rolle spielen darf. Das Umformen oder Ersetzen von wenigen Wörtern senkt zwar die Summe an erkannten Übereinstimmungen, sollte aber bei Verwendung einer geeigneten Schranke (z.B. ab 60% Übereinstimmung) korrekt erkannt werden. Eine geringe Anzahl der MatchingStrings ist ein weiteres Indiz für eine Kopie.

Kopie aus mehreren Quellen

Wenn die Abgabe aus 2 oder mehr anderen Abgaben zu gleichen Teilen zusammengesetzt worden ist, besteht faktisch keine Chance für YAPLAF in der derzeitigen Version, das auch zu erkennen. Hier können eventuell statistische Auswertungen der gefundenen MatchingStrings oder auch der Vergleich mit mehreren Abgaben in zukünftigen Entwicklungen behilflich sein. (siehe Sektion 6.5)

6.4.2 Testfälle

Im Zuge der Erstellung des Programms sind 2 Klassen zum automatischen Testen der Klassen MatchingString und MatchingStringContainer entstanden.

MatchingString

Die Testklasse instanziiert mehrere MatchingStrings und überprüft die Methode overlap. Sie trifft Annahmen, ob sie sich überlappen und überprüft diese Annahmen. (Listing 6.12)

Listing 6.12: Klasse zum automatischen Testen der Methode overlap

```

1  package TEST;
2
3  import YAPLAF.LCSV.*;
4  import java.io.*;
5
6  public class MatchingStringTestAutomat {
7      /**
8       * @param args
9       */
10     public static void main(String[] args) {
11         PrintStream out = System.out;
12         //Referenz String & Referenz MatchingString
13         String refString = "abcdefghijklmnopqrstuvwxyz";
14         out.println("Länge:␣" + refString.length());
15         out.println(refString);
16         out.println();
17
18         int start0;
19         int start1 = 0;
20         int laenge;
21         int startRef0 = 2;
22         int startRef1 = 0;
23         int laengeRef = 4;
24         MatchingString refMatch = new MatchingString(startRef0 ,startRef1 , laengeRef);
25         //Referenz
26         out.print("Referenz:␣");
27         String temp = refString.substring(startRef0 , startRef0 + laengeRef);
28         out.println(refMatch + "␣-␣" + temp);
29         out.println();

```

6 technischer Teil

```
30     int count = 0;
31     boolean vorhersage;
32     boolean tatsache;
33     int fehler = 0;
34     /*
35      * Für jede Möglichkeit an Start und Endpunkt
36      * kleinster Startpunkt: 0, kleinster Endpunkt: 1 (Länge 1)
37      * größter Startpunkt: 8, größter Endpunkt: 9 (Länge 1)
38      */
39     int startpunktMax = 8;
40     //Startpunkte zwischen 0 und 8 setzen
41     for (int i = 0; i < startpunktMax; i++) {
42         //Endpunkte zwischen 1 und 9
43         for (int j = i + 1; j < startpunktMax + 1; j++) {
44             /*
45              * Für jedes Wertepaar erstelle MatchingString
46              */
47             count++;
48             start0 = i;
49             laenge = j-i;
50             //Referenz
51             MatchingString test = new MatchingString(start0, start1, laenge
52             );
53             out.print("Fall␣" + count + ":\n");
54             out.println(test + "␣_" + refString.substring(start0, start0 +
55             laenge) + "␣(" + refString.substring(startRef0, startRef0
56             + laengeRef) + ")");
57             /*
58              * Erstelle die Vorhersage...
59              * mit Methode overlap
60              */
61             out.print("Vorhersage:\n");
62             if(vorhersage = refMatch.overlap(test, 0)) { //Fehler
63                 out.println("überlappen");
64             }
65             else {
66                 out.println("NICHT_überlappen");
67             }
68             /*
69              * Testprozedur
70              */
71             tatsache = false;
72             out.print("Tatsache:\n");
73             for (int z = i; z < j; z++) {
74                 if (temp.indexOf(refString.charAt(z)) != -1) {
75                     tatsache = true;
76                 }
77             }
78             if(tatsache) {
79                 out.println("überlappen");
80             }
81             else {
82                 out.println("NICHT_überlappen");
83             }
84             if(vorhersage == tatsache) {
```

```

83         out.println("OK");
84     }
85     else {
86         out.println("FEHLER");
87         fehler++;
88     }
89 }
90 }
91 out.println();
92 out.println("Anzahl_Fehler: " + fehler);
93 }
94 }

