Projektgruppenantrag der Projektgruppe **SciencePlag**

1 Thema

Wie lässt sich erschließen, ob Teile eines Dokuments abgeschrieben sind?
Wir werden uns in der PG mit der automatischen Suche nach Plagiaten beschäftigen.
Dazu werden wir Techniken aus der String-Algorithmik einsetzen.

2 Zeitraum

Sommersemester 2015, Wintersemester 2015/16; 8 SWS pro Semester.

3 Veranstalter

Lehrstuhl XI:

- **Prof. Dr. Johannes Fischer**, OH14/212, Tel. 7711, `johannes.fischer@cs.tu-dortmund.de`
- **Prof. Dr. Sven Rahmann**, OH14/214, Tel. 7713, `sven.rahmann@tu-dortmund.de`
- Dipl.-Math. Dominik Köppl, OH14/209, Tel. 7712, `dominik.koeppl@tu-dortmund.de`
- M.Sc. Florian Kurpicz, OH14/213, Tel. 7725, `florian.kurpicz@tu-dortmund.de`

4 Aufgabe

4.1 Hintergrund. Unter einem Plagiat versteht man ein Werk, das sich fremden Inhalts bedient, ohne die Originalquellen ausreichend zu kennzeichnen. Wer in einer schriftlichen Ausarbeitung Textpassagen aus anderen Arbeiten paraphrasiert oder Argumente und Faktenangaben übernimmt, ohne die Quellen im Einzelnen anzugeben, begeht ein Plagiat in diesem Sinne. Schon zu Zeiten des römischen Dichters Martial (40 n. Chr) wurde die Anmaßung fremden Inhalts öffentlich disputiert. Auch in den letzten Jahren sorgten Plagiatsvorfälle für Schlagzeilen. Webseiten wie GuttenPlag oder VroniPlag bieten eine Plattform für kollaborative Plagiatsdokumentationen.[1] Ihr Fokus liegt v.a. auf Dissertationen von Persönlichkeiten aus der Politik, Medizin und Wirtschaft. Im wissenschaftlichen Bereich kann ein Plagiat gegen Prüfungsordnungen, Arbeitsverträge oder Universitätsrecht im Sinne von Täuschung verstoßen. Schriftliche Ausarbeitungen, die anstelle einer völlig selbständig erstellten Arbeit nicht explizit ausgewiesene Anteile anderer Arbeiten enthalten, sind ein Verstoß gegen wissenschaftliche Grundregeln, die den Tatbestand der Täuschung erfüllen. Trotz der förmlichen Erklärungen über die selbständige Anfertigung ist in letzter Zeit leider ein erschreckender Anstieg von Täuschungsversuchen in studentischen Ausarbeitungen zu verzeichnen [10]. Auch die Vorgehensweise des Plagiierens hat sich durch die Digitalisierung textuellen Inhalts gewandelt [12]. Statt Bücher in einer Bibliothek werden Inhalte in Webseiten per Internet-Suchmaschinen gesucht und per “Copy&Paste” zu einem Werk zusammengefügt, wie Abbildung 1 veranschaulicht.

4.2 Aufgabenstellung. Ziel dieser PG ist die Kombination von Konzepten aus der *Stringology* sowie dem *Information Retrieval* zur Konzeption und Entwicklung eines Verfahrens, welches eine schnelle und genaue Suche nach Plagiaten in Textdokumenten (hier insb. PDF- und TXT-Dateien) ermöglicht. Gegeben ist hierbei eine Sammlung von Dokumenten, die als Basis für die Suche nach plagiierten Stellen in weiteren Dokumenten dient (z.B. eine Sammlung von Abschlussarbeiten).

