


Detección de plagio

La **detección de plagio** es el proceso de localizar instancias de plagio dentro de un trabajo o documento. El uso generalizado de computadoras y la llegada de Internet ha facilitado el plagio del trabajo de otros. La mayoría de los casos de plagio se encuentran en la academia, donde los documentos suelen ser ensayos o informes. Sin embargo, el plagio se puede encontrar en prácticamente cualquier campo, incluidas las novelas, documentos científicos, diseños de arte y código fuente.

La detección de plagio puede ser manual o asistida por software. La detección manual requiere un esfuerzo considerable y una memoria excelente, y no es práctico en casos en los que se deben comparar demasiados documentos, o los documentos originales no están disponibles para la comparación. La detección asistida por software permite que grandes colecciones de documentos se puedan comparar entre sí, lo que hace que la detección exitosa sea mucho más probable.

La práctica de plagiar mediante el uso de suficientes sustituciones de palabras para eludir el software de detección se conoce como rogeting. [1] 

Contenido

- 1.Detección asistida por software
 - 1.1En documentos de texto
 - 1.1.1 Eficacia de en entornos de educación superior
 - 1.1.2Enfoques
 - 1.1.2.1Fingerprinting
 - 1.1.2.2String matching
 - 1.1.2.3Bol de palabras
 - 1.1.2.4Análisis de citas
 - 1.1.2.5Stylometry
 - 1.1.3Performance
 - 1.1.4Software
 - 1.2Código de fuente
- 2Ver también
- 3Referencias
- 4Literatura
- 5Enlaces externos

Detección asistida por software [\[editar\]](#)

La detección de plagio asistida por computadora (CaPD) es una tarea de recuperación de información (IR) respaldada por sistemas IR especializados, conocidos como sistemas de detección de plagio (PDS).

En documentos de texto [\[editar\]](#)

Los sistemas para la detección de plagio de texto implementan uno de dos enfoques de detección genéricos, uno es externo, el otro es intrínseco. [2] Los sistemas de detección externos comparan un documento sospechoso con una colección de referencia, que es un conjunto de documentos que se supone genuinos. [3] Según el modelo de documento elegido y los criterios de similitud predefinidos, la tarea de detección consiste en recuperar todos los documentos que contienen texto que es similar a un grado superior al umbral elegido para el

texto en el documento sospechoso. [4] Los PDS intrínsecos solo analizan el texto a evaluar sin realizar comparaciones con documentos externos. Este enfoque tiene como objetivo reconocer los cambios en el estilo de escritura único de un autor como indicador de posible plagio. [5] Los PDS no son capaces de identificar confiablemente el plagio sin juicio humano. Las similitudes se calculan con la ayuda de modelos de documentos predefinidos y pueden representar falsos positivos. [6] [7] [8] [9] [10]

Eficacia de en entornos de educación superior [\[editar\]](#)

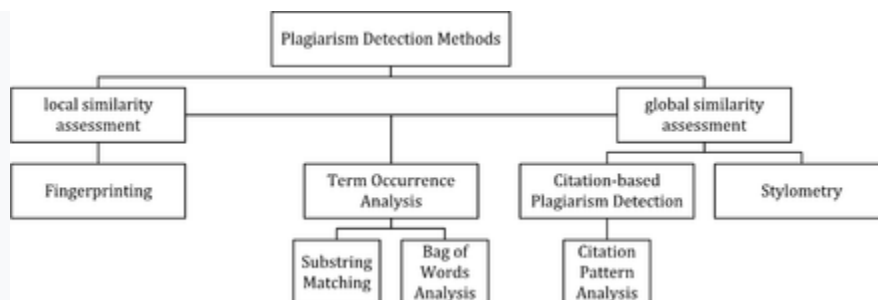


Esta sección se basa en gran parte o en su totalidad en una sola fuente. La discusión relevante se puede encontrar en la página de discusión. Ayúdenos a mejorar este artículo introduciendo citas a fuentes adicionales. (Diciembre de 2017)