```

MatchingStringContainer

Die Testklasse generiert mit einem Zufallsgenerator mehrere MatchingStrings und überprüft die Methode insert. Sie trifft Annahmen, ob sie eingefügt werden sollen und überprüft diese Annahme. (Listing 6.13)

Listing 6.13: Klasse zum automatischen Testen der Methode insert

```

1  package TEST;
2
3  import YAPLAF.LCSV.*;
4
5  public class MatchingStringContainerTestAutomat {
6      /**
7       * Idee :
8       *
9       * Es werden mit Math.Random Zufallszahlen und damit MatchingStrings erzeugt
10      *
11      * Anhand der schon geprüften Methode MatchingString.overlap wird ein neue
12      * Länge des LCSVektors vorhergesagt
13      *
14      * @param args
15      */
16      @SuppressWarnings("deprecation")
17      public static void main(String[] args) {
18
19          int startmaximum = 5000;
20          int laengemaximum = 500;
21          int anzdurchlaeufe = 10000;
22
23          int start0 = 0;
24          int start1 = 0;
25          int laenge = 0;
26          int laengemax = 0;
27
28          int LCSVektorLaenge = 0;
29
30          MatchingString ms;
31          MatchingString tempMS;
32          MatchingStringContainer msc = new MatchingStringContainer();
33          String achtungString = "";

```

6 technischer Teil

```
34
35     int tempLaenge0;
36     int tempLaenge1;
37     int calcLaenge;
38     boolean achtung = false;
39     /*
40      * Zufallszahlen erzeugen: - start (2mal!) mit nextInt (gleichverteilt)
41      * - laenge mit nextGaussian (Normalverteilt um vor allem kürzere
42      * MatchingStrings zu simulieren
43      */
44     java.io.PrintStream out = System.out;
45     java.util.Random r = new java.util.Random();
46     for (int i = 0; i < anzdurchläufe; i++) {
47         tempLaenge0 = 0;
48         tempLaenge1 = 0;
49         calcLaenge = 0;
50         start0 = r.nextInt(startmaximum);
51         start1 = r.nextInt(startmaximum);
52         laenge = 1 + (int) (Math.abs(r.nextGaussian()) * (laengemaximum / 5));
53         // Annahme:
54
55
56
57         if (laenge > laengemax) {
58             laengemax = laenge;
59         }
60         ms = new MatchingString(start0, start1, laenge);
61         /*
62          * Überlappungen und neue Länge des LCSVektor berechnen
63          */
64         boolean eins;
65         boolean zwei;
66
67         msc.resetPointer();
68         while (msc.hasNextMatchingString()) {
69             tempMS = msc.nextMatchingString();
70             /*
71              * Überlappungen berechnen (Wir gehen davon aus, dass overlap
72              * korrekt arbeitet!!!)
73              */
74             if (eins = ms.overlap(tempMS, 0)) {
75                 tempLaenge0 += tempMS.getLength();
76                 // tempList0.add(tempMS);
```