Die zu entwickelnde Methode besteht aus zwei eng miteinander verknüpften Teilen. Zunächst muss die Dokumentenbasis vorverarbeitet (Bereich Information Retrieval, u.a. Stemming) und indiziert (Bereich Stringology, u.a. Suffixarrays, Suffixbäume) werden. Die Dokumentenbasis kann mehrere Zehntausend Dokumente enthalten. Eventuell werden für diese Aufgabe Algorithmen benötigt, die I/O-effizient im externen Speicher arbeiten. Für die Konstruktion einer möglichen Indexstruktur im

[1] GuttenPlag: `http://de.guttenplag.wikia.com/`, VroniPlag: `http://de.vroniplag.wikia.com`

externen Speicher existieren bereits praxiserprobte Algorithmen [1, 5], welche in der PG angepasst und verwendet werden können. Der zweite Teil besteht aus der Verwendung der zuvor implementierten Indexstruktur zur Suche nach plagiierten Stellen in Dokumenten und einer Bewertung der identifizierten, möglicherweise plagiierten Textbereiche.

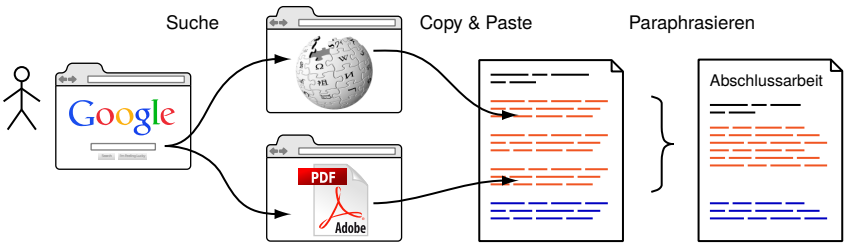

Abbildung 1: Elementare Schritte bei einem Plagiat aus dem Web [6].

Der Unterschied dieses Projekts zu existierenden Lösungen ist, dass die zu Grunde liegenden Techniken aus dem Bereich der Stringology und nicht, wie in vielen bestehenden Ansätzen, aus dem Bereich des *Machine Learning* stammen. Nach der Indizierung der Datenbasis folgt bei dem hier vorgeschlagenen Ansatz somit keine Lernphase; der Algorithmus ist direkt in der Lage, Plagiate zu erkennen.

Aus informatischer Sicht geht es also darum, eine (I/O-effziente) Datenstruktur für einen Volltext-Index (auf bereinigten Texten) zu entwickeln, die zum schnellen Auffinden von Plagiaten in Texten genutzt werden kann.

Neben der I/O-Effizienz gibt es weitere Anforderungen an die Datenstruktur für den Volltext-Index: Sie muss Anfragen schnell und verlässlich beantworten. Als Kennzahl für die Qualität richten wir uns nach den Größen, die für den Wettbewerb von Plagiat-erkennender Software im Rahmen der PAN[2] genutzt werden [7]. Dort wird ein Maß aus *Precision* und *Recall* in unterschiedlichen Umgebungen sowie der *Granularität* verwendet [8]. Neben dem Index müssen auch Verfahren aus dem Bereich Information Retrieval betrachtet werden. Vor der Erstellung des Index muss der Text mit Information-Retrieval-Verfahren vorverarbeitet werden. Dies ist notwendig, damit auch leicht abgeänderte Textstellen als Plagiat erkannt werden (Stichwort *Stemming*). Auch das Entfernen von *Stoppwörtern* spielt einen wichtigen Faktor bei der Vorverarbeitung der Daten. Hilfreich ist, dass die *Term Frequency* und *Inverted Document Frequency* ebenfalls mithilfe von Suffixarrays bestimmt werden kann [13].

Gerade in der Verwendung der Methoden aus der Stringology sehen die Veranstalter eine Möglichkeit, ein neues Verfahren für die Suche nach Plagiaten zu etablieren. Dementsprechend liegt der Fokus auf Datenstrukturen und Methoden aus diesem Bereich. Zur Zeit scheint es keine Software zur Erkennung von Plagiaten zu geben, welche vergleichbare Methoden und Konzepte nutzt. Somit bietet diese PG die Möglichkeit, eine neue Vorgehensweise für ein altbekanntes Problem zu erforschen.

