Plagiatserkennung

Von Mimmi Plagatia

Plagiatserkennung ist der Prozess des Auffindens von Plagiaten in einer Arbeit oder einem Dokument. Die weit verbreitete Nutzung von Computern und das Aufkommen des Internets haben es erleichtert, die Arbeit anderer zu plagiieren. Die meisten Fälle von Plagiaten finden sich in der Wissenschaft, wo Dokumente typischerweise Essays oder Berichte sind. Allerdings kann Plagiat in praktisch jedem Bereich gefunden werden, einschließlich Romanen, wissenschaftlichen Arbeiten, Kunstdesigns und Quellcodes.
Die Erkennung von Plagiaten kann entweder manuell oder softwaregestützt sein. Die manuelle Erkennung erfordert einen beträchtlichen Aufwand und einen ausgezeichneten Speicher und ist nicht praktikabel, wenn zu viele Dokumente verglichen werden müssen oder Originaldokumente nicht zum Vergleich verfügbar sind. Die softwaregestützte Erkennung ermöglicht es, umfangreiche Dokumentensammlungen miteinander zu vergleichen, wodurch eine erfolgreiche Erkennung viel wahrscheinlicher wird.
Die Praxis des Plagiierens durch Verwendung von ausreichenden Wortsubstitutionen, um sich einer Erkennungssoftware zu entziehen, wird als Rogeting bezeichnet. [1]

Software-unterstützte Erkennung

Computergestützte Plagiatdetektion (CaPD) ist eine Information Retrieval (IR) -Task, die von spezialisierten IR-Systemen unterstützt wird, die als Plagiatdetektionssysteme (PDS) bezeichnet werden.
In Textdokumenten
Systeme zur Erkennung von Textplagiaten implementieren einen von zwei generischen Erkennungsansätzen, von denen einer extern und der andere intrinsisch ist. [2] Externe Erkennungssysteme vergleichen ein verdächtiges Dokument mit einer Referenzsammlung, bei der es sich um eine Gruppe von Dokumenten handelt, von denen angenommen wird, dass sie echt sind. [3] Basierend auf einem ausgewählten Dokumentmodell und vordefinierten Ähnlichkeitskriterien besteht die Erkennungsaufgabe darin, alle Dokumente, die Text enthalten, der einem Grad oberhalb eines ausgewählten Schwellenwerts ähnelt, für den Text im verdächtigen Dokument abzurufen. [4] Intrinsische PDS analysieren ausschließlich den zu bewertenden Text, ohne Vergleiche mit externen Dokumenten durchzuführen. Dieser Ansatz zielt darauf ab, Veränderungen im einzigartigen Schreibstil eines Autors als Indikator für potentielles Plagiat zu erkennen. [5] PDS sind nicht in der Lage, Plagiate ohne menschliches Urteilsvermögen zuverlässig zu identifizieren. Ähnlichkeiten werden mit Hilfe vordefinierter Dokumentmodelle berechnet und stellen möglicherweise falsche Positive dar. [6] [7] [8] [9] [10]
Effektivität von Hochschuleinstellungen
Dieser Abschnitt basiert weitgehend oder vollständig auf einer einzigen Quelle. Relevante Diskussionen finden Sie auf der Diskussionsseite. Bitte helfen Sie diesen Artikel zu verbessern, indem Sie Zitate in zusätzliche Quellen einfügen. (Dezember 2017)
Eine Studie wurde durchgeführt, um die Wirksamkeit von Plagiatserkennungssoftware in einem Hochschulbereich zu testen. Ein Teil der Studie beauftragte eine Gruppe von Studenten, eine Arbeit zu schreiben. Diese Studenten wurden zuerst über Plagiate

informiert und informierten, dass ihre Arbeit durch ein Plagiatserkennungssystem geführt werden sollte. Eine zweite Gruppe von Studenten wurde beauftragt, eine Zeitung ohne jegliche Information über Plagiate zu schreiben. Die Forscher erwarteten niedrigere Raten in der ersten Gruppe, fanden aber in beiden Gruppen ungefähr die gleichen Raten von Plagiaten. [11]

Ansätze

Die folgende Abbildung zeigt eine Klassifizierung aller Erkennungsmethoden, die derzeit für die computergestützte Plagiatserkennung verwendet werden. Die Ansätze sind durch die Art der Ähnlichkeitsbewertung gekennzeichnet, die sie durchführen: global oder lokal. Globale Ähnlichkeitsbeurteilungsansätze verwenden die Merkmale, die aus größeren Teilen des Textes oder des Dokuments als Ganzes genommen werden, um Ähnlichkeit zu berechnen, während lokale Verfahren nur vorausgewählte Textsegmente als Eingabe untersuchen.