6 technischer Teil

```
77         }
78         if (zwei == ms.overlap(tempMS, 1)) {
79             tempLaenge += tempMS.getLength();
80             // tempList.add(tempMS);
81         }
82         if (eins || zwei) {
83             // tempList.add(tempMS);
84             calcLaenge += tempMS.getLength();
85         }
86     }
87     /*
88     * Bestimmen, ob eingefügt werden soll oder nicht ACHTUNG!!! Dieser
89     * Teil ist komplett ident mit Original
90     */
91     if (ms.getLength() > calcLaenge) {
92         /*
93         * Einfügen simulieren... calcLaenge minus ms.length plus
94         */
95         if (calcLaenge > ms.getLength()) {
96             achtung = true;
97             achtungString += i + "_mslength_" + ms.getLength()
98                 + "_calcLaenge_" + calcLaenge + "\r\n";
99         }
100         LCSVektorLaenge = LCSVektorLaenge - calcLaenge + ms.getLength()
101             ;
102     }
103     msc.insert(ms);
104
105     out.println(i + "/" + anzdurchlaeufer + ":" + start0 + " " + start1 + "
106         " + laenge);
107     out.println("CalcLaenge:" + calcLaenge + "Vorhersage:" + msc.
108         getMatchingStringSum() + "Berechnung:" + LCSVektorLaenge);
109 }
110 out.println(laengemax);
111 if (achtung)
112     out.println(achtungString);
113 else
114     out.println("ALLES_OK");
115 }
```

6.5 Ausblick

Im Sommersemester 2008 hat ein komplettes Redesign von Yaplaf unter dem Titel „YAPLAF++“ statt gefunden. Das Ziel war eine auf Komponenten basierende Software, deren Domänenmodell in Abbildung 6.3 skizziert ist.

Die Aufgaben von YAPLAF sollen stark modularisiert und in mehrere Teilbereiche gegliedert werden:

- Einlesen der Daten aus wählbaren Quellen (z.B. XML, pdf, Klartext)

YAPLAF – die Zukunft?

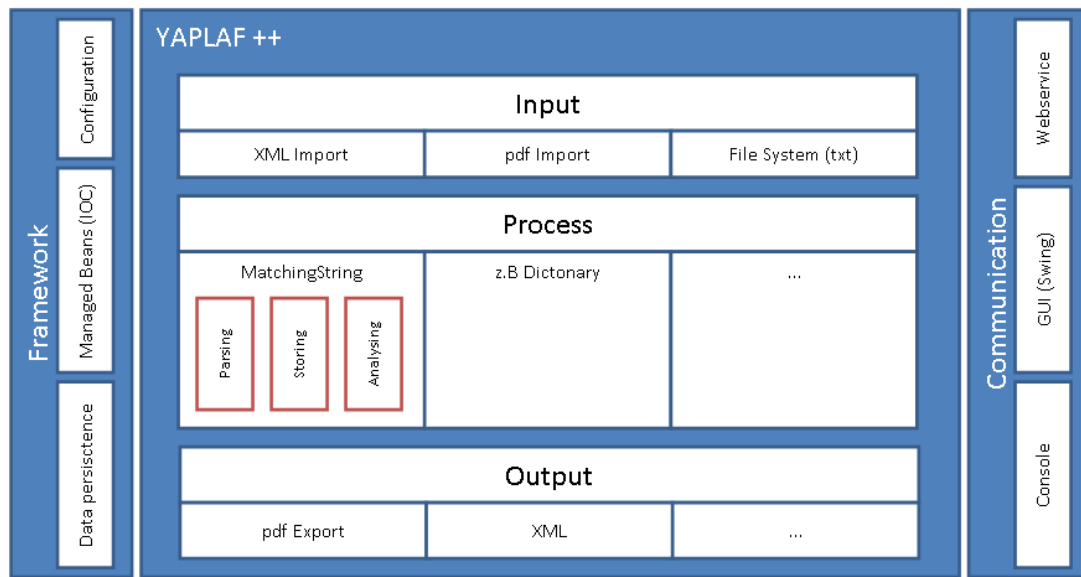


Abbildung 6.3: Mögliches Domänenmodell des zukünftigen YAPLAF

- Filtern der eingelesenen Texte (z.B. Leerzeichen entfernen)
- Verarbeiten der gefilterten Texte
- Analysieren der gewonnenen Daten
- Ausgabe der Ergebnisse in wählbaren Format (z.B. XML, pdf)

Alle Komponenten sollen als Plug-ins frei wählbar und konfigurierbar sein. So sollen z.B. aufeinander folgende Durchläufe mit veränderten Parametern oder auch andere Analysemethoden wie Wörterbuchanalysen möglich sein.

Die Aufgabe von YAPLAF++ ist hier die Bereitstellung eines Frameworks mit stabilen Schnittstellen.