Se realizó un estudio para evaluar la efectividad del software de detección de plagio en un entorno de educación superior. Una parte del estudio asignó a un grupo de estudiantes para escribir un artículo. Estos estudiantes fueron educados por primera vez acerca del plagio e informaron que su trabajo se someterá a un sistema de detección de plagio. Un segundo grupo de estudiantes fue asignado a escribir un artículo sin ninguna información sobre plagio. Los investigadores esperaban encontrar tasas más bajas en el grupo uno, pero encontraron aproximadamente las mismas tasas de plagio en ambos grupos. [11]

Enfoques [\[editar\]](#)

La siguiente figura representa una clasificación de todos los enfoques de detección actualmente en uso para la detección de plagio asistida por computadora. Los enfoques se caracterizan por el tipo de evaluación de similitud que realizan: global o local. Los enfoques de evaluación de similitud global utilizan las características tomadas de partes más grandes del texto o del documento como un todo para calcular la similitud, mientras que los métodos locales solo examinan segmentos de texto preseleccionados como entrada.



Classification of computer-assisted plagiarism detection methods

Huella digital [\[editar\]](#)

La toma de huellas dactilares es actualmente el enfoque más aplicado para la detección de plagio. Este método forma compendios representativos de documentos al seleccionar de ellos un conjunto de subcadenas múltiples (n-grams). Los conjuntos representan las huellas dactilares y sus elementos se llaman minutiae. [12] [13] Un documento sospechoso se verifica para ver si hay plagio al computar su huella digital y consultar minutiae con un índice precalculado de huellas dactilares para todos los documentos de una colección de referencia. Las minucias que coinciden con las de otros documentos indican segmentos de texto compartido y sugieren posibles plagios si superan un umbral de similitud elegido. [14] Los recursos computacionales y el tiempo son factores limitantes para la toma de huellas dactilares, por lo que este método generalmente solo compara un subconjunto de minucias para acelerar el cálculo y permitir controles en colecciones muy grandes, como Internet. [12]

Coincidencia de cadenas [\[editar\]](#)

La coincidencia de cadenas es un enfoque predominante utilizado en ciencias de la computación. Cuando se aplica al problema de detección de plagio, los documentos se comparan para superposiciones de texto literales. Se han propuesto numerosos métodos para abordar esta tarea, algunos de los cuales se han adaptado a la detección de plagio externo. Verificar un documento sospechoso en esta configuración requiere el cálculo y el almacenamiento de representaciones que sean comparables de manera eficiente para todos los documentos en la colección de referencia para compararlos por parejas. En general, los modelos de documento de sufijo, como árboles de sufijos o vectores de sufijo, se han utilizado para esta tarea. No obstante, la coincidencia de subcadenas sigue siendo computacionalmente costosa, lo que la convierte en una solución inviable para verificar grandes colecciones de documentos. [15] [16] [17]

Bolsa de palabras [\[editar\]](#)

El análisis de bolsa de palabras representa la adopción de la recuperación de espacio vectorial, un concepto de IR tradicional, al dominio de la detección de plagio. Los documentos se representan como uno o múltiples vectores, p. para diferentes partes del documento, que se usan para cálculos de similitud por pares. El cálculo de similitud puede entonces basarse en la medida de similitud de coseno tradicional o en medidas de similitud más sofisticadas. [18] [19] [20]

Análisis de citas [\[editar\]](#)