Klassifizierung von computergestützten Plagiatserkennungsmethoden

Fingerabdruck

Fingerprinting ist derzeit der am weitesten verbreitete Ansatz zur Erkennung von Plagiaten. Diese Methode erstellt repräsentative Digests von Dokumenten, indem sie eine Menge mehrerer Teilstrings (N-Gramme) aus ihnen auswählt. Die Sets stellen die Fingerabdrücke dar und ihre Elemente werden als Minutien bezeichnet. [12] [13] Ein verdächtiges Dokument wird auf Plagiat geprüft, indem sein Fingerabdruck berechnet und Minuzien mit einem vorberechneten Index von Fingerabdrücken für alle Dokumente einer Referenzsammlung abgefragt werden. Minutien, die mit denen anderer Dokumente übereinstimmen, zeigen geteilte Textsegmente an und schlagen potentielles Plagiat vor, wenn sie eine gewählte Ähnlichkeitsschwelle überschreiten. [14] Computerressourcen und Zeit sind limitierende Faktoren für den Fingerabdruck, weshalb diese Methode in der Regel nur eine Teilmenge von Minutien vergleicht, um die Berechnung zu beschleunigen und Prüfungen in sehr großen Sammlungen wie dem Internet zu ermöglichen. [12]

String-Übereinstimmung

String-Matching ist ein vorherrschender Ansatz, der in der Informatik verwendet wird. Bei der Anwendung auf das Problem der Plagiatserkennung werden Dokumente für wörtliche Textüberlappungen verglichen. Für diese Aufgabe wurden zahlreiche Methoden vorgeschlagen, von denen einige an die externe Plagiatserkennung angepasst wurden. Das Überprüfen eines verdächtigen Dokuments in dieser Einstellung erfordert die Berechnung und Speicherung von effizient vergleichbaren Darstellungen für alle

Dokumente in der Referenzsammlung, um sie paarweise zu vergleichen. Im Allgemeinen wurden für diese Aufgabe Suffixdokumentmodelle wie Suffixbäume oder Suffixvektoren verwendet. Nichtsdestoweniger bleibt die Anpassung von Teilstrings rechenintensiv, was es zu einer unrentablen Lösung für die Überprüfung großer Dokumentensammlungen macht. [15] [16] [17]

Beutel mit Wörtern

Die Bag of Words-Analyse repräsentiert die Einführung der Vektorraumsuche, ein traditionelles IR-Konzept, in die Domäne der Plagiatserkennung. Dokumente werden als ein oder mehrere Vektoren, z.B. für verschiedene Dokumententeile, die für paarweise Ähnlichkeitsberechnungen verwendet werden. Die Ähnlichkeitsberechnung kann sich dann auf das traditionelle Kosinusähnlichkeitsmaß oder auf komplexere Ähnlichkeitsmaße stützen. [18] [19] [20]

Zitatanalyse

Citation-based plagiarism detection (CbPD) [21] beruht auf Zitationsanalyse und ist der einzige Ansatz zur Erkennung von Plagiaten, der nicht auf der textlichen Ähnlichkeit beruht. [22] CbPD untersucht die Zitat- und Referenzinformationen in Texten, um ähnliche Muster in den Zitatsequenzen zu identifizieren. Daher eignet sich dieser Ansatz für wissenschaftliche Texte oder andere akademische Dokumente, die Zitate enthalten. Die Zitieranalyse zur Erkennung von Plagiaten ist ein relativ junges Konzept. Es wurde nicht von kommerzieller Software übernommen, aber es existiert ein erster Prototyp eines zitationsbasierten Plagiatdetektionssystems. [23] Eine ähnliche Reihenfolge und Nähe von Zitaten in den untersuchten Dokumenten sind die Hauptkriterien, die verwendet werden, um Zitationsmusterähnlichkeiten zu berechnen. Zitationsmuster stellen Teilsequenzen dar, die nicht ausschließlich Zitate enthalten, die von den verglichenen Dokumenten geteilt werden. [22] [24] Faktoren wie die absolute Anzahl oder der relative Anteil gemeinsamer Zitate im Muster sowie die Wahrscheinlichkeit, dass Zitate in einem Dokument vorkommen, werden ebenfalls berücksichtigt, um den Grad der Ähnlichkeit der Muster zu quantifizieren. [22] [24] [25] [26]