Im Vorbereitungsseminar soll die Gruppe nötige informatische Grundlagen, insbesondere aus dem Bereich der Stringology (Textsuche, Hashing, Volltextindizierung) und die hierfür benötigten Datenstrukturen und deren Konstruktion auffrischen bzw. erarbeiten. Aktuelle und relevante Ergebnisse werden hierbei berücksichtigt, z.B. die Suffixarray-Konstruktion im externen Speicher [1, 5]. Auch die notwendigen Techniken aus dem Bereich Information Retrieval (Dokumentenformate, Stemming, Stoppwörter) werden im Rahmen des Seminars besprochen. Des Weiteren werden grundlegende Techniken des Projektmanagements, der Arbeit im Team sowie der Softwarekonstruktion

[2]Evaluation lab on uncovering plagiarism, authorship, and social software misuse: `http://pan.webis.de/`

vorgestellt.

Im ersten PG-Semester soll sich die Gruppe mit der Vorbereitung der Plagiatsfindung beschäftigen und insbesondere an Algorithmen zur Auswertung von Plagiatsfundstellen arbeiten. Dabei soll die Gruppe

- sich auf die zu verwendende Technik einigen,
- sich Gedanken über eine Softwarearchitektur machen, die festgelegte Schnittstellen unterstützt,
- einige der o.g. (algorithmischen) Techniken daraufhin untersuchen, ob sie sich für Plagiatsentdeckungen eignen, und auswählen,
- ein System zur Vorverarbeitung von Texten entwickeln (Prototyp) und
- eine Index-Datenstruktur entwickeln oder anpassen, die sich für die Verwendung in einem Plagiatsfinder eignet.

Nach Ablauf des ersten Semesters soll ein Software-Framework bereitstehen, das auf Basis einer Menge an Dokumenten in der Lage ist, eine für eine spätere Plagiatsuntersuchung geeignete Index-Datenstruktur zu erzeugen. Das Framework muss die Möglichkeit bieten, anhand des erstellten Indexes ein Eingabedokument als Plagiat zu klassifizieren.

Im zweiten Semester soll die Gruppe eine Software zur Plagiatsklassifizierung basierend auf der Datenstruktur entwickeln und dazu

- Schwächen des bestehenden Ansatzes untersuchen und ggf. diese ausbessern,
- das Framework um eine Benutzerschnittstelle in Form einer GUI erweitern,
- eine Evaluation der Geschwindigkeit, Speicherbedarf (RAM und extern) und Sensitivitätsgarantien durchführen und diese mit anderer verfügbarer Software vergleichen und
- ggf. die Vergrößerung des Datenbestands bzw. dessen Qualität vorantreiben (Crawler).

Am Ende des zweiten Semesters soll eine Software entstanden sein, das mit den für die PAN-Konferenz üblichen Ein- und Ausgabeformaten umgehen kann und in akzeptabler Geschwindigkeit eine Klassifizierung ermöglicht. Idealerweise funktioniert die Plagiatsklassifizierung auch mit großen Datensätzen (z.B. DBLP[3]) und bietet verschiedene Verfahren zur Klassifizierung.

Um die genannten Ziele zu erreichen, sollen die PG-Teilnehmer eine modulare objektorientierte Software-Plattform erstellen. Die Sprache kann von den PG-Teilnehmern gewählt werden; die Veranstalter empfehlen aus eigener Erfahrung, eine maschinennahe Programmiersprache wie C oder C++ zu verwenden. Wir legen Wert darauf, dass eine durchdachte Software-Architektur durch qualitativ hochwertige Entwicklungsarbeit entsteht. Hierzu gehören unter anderem automatisierte Tests und eine exzellente Dokumentation. Um die Software für Anwender sowohl aus der Informatik als auch außenstehend nutzbar zu machen, soll neben verschiedenen miteinander kombinierbaren Kommandozeilentools auch eine GUI-Version oder Browser-Version entstehen. Hierzu ist die Benutzerinteraktion möglichst gut vom algorithmischen Kern zu entkoppeln.

Da die zur Verfügung stehenden Ressourcen begrenzt sind, sollen alle Methoden zunächst auf (relativ) kleinen Testdatensätzen (z.B. aus den Testdatensätzen der PAN-Competition) ausführlich erprobt werden, bevor sie auf vollständige Datensätze im Produktivbetrieb angewendet werden.

5 Teilnahmevoraussetzungen

Legende: (V) Voraussetzung; (W) wünschenswert.