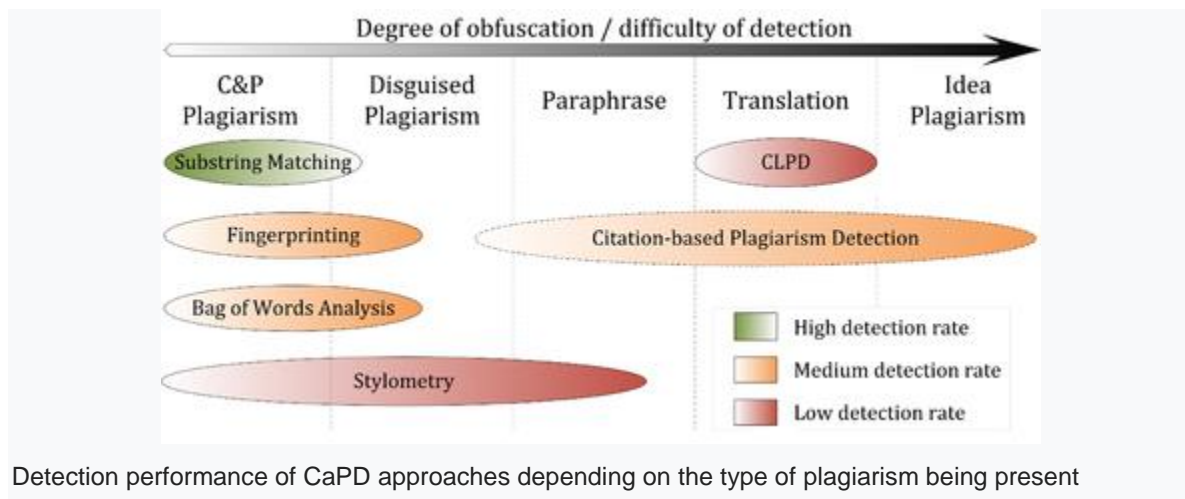
La detección de plagio basada en citas (CbPD) [21] se basa en el análisis de citas, y es el único enfoque para la detección de plagio que no se basa en la similitud textual. [22] CbPD examina la información de citas y referencias en textos para identificar patrones similares en las secuencias de citas. Como tal, este enfoque es adecuado para textos científicos u otros documentos académicos que contienen citas. El análisis de citas para detectar el plagio es un concepto relativamente joven. No ha sido adoptado por software comercial, pero existe un primer prototipo de sistema de detección de plagio basado en citas. [23] El orden similar y la proximidad de las citas en los documentos examinados son los principales criterios utilizados para calcular similitudes de patrones de citas. Los patrones de citas representan subsecuencias que contienen citas no exclusivas compartidas por los documentos comparados. [22] [24] Los factores, incluido el número absoluto o la fracción relativa de citas compartidas en el patrón, así como la probabilidad de que las citas coincidan en un documento también se consideran para cuantificar el grado de similitud de los patrones. [22] [24] [25] [26]

Estilometría [\[editar\]](#)

La estilometría incluye métodos estadísticos para cuantificar el estilo de escritura único de un autor [27] [28] y se usa principalmente para la atribución de autoría o CaPD intrínseca. Al construir y comparar modelos estilométricos para diferentes segmentos de texto, se pueden detectar pasajes que son estilísticamente diferentes de otros, por lo tanto potencialmente plagiados. [5]

Rendimiento [\[editar\]](#)

Las evaluaciones comparativas de los sistemas de detección de plagio [3] [29] [30] [31] [32] [33] indican que su desempeño depende del tipo de plagio presente (ver figura). Excepto por el análisis del patrón de citas, todos los enfoques de detección se basan en la similitud textual. Por lo tanto, es sintomático que la precisión de detección disminuya cuanto más casos de plagio se ofuscan.



Las copias “textuales”, también conocidas como copiar y pegar o casos de plagio disfrazados pueden ser detectados con alta precisión por el software de detección de plagio si se tiene acceso a la fuente. Se obtienen buenos resultados especialmente en los procedimientos de similitud de textos breves ya que comúnmente usan modelos de documentos sin pérdida, como árboles de sufijos. El rendimiento del software que utiliza análisis de huellas dactilares o bolsas de palabras en la detección de copias depende de la pérdida de información incurrida por el modelo del documento utilizado. Mediante estrategias flexibles de fragmentación y selección, el software es más capaz de detectar formas leves de plagio disfrazado en comparación con los procedimientos de similitud de textos cortos.