Stylometrie

Die Stylometrie fasst statistische Methoden zur Quantifizierung des einzigartigen Schreibstils eines Autors zusammen [27] [28] und wird hauptsächlich für Autorenattribution oder intrinsische CaPD verwendet. Durch die Konstruktion und den Vergleich von Stylometriemodellen für verschiedene Textsegmente können Passagen erkannt werden, die sich stilistisch von anderen unterscheiden und daher möglicherweise plagiiert werden. [5]

Performance

Vergleichende Auswertungen von Plagiatserkennungssystemen [3] [29] [30] [31] [32] [33] weisen darauf hin, dass ihre Leistungsfähigkeit von der Art des vorliegenden Plagiats abhängt (siehe Abbildung). Mit Ausnahme der Zitationsmusteranalyse beruhen alle Erkennungsansätze auf textähnlicher Ähnlichkeit. Es ist daher symptomatisch, dass die Erkennungsgenauigkeit abnimmt, je mehr Plagiatsfälle verschleiert werden.

Detection performance of CaPD approaches depending on the type of plagiarism being present

Komplettkopien, auch *Copy & Paste*-Plagiat genannt, oder verschleierte Plagiate können mit hohem Genauigkeit erkannt werden durch aktuelle Plagiatserkennungssoftware, falls die Quelle zugänglich ist für die Software. Besonders Teilzeichenreihenvergleich-Prozeduren erreichen gute Ergebnisse für C&P-Plagiat, da sie in der Regel verlustfreie Dokumentenmodelle verwenden, zum Beispiel *suffix trees*. Bei Systeme die Fingerabdrücke oder Multimengen von Wörtern (*bag of words*) Verfahren bei der Detektion anwenden hängt der Güte der Ergebnisse von der Informationsverlust ab die durch den Dokumentenmodell gegeben ist. Durch die Anwendung von flexiblen Ausschnitt- bzw. Auswahl-Strategien, sind sie besser dazu in der Lage, moderate Formen von Verschleierungen zu erkennen im Vergleich mit Teilzeichenreihenvergleicher.

Intrinsischer Plagiatserkennung mit Stilometrie kann die Grenzen von Textähnlichkeit überwinden zum Teil durch linguistischer Ähnlichkeitsvergleiche. Da die stilistische Differenzen zwischen plagiierte und originale Textabschnitte signifikant sind und zuverlässig erkannt werden können, ist Stilometrie dazu in der Lage, verschleierte oder paraphrasierte Plagiat zu erkennen. Stilometrische Vergleiche können scheitern in den Fälle, in den Textabschnitte sehr stark paraphrasiert sind, so stark dass sie eher der persönlichen Schreibstil des Plagiaristen ähneln, oder falls einen Text durch mehreren Autoren zusammengestellt wurde. Die Resultate von der *International Competitions on Plagiarism Detection*, die in 2009, 2010 und 2011 abgehalten worden sind, [3][32][33] sowie die Experimente, die durch Stein durchgeführt worden sind [34], zeigen dass stilometrische Analyse nur dann zuverlässig funktioniert wenn die Dokumente mehrere Tausend oder gar Zehntausende von Wörter umfasst. Das schränkt die Anwendbarkeit der Methode stark ein.