Tabellenverzeichnis

2.1	Ghostwriting Anbieter	13
3.1	Plagiats Software [Sie07]	18
6.1	Aus Abbildung 6.2 generierte MatchingStrings	46
6.2	Die Einfügeoperationen nach Tabelle 6.1	48

Abbildungsverzeichnis

2.1	Babel Fish Uebersetzung	11
3.1	Turnitin	16
3.2	MyDropBox [Haf06]	17
3.3	Ephorus	17
5.1	Anzahl der Fälle - Aufteilung der Fälschungen nach Warengruppen [fFs07]	33
5.2	Fälschungen nach Warengruppen innerhalb der EU [Com07]	34
5.3	registrierte Fälle 1996-2006 [Com07]	34
5.4	Anzahl der Fälle - Herkunftsland gefälschter Artikel in Österreich [fFs07] .	36
5.5	Herkunftsland gefälschter Artikel in der EU [Com07]	36
6.1	Eingabeseite des ersten Prototypen in PHP	39
6.2	die Matrix der Übereinstimmungen	45
6.3	Mögliches Domänenmodell des zukünftigen YAPLAF	57

Listings

6.1	Rekursive Funktion in Algorithmus aspHeute	40
6.2	Die Konfigurationsdatei YAPLAF.conf	41
6.3	Die Ausgabe von YAPLAF auf der Konsole und in die log-Datei	41
6.4	die 3 wesentlichen Methoden des SAX Parsers	42
6.5	die doppelte for Schleife vergleicht die Texte paarweise	44
6.6	die Zeichenvergleiche in der Klasse LCSVektor	44
6.7	Der Konstruktor der Klasse MatchingString	47
6.8	Die Methode unbind() der Klasse MatchingString zum lösen aus der linearen Liste	47
6.9	Die Methode overlap(MatchingString other, int text) der Klasse MatchingString	47
6.10	Die Methode calculateOverlappingStringLength() der Klasse MatchingStringContainer	49
6.11	Die Methode insertOperation() der Klasse MatchingStringContainer	50
6.12	Klasse zum automatischen Testen der Methode overlap	52
6.13	Klasse zum automatischen Testen der Methode insert	54

Literaturverzeichnis

- [Bf] Babel Fisch Übersetzer. Website. <http://lingobot.translated.net/site/german/babel/altavista/com/>.
- [Bra04] Sebastian Brant. *Das Narrenschiff*. Marix Verlag, Jan 2004.
- [Bus07] Rido Busse. Aktion Plagiarius. Website, 2007. http://www.plagiarius.com/aktion_plag.html.
- [Com07] European Commission. Statistics. Website, Jul 2007. http://ec.europa.eu/taxation_customs/customs/customs_controls/counterfeit_piracy/statistics/index_en.htm.
- [DP08] WU Wien PL/AK/MN/MS DUSS Projektteam. Copy and Paste Plagiate an der Universität. Website, 2008. http://www.wu-wien.ac.at/lehre/support/schreiben_zitieren/plagiate.
- [Dri06] Dana Lynn Driscoll. Quoting, Paraphrasing, and Summarizing. Website, Sep 2006. <http://owl.english.purdue.edu/owl/resource/563/01/>.
- [EKW03] Hannes Etzlsdorfer, Willibald Katzinger, and Wolfgang Winkler. *echt falsch*. Kremaer and Scheriau/Orac, Oct 2003.
- [Fac08] Fachhochschulrat. Plagiat von Wissenschaftlichen Aarbeiten. Website, 2008. http://www.fhr.ac.at/fhr_inhalt/03_studium/pruefungsorganisation.htm.
- [fBW07] Universität für Bodenkultur Wien. Plagiatsprüfung mitMyDropBox. Website, Feb 2007. <http://www.boku.ac.at/zid-pro-plagiat-2006.html>.
- [fFs07] Bundesministerium für Finanz Österreich. Produktpiraterie - Aufgriffe der Österreichischen Zollverwaltung 2007. Website des Bundesministeriums für Finanz, 2007. http://www.bmf.gv.at/Zoll/Produktpiraterie/DatenundFakten_3041.htm.
- [fFs08] Bundesministerium für Finanz Österreich. Internet-Shopping. Webseite des Bundesministeriums für Finaz, 2008. http://www.bmf.gv.at/Zoll/Wissenswertes/Postverkehr/InternetShopping/_start.htm.
- [fITUW07] Fakultät für Informatik Technische Universität Wien. Code of Ethics. Mar 2007. http://www.inso.tuwien.ac.at/fileadmin/Files/Graduands_Package/CodeOfEthics.pdf.