La detección de plagio mediante el uso de estilometría puede mejorar la similitud textual hasta cierto punto comparando la similitud lingüística. Dado que las diferencias estilísticas entre los segmentos plagiados y originales son significativas se pueden identificar de manera confiable. La estilometría puede ayudar a identificar plagio de textos disfrazados y parafraseados. Es probable que las comparaciones estilométricas fallen en los casos en los que los segmentos están fuertemente parafraseados hasta el grado en el que se asemejan más al estilo de redacción personal del plagiario o si un texto fue compilado por múltiples autores. Los resultados de las Competencias Internacionales de Detección de Plagio celebradas en 2009, 2010 y 2011, [3] [32] [33], así como los experimentos realizados por Stein, [34] indican que el análisis estilométrico parece funcionar de manera confiable solo para documentos con longitud no mayor a decenas de miles de palabras, lo que limita la aplicabilidad del método a la configuración de la detección de plagio asistido por software.

Cada vez se realizan más investigaciones sobre métodos y sistemas capaces de detectar plagios traducidos. Actualmente, la detección de plagio en varios idiomas (CLPD, por sus siglas en inglés) no se considera una tecnología robusta [35] y los sistemas actuales no han logrado resultados satisfactorios de detección en la práctica. [31]

La detección de plagio basada en citas mediante el análisis de patrones de citas es capaz de identificar paráfrasis más complejas y traducciones con mayores tasas de éxito en comparación con otros enfoques de detección, porque es independiente de las características textuales. [22] [25] Sin embargo, dado que el análisis del patrón de citas depende de la disponibilidad de información de las mismas, se limita sólo a textos académicos. Este enfoque sigue siendo inferior a los basados en texto para detectar plagio de textos más cortos, que son típicos de los tipo de casos de plagio de copiar y pegar o de mezclar y pegar; éste último referido a la mezcla de fragmentos levemente alterados de diferentes fuentes. [36]

Software [\[editar\]](#)

El diseño del software de detección de plagio para su uso en documentos de texto se caracteriza por una serie de criterios: [es necesario citar]

Criterio	Descripción y alternativas
Alcance de la búsqueda	En internet abierto usando motores de búsqueda, bases de datos institucionales o cerradas, bases de datos específicas. [es necesario citar]
Tiempo de análisis	Tiempo que transcurre entre el momento en que se envía un documento y el momento en que el reporte de resultados está disponible. [es necesario citar]
Capacidad de procesamiento	Cantidad de documentos que el sistema puede procesar en un cierto periodo de tiempo. [es necesario citar]
Control de intensidad	¿Con qué frecuencia y para qué tipos de textos y documentos (párrafos, oraciones, secuencias de palabras, etc.) el sistema compara con las fuentes externas como los motores de búsqueda?
Tipo de algoritmo de comparación	Los algoritmos que definen la forma que el software usa para comparar los documentos entre sí. [es necesario citar]
Precisión y registro	El número de documentos identificados correctamente como plagiados en comparación con el número total de documentos analizados y con el número total de documentos que realmente fueron plagiados. La precisión significa que se encontraron pocos positivos falsos, y el alto registro significa que pocos negativos falsos se dejaron de detectar. [es necesario citar]

La mayoría de los sistemas de detección de plagio a gran escala utilizan bases de datos internas grandes (además de otros recursos) que continúan creciendo con cada documento adicional que es enviado para su análisis. Sin embargo, esta característica es considerada por algunos como una violación a los derechos de autor de los estudiantes. [es necesario citar]

Código de programación [\[editar\]](#)

El plagio de códigos de programación también es frecuente y exige el uso de herramientas diferentes a las utilizadas para las comparaciones de texto en documentos. Se han realizado importantes investigaciones sobre el plagio de código de programación en el contexto académico. [37]