Forschung im Bereich der Methoden und Systeme die Übersetzungsplagiate erkennen können nimmt stark zu. Aktuell kann intersprachliche Plagiatserkennung (*cross-language plagiarism detection*, CLPD) nicht als ausgereifte Technologie angesehen werden [35] und die entsprechenden Systemen haben gute Erkennungsresultae nicht im Praxistest zeigen können. [31]

Zitationsbasierte Plagiatserkennung mit Hilfe von Zitationsmusteranalyse ist dazu in der Lage, stärkere Paraphrasen und Übersetzungen mit höheren Erfolgsquoten zu erkennen im Vergleich zu anderen Methoden, weil es unabhängig von Texteigenschaften ist.[22][25] Aber da Zitationsmusteranalyse sich stark darauf verlässt, das ausreichende Beleginformation vorhanden ist, ist es nur für akademische Texte anwendbar. Es bleibt weit zurück gegenüber textbasierte Vorhaben in der Erkennung von kürzeren plagiierte Passagen, die typisch sind für *copy-and-paste* oder *shake-and-paste* Plagiate. Letzteres ist ein Verfahren, dass leicht veränderte Fragmente von unterschiedliche Quellen vermischt [36].

Software

Der Architektur von Plagiatserkennungssoftware für Textdokumenten wird durch mehreren Faktoren charakterisiert:

Faktor	Beschreibung und Alternativen
Suchumfang	Im offenen Internet, mit Hilfe von Suchmaschinen / Institutionelle Datenbanken / Lokaler, system-spezifischer Datenbank.
Analysezeit	Differenz zwischen die Zeit, an dem ein Dokument eingereicht wird und der Zeitpunkt, an dem die Ergebnisse vorhanden sind.
Dokumentenkapazität / Stapelverarbeitung	Anzahl der Dokumenten, die das System pro Zeiteinheit verarbeiten kann.
Prüfintensität	Wie oft und für welche Arten von Dokumentenfragmente (Absätze, Sätze, Wordsequenzen fixer Länge) fragt das System nach externe Ressourcen, wie zum Beispiel Suchmaschinen?
Vergleichsalgorithmus	Die Algorithmen, die definieren wie das System Dokumente miteinander vergleicht.
Präzision und Recall	Anzahl von Dokumente die korrekt als Plagiat erkannt worden sind vergleichen mit dem Gesamtzahl der erkannte Dokumente, und zu den Gesamtzahl der Dokumente die wirklich plagiiert worden sind. Hohe Präzision bedeutet, dass es wenige falsch Positive gefunden worden sind, hoher Recall bedeutet, dass wenige falsch Negativen unerkannt worden sind.

Die meisten großen Plagiatserkennungssystem verwenden große, interne Datenbanken und andere Ressourcen. Diese wachsen mit jeder zusätzlicher Dokument, das zum Analysieren eingereicht wird. Aber manche halten dieses Vorgehen für ein Vergehen gegen das Urheberrecht der Studierenden.

In Quellcode

Plagiat in Computerquellcode ist auch häufig, und benötigt andere Werkzeuge als die, die für Textvergleiche von Dokumente verwendet werden. Viel Forschung ist bereits investiert worden in akademischer Quellcode Plagiat. [37]

Ein besonderer Aspekte von Quellcodeplagiat ist, dass es keine Aufsatzdatenbanken gibt, wie es für traditioneller Plagiat gibt. Da die meisten Programmieraufgaben erwarten, dass Studenten Programme schreiben mit ganz spezifischer Anforderungen, ist es sehr schwer vorhandene Programme zu finden, die diese Anforderungen erfüllen. Da es schwieriger ist, externe Code zu integrieren als es selber zu schreiben, plagiieren die meisten Studenten von ihre Kommilitonen.

Nach Roy und Cordy,[38] kann man Quellcode-Ähnlichkeits-Erkennungsalgorithmen wie folgt klassifizieren:

- Zeichenreihen – genaue Textübereinstimmung für Segmente, wie zum Beispiel fünf Wörter, werden gesucht. Das ist schnell, aber wird durch die gebundene Umbenennung von Variablen verwirrt.
- Token – wie mit Zeichenreichen, aber ein Lexer wir zuerst über den Code geschickt, um das Programm in einer Tokensequenz zu konvertieren. Damit werden Leerzeichen, Kommentare, Identifikatorennamen, und so weiter entfernt. Somit wird das System resistent gegen einfache Textvertauschungen. Die meisten akademischen Plagiatserkennungssysteme arbeiten auf dieser Ebene, mit unterschiedlichen Algorithmen um die Ähnlichkeit zu messen.
- Abstrakt Syntaxbäume – erstelle und vergleiche abstrakte Systaxbäume. Damit können Ähnlichkeiten auf höheren Ebenen gefunden werden. Zum Beispiel kann man mit Baumvergleiche normalisierte bedingte Anweisungen erkennen, und auch äquivalente Konstrukte können erkannt werden.
- Programabhängigkeiten-Graphen (PDGs) – ein PDG verzeichnet das tatsächlicher Kontrollfluss in einem Programm und erlaubt Äquivalenzen auf einen wesentlichen höheren Niveau zu erkennen. Allerdings gibt es hohe Komplexitäten und benötigten Rechenzeit.
- Metriken – Metriken errechnen Werte für Code-Segmenten für bestimmte Kriterien, z. B. “die Anzahl von Schleifen und If-Anweisungen” oder die “Anzahl unterschiedliche Variablen”. Metriken sind einfach zu berechnen und können schnell verglichen werden, aber können viele falsche Positiven lievern, da zwei Fragmente mit den gleichen Werten durchaus anderes machen können.
- Hybride Vorgehen – zum Beispiel abstrakte Syntaxbäume und Suffixbäume können kombiniert werden, um die Erkennungskapazitäten von abstrakte Syntaxbäume mit der Geschwindigkeit von Suffixbäume zu kombinieren, eine Art von Zeichenreihenvergleichsdatenstruktur.

Das vorangegangene Klassifikationssystem wurde für Code-Refactoring entwickelt und nicht für akademischer Plagiatserkennung. Ein wichtiger Ziel von Refactoring ist es, Code-Wiederholung zu vermeiden. Dieser werden code clones in der Literatur genannt. Die oben genannten Vorgehensweisen sind effektiv gegenüber unterschiedliche Ebenen der Ähnlichkeit. Niedrige Ähnlichkeit ist Textidentität, hoher Ähnlichkeit kann daraus resultieren, dass ähnliche Spezifikationen gegeben worden sind. In einer akademischer Umgebung, wenn von allen Studierenden erwartet wird, dass sie Code produzieren für identische Spezifikationen, wird erwartet dass der Code funktional äquivalent ist. Nur Niedrige Ähnlichkeit wird als ein Beweis angesehen, dass geschummelt worden ist.