[3] `http://www.informatik.uni-trier.de/~ley/db/`

- (V) nachgewiesene Algorithmische Kompetenz (z.B. Modul im Forschungsbereich D)
- (V) sehr gute Programmierkenntnisse in C/C++ **oder** Java **oder** Python
- (W) Kenntnisse in oder äquivalent zu „Algorithmen auf Sequenzen“ [9] oder „Textindizierung und Information Retrieval“ [2] oder die Bereitschaft, sich diese im Laufe der PG anzueignen.

6 Minimalziel

Mindestens muss ein lauffähiges System zum Aufbau einer Indexstruktur aus Dokumenten sowie zum Auffinden von Plagiaten mithilfe dieser Indexstruktur vorliegen. Beide Systeme müssen Dokumente im Portable Document Format (PDF) und Textdateien unterstützen. Die Ausgabe aller kritischen Stellen erfolgt als XML-Format nach dem Schema der PAN 2015[4]. Zur einfacheren Interpretation der Ergebnisse ist eine Oberfläche zu entwickeln, welche die XML-Dateien optisch aufbereitet. Testdaten werden von den Veranstaltern bereit gestellt. Das Projekt beinhaltet einen Zwischenbericht, und endet mit einem Bericht und einer Abschlusspräsentation, welche u.a. einen Vergleich mit weiteren Software zur Erkennung vom Plagiaten enthält.

7 Literatur

[1] T. Bingmann, J. Fischer, and V. Osipov. Inducing Suffix and LCP Arrays in External Memory. In *Algorithm Engineering and Experiments (ALENEX)*, pages 88–102. SIAM, 2013.

[2] J. Fischer. Text-Indexierung und Information Retrieval. TU Dortmund, Vorlesungsskript, 2014.

[3] M. R. Henzinger. Finding near-duplicate web pages: a large-scale evaluation of algorithms. In *Special Interest Group on Information Retrieval (SIGIR)*, pages 284–291. ACM, 2006.

[4] T. C. Hoad and J. Zobel. Methods for Identifying Versioned and Plagiarized Documents. *JASIST*, 54(3):203–215, 2003.

[5] J. Kärkkäinen and D. Kempa. LCP Array Construction in External Memory. In *Experimental Algorithms*, volume 8504 of *LNCS, Springer*, pages 412–423. 2014.

[6] M. Potthast. *Technologies for Reusing Text from the Web.* Dissertation, Bauhaus-Universität Weimar, Dec. 2011.

[7] M. Potthast et al. Overview of the 4th International Competition on Plagiarism Detection. In *CLEF*, 2012.

[8] M. Potthast, B. Stein, A. Barrón-Cedeño, and P. Rosso. An Evaluation Framework for Plagiarism Detection. In *COLING*, pages 997–1005, 2010.

[9] S. Rahmann. Algorithmen auf Sequenzen. TU Dortmund, Vorlesungsskript, 2014.

[10] S. Sattler. *Plagiate in Hausarbeiten. Erklärungsmodelle mit Hilfe der Rational Choice Theorie.* Verlag Dr. Kovac, 2007.

[11] B. Stein, S. M. zu Eissen, and M. Potthast. Strategies for Retrieving Plagiarized Documents. In *SIGIR*, pages 825–826. ACM, 2007.

[12] Y. Wilks. On the Ownership of Text. *Computers and the Humanities*, 38(2):115–127, 2004.

[13] M. Yamamoto and K. W. Church. Using Suffix Arrays to Compute Term Frequency and Document Frequency for All Substrings in a Corpus. *Computational Linguistics*, 27(1):1–30, 2001.

8 Rechtliche Hinweise

Die Ergebnisse der Projektarbeit und die dabei erstellte Software sollen der Fakultät für Informatik der TU Dortmund und der Medizinischen Fakultät der Universität Duisburg-Essen uneingeschränkt für Lehr- und Forschungszwecke zur freien Verfügung stehen. Darüber hinaus sind keine Einschränkungen der Verwertungsrechte an den Ergebnissen der Projektgruppe und keine Vertraulichkeitsvereinbarungen vorgesehen. Bei Zustimmung aller Projektteilnehmer wird die Software als Open-Source-Projekt angelegt.

[4] `http://www.uni-weimar.de/medien/webis/research/events/pan-15/pan15-web/plagiarism-detection.html`