Un aspecto distintivo del plagio de código de programación es que no hay “fábricas de código” (el equivalente a los “essay mills” o “vendedores de ensayos”), como sí los hay en el plagio de textos y documentos tradicionales. Como en la mayoría de las tareas y proyectos de programación se espera que los estudiantes escriban códigos con instrucciones muy específicas, es muy difícil encontrar programas ya existentes que las cumplan. Y debido a que el uso de código ajeno en programas propios es comúnmente más complejo que escribir desde cero, la mayoría de los estudiantes que plagian lo hacen pero de programas de sus compañeros.

De acuerdo con Roy y Cordy, los algoritmos de identificación de similitud en códigos de programación pueden clasificarse de la siguiente forma:

- Cadenas de texto – busca coincidencias textuales de segmentos, por ejemplo, de oraciones de cinco palabras. Rápido, pero puede confundirse al cambiar el nombre de las variables.
- Tokens – de secuencias de caracteres, pero usando un “lexer” para traducir el programa en tokens primero. Esto descarta espacios en blanco, comentarios y nombres de las variables, lo que hace que el algoritmo sea más robusto a los simples reemplazos de texto. La mayoría de los sistemas académicos de detección de texto funcionan a este nivel, utilizando diferentes algoritmos para medir la similitud entre las secuencias de tokens o caracteres.
- [Parse Trees](#) – construyen y comparan árboles de análisis sintáctico. Esto permite que se detecten similitudes de mayor nivel. Por ejemplo, la comparación de árbol puede normalizar declaraciones condicionales y detectar construcciones tanto equivalentes como similares entre sí.
- Gráficos de dependencia de programas (PDGs): un PDG captura el flujo real de control en un programa y permite ubicar equivalencias de niveles mucho más altos, con una mayor complejidad y tiempo de cálculo.
- Metrics – las métricas capturan los “puntuajes” de segmentos de código de acuerdo con criterios específicos como por ejemplo, “el número de ciclos de retroalimentación y condicionales”, o “el número de variables diferentes utilizadas”. Las métricas son fáciles de calcular y se pueden comparar rápidamente, pero también pueden provocar falsos positivos: dos fragmentos con los mismos puntuajes en un conjunto de variables pueden hacer cosas completamente diferentes.
- Enfoques híbridos – por ejemplo, combinar árboles de análisis sintáctico y árboles de sufijos para obtener la capacidad de detección de árboles de análisis sintáctico con la velocidad que permiten los árboles de sufijos, un tipo de estructura de datos de coincidencia de cadenas.

La clasificación del párrafo anterior fue desarrollada para la refactorización de código y no para la detección de plagio académico (un objetivo importante de la refactorización es evitar la redundancia en el código, conocido como código duplicado o clonado en la literatura). Los enfoques anteriores son efectivos a distintos niveles de similitud; la similitud de bajo nivel se refiere a un texto idéntico, mientras que la similitud de alto nivel puede deberse a especificaciones similares. En un entorno académico, cuando se espera que todos los estudiantes codifiquen según las mismas especificaciones, se espera completamente un código funcionalmente equivalente (con similitud de alto nivel), y solo la similitud de bajo nivel se considera prueba razonable de trampa.

Referencias [\[editar\]](#)

1. **Jump up**[^] 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. **Jump up**[^] 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. **Jump up**[^] ^a ^b ^c 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. **Jump up**[^] 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. **Jump up**[^] ^a ^b 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. **Jump up**[^] Bao, Jun-Peng; Malcolm, James A. (2006), "Text similarity in academic conference papers", *2nd International Plagiarism Conference Proceedings* (PDF), Northumbria University Press
7. **Jump up**[^] 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. **Jump up**[^] 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. **Jump up**[^] 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. **Jump up**[^] 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. **Jump up**[^] 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. **Jump up**[^] ^a