Literaturliste

1. Grove, Jack (7 August 2014). "Sinister buttocks? Roget would blush at the crafty cheek Middlesex lecturer gets to the bottom of meaningless phrases found while marking essays". Times Higher Education. Retrieved 15 July 2015.
2. Stein, Benno; Koppel, Moshe; Stamatatos, Efstathios (Dec 2007), "Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection PAN'07" (PDF), SIGIR Forum, **41** (2), doi:10.1145/1328964.1328976
3. Potthast, Martin; Stein, Benno; Eiselt, Andreas; Barrón-Cedeño, Alberto; Rosso, Paolo (2009), "Overview of the 1st International Competition on Plagiarism Detection", PAN09 - 3rd Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse and 1st International Competition on Plagiarism Detection (PDF), CEUR Workshop Proceedings, **502**, pp. 1–9, ISSN 1613-0073, archived from the original (PDF) on 2 April 2012
4. Stein, Benno; Meyer zu Eissen, Sven; Potthast, Martin (2007), "Strategies for Retrieving Plagiarized Documents", Proceedings 30th Annual International ACM SIGIR Conference (PDF), ACM, pp. 825–826, doi:10.1145/1277741.1277928, ISBN 978-1-59593-597-7
5. Meyer zu Eissen, Sven; Stein, Benno (2006), "Intrinsic Plagiarism Detection", Advances in Information Retrieval 28th European Conference on IR Research, ECIR 2006, London, UK, April 10–12, 2006 Proceedings (PDF), Lecture Notes in Computer Science, **3936**, Springer, pp. 565–569, doi:10.1007/11735106_66
6. Bao, Jun-Peng; Malcolm, James A. (2006), "Text similarity in academic conference papers", 2nd International Plagiarism Conference Proceedings (PDF), Northumbria University Press
7. Clough, Paul (2000), Plagiarism in natural and programming languages an overview of current tools and technologies (PDF) (Technical Report), Department of Computer Science, University of Sheffield, archived from the original (PDF) on 18 August 2011
8. Culwin, Fintan; Lancaster, Thomas (2001), "Plagiarism issues for higher education" (PDF), Vine, **31** (2): 36–41, doi:10.1108/03055720010804005, archived from the original (PDF) on 5 April 2012
9. Lancaster, Thomas (2003), Effective and Efficient Plagiarism Detection (PDF) (PhD Thesis), School of Computing, Information Systems and Mathematics South Bank University[permanent dead link]
10. Maurer, Hermann; Zaka, Bilal (2007), "Plagiarism - A Problem And How To Fight It", Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007, AACE, pp. 4451–4458
11. Youmans, Robert J. (November 2011). "Does the adoption of plagiarism-detection software in higher education reduce plagiarism?". Studies in Higher Education. **36** (7): 749–761. doi:10.1080/03075079.2010.523457.
12. Hoad, Timothy; Zobel, Justin (2003), "Methods for Identifying Versioned and Plagiarised Documents" (PDF), Journal of the American Society for Information Science and Technology, **54** (3): 203–215, CiteSeerX 10.1.1.18.2680 , doi:10.1002/asi.10170
13. Stein, Benno (July 2005), "Fuzzy-Fingerprints for Text-Based Information Retrieval", Proceedings of the I-KNOW '05, 5th International Conference on Knowledge Management, Graz, Austria (PDF), Springer, Know-Center, pp. 572–579
14. Brin, Sergey; Davis, James; Garcia-Molina, Hector (1995), "Copy Detection Mechanisms for Digital Documents", Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data (PDF), ACM, pp. 398–409, doi:10.1145/223784.223855, ISBN 1-59593-060-4
15. Monostori, Krisztián; Zaslavsky, Arkady; Schmidt, Heinz (2000), "Document Overlap Detection System for Distributed Digital Libraries", Proceedings of the fifth ACM conference on Digital libraries (PDF), ACM, pp. 226–227, doi:10.1145/336597.336667, ISBN 1-58113-231-X
16. Baker, Brenda S. (February 1993), On Finding Duplication in Strings and Software (Technical Report), AT&T Bell Laboratories, NJ, archived from the original (gs) on 30 October 2007
17. Khmelev, Dmitry V.; Teahan, William J. (2003), "A Repetition Based Measure for Verification of Text Collections and for Text Categorization", SIGIR'03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, pp. 104–110, CiteSeerX 10.1.1.9.6155 , doi:10.1145/860435.860456
18. Si, Antonio; Leong, Hong Va; Lau, Rynson W. H. (1997), "CHECK: A Document Plagiarism Detection System", SAC '97: Proceedings of the 1997 ACM symposium on Applied computing (PDF), ACM, pp. 70–77, doi:10.1145/331697.335176, ISBN 0-89791-850-9
19. Dreher, Heinz (2007), "Automatic Conceptual Analysis for Plagiarism Detection" (PDF), Information and Beyond: The Journal of Issues in Informing Science and Information Technology, **4**: 601–614