- [Ger08] Ephorus Germany, 2008. <http://www.ephorus.de/start>.
- [Haf06] Michaela Hafner. Mydropbox auf der Spur von Plagiaten. Website, Mar 2006. <http://www.dieuniversitaet-online.at/beitraege/news/mydropbox-auf-der-spur-von-plagiaten/10/neste/2.html>.
- [Hum05] Kerstin Humberg. Die Mentalität beginnt schon in der Schule. Website, Nov 2005. <http://www.spiegel.de/unispiegel/studium/0,1518,382788,00.html>.
- [iL08] iParadigms LLC., 2008. <http://turnitin.com/static/index.html>.
- [is08] industrial software. Plagiate. Website, Oct 2008. <http://www.inso.tuwien.ac.at/lectures/plagiate/>.
- [Kar01] Josef Karner. Der Plagiator Spione und Freaks. (22), Mar 2001. <http://old.onlinejournalismus.de/meinung/plagiator/spione.html>.
- [Kut97] Karsten Kutschera. Seminar Multimedia und Electronic Publishing kapitel 11: Rechtliche aspekte elektronischer veröffentlichung. Website der Universität Karlsruhe, Institut für Betriebs- und Dialogsysteme, Jan 1997. http://i31www.ira.uka.de/docs/mm+ep/11_RECHT/main_html.html.
- [Löb06] Prof. Sebastian Löbner. Zum Umgang mit den Quellen in Hausarbeiten. Website, 2006. Website mit Beispielen zu Kurzplagiaten, <http://user.phil-fak.uni-duesseldorf.de/~loebner/lehre/hausarb/plagiate.htm>.
- [Mal06] Helmut Malleck. Plagiate an den Universitäten. 4, 31, Nov 2006. <http://www.ocg.at/publikationen/oj/pdfs/oj0604.pdf>.
- [Pra07] Andreas Praefcke. Plagiatsfalle. Website, Nov 2007. <http://de.wikipedia.org/w/index.php?title=Plagiatsfalle&oldid=39036645>.
- [sH07] Österreichische HochschülerInnenschaft. Plagiate. Website, Sep 2007. http://www.oeh.ac.at/politik_und_gesellschaft/bildungspolitik/plagiate/?no_cache=1&sword_list%5B%5D=plagiat.
- [Sie07] Richard Sietmann. Plagiatsfahnder fielen durch. *c't*, 22, Oct 2007.
- [Öst] Bundeskanzleramt Österreich. Gesetzestexte Österreich. Website. <http://www.ris.bka.gv.at/bundesrecht/>.
- [Web07] Stefan Weber. *Das Google-Copy-Paste-Syndrom : wie Netzplagiate Ausbildung und Wissen gefährden*. Heise, 2007.
- [Wie04] Universität Wien. Was ist ein Plagiat? Website, 2004. <http://public.univie.ac.at/index.php?id=17954>.
- [Wie06] Universität Wien. Richtlinien, Verordnungen. Jan 2006. http://www.univie.ac.at/mtbl02/2005_2006/2005_2006_112.pdf.

Literaturverzeichnis

- [Wik08] Wikipedia. Longest common substring problem. Website, Oct 2008. http://en.wikipedia.org/w/index.php?title=Longest_common_substring_problem&oldid=248357663.
- [Wil06] Christoph Wille. Plagiat oder nicht? Website, 2006. <http://www.aspheute.com/artikel/20030910.htm>.
- [Wul07] Deborah Weber Wulf. Fremde Federn finden. Website, 2007. <http://plagiat.fhtw-berlin.de/ff/>.