20. *Muhr, Markus; Zechner, Mario; Kern, Roman; Granitzer, Michael (2009), "External and Intrinsic Plagiarism Detection Using Vector Space Models", PAN09 - 3rd Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse and 1st International Competition on Plagiarism Detection (PDF), CEUR Workshop Proceedings, **502**, pp. 47–55, ISSN 1613-0073, archived from the original (PDF) on 2 April 2012*
21. *Gipp, Bela (2014), Citation-based Plagiarism Detection, Springer Vieweg Research, ISBN 978-3-658-06393-1*
22. *Gipp, Bela; Beel, Jöran (June 2010), "Citation Based Plagiarism Detection - A New Approach to Identifying Plagiarized Work Language Independently", Proceedings of the 21st ACM Conference on Hypertext and Hypermedia (HT'10) (PDF), ACM, pp. 273–274, doi:10.1145/1810617.1810671, ISBN 978-1-4503-0041-4*
23. *Gipp, Bela; Meuschke, Norman; Breitinger, Corinna; Lipinski, Mario; Nürnberger, Andreas (28 July 2013), "Demonstration of Citation Pattern Analysis for Plagiarism Detection", Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (PDF), ACM, doi:10.1145/2484028.2484214*
24. *Gipp, Bela; Meuschke, Norman (September 2011), "Citation Pattern Matching Algorithms for Citation-based Plagiarism Detection: Greedy Citation Tiling, Citation Chunking and Longest Common Citation Sequence", Proceedings of the 11th ACM Symposium on Document Engineering (DocEng2011) (PDF), ACM, pp. 249–258, doi:10.1145/2034691.2034741, ISBN 978-1-4503-0863-2*
25. *Gipp, Bela; Meuschke, Norman; Beel, Jöran (June 2011), "Comparative Evaluation of Text- and Citation-based Plagiarism Detection Approaches using GuttenPlag", Proceedings of 11th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'11) (PDF), ACM, pp. 255–258, doi:10.1145/1998076.1998124, ISBN 978-1-4503-0744-4*
26. *Gipp, Bela; Beel, Jöran (July 2009), "Citation Proximity Analysis (CPA) - A new approach for identifying related work based on Co-Citation Analysis", Proceedings of the 12th International Conference on Scientometrics and Informetrics (ISSI'09) (PDF), International Society for Scientometrics and Informetrics, pp. 571–575, ISSN 2175-1935*
27. *Holmes, David I. (1998), "The Evolution of Stylometry in Humanities Scholarship", Literary and Linguistic Computing, **13** (3): 111–117, doi:10.1093/llc/13.3.111*
28. *Juola, Patrick (2006), "Authorship Attribution" (PDF), Foundations and Trends Information Retrieval, **1**: 233–334, doi:10.1561/1500000005, ISSN 1554-0669*
29. *Portal Plagiat - Softwaretest 2004 (in German), HTW University of Applied Sciences Berlin, retrieved 6 October 2011*
30. *Portal Plagiat - Softwaretest 2008 (in German), HTW University of Applied Sciences Berlin, retrieved 6 October 2011*
31. *Portal Plagiat - Softwaretest 2010 (in German), HTW University of Applied Sciences Berlin, retrieved 6 October 2011*
32. *Potthast, Martin; Barrón-Cedeño, Alberto; Eiselt, Andreas; Stein, Benno; Rosso, Paolo (2010), "Overview of the 2nd International Competition on Plagiarism Detection", Notebook Papers of CLEF 2010 LABs and Workshops, 22–23 September, Padua, Italy (PDF)*
33. *Potthast, Martin; Eiselt, Andreas; Barrón-Cedeño, Alberto; Stein, Benno; Rosso, Paolo (2011), "Overview of the 3rd International Competition on Plagiarism Detection", Notebook Papers of CLEF 2011 LABs and Workshops, 19–22 September, Amsterdam, Netherlands (PDF)*
34. *Stein, Benno; Lipka, Nedim; Prettenhofer, Peter (2011), "Intrinsic Plagiarism Analysis" (PDF), Language Resources and Evaluation, **45** (1): 63–82, doi:10.1007/s10579-010-9115-y, ISSN 1574-020X*
35. *Potthast, Martin; Barrón-Cedeño, Alberto; Stein, Benno; Rosso, Paolo (2011), "Cross-Language Plagiarism Detection" (PDF), Language Resources and Evaluation, **45** (1): 45–62, doi:10.1007/s10579-009-9114-z, ISSN 1574-020X*
36. *Weber-Wulff, Debora (June 2008), "On the Utility of Plagiarism Detection Software", In Proceedings of the 3rd International Plagiarism Conference, Newcastle Upon Tyne (PDF)*
37. "Plagiarism Prevention and Detection - On-line Resources on Source Code Plagiarism". Higher Education Academy, University of Ulster.
38. Roy, Chanchal Kumar;Cordy, James R. (26 September 2007)."A Survey on Software Clone Detection Research". School of Computing, Queen's University, Canada.
39. Carrol, J. (2002). A *handbook for deterring plagiarism in higher education*. Oxford: The Oxford Centre for Staff and Learning Development, Oxford Brookes University. (96 p.), ISBN 1873576560
40. Zeidman, B. (2011). *The Software IP Detective's Handbook*. Prentice Hall. (480 p.), ISBN 0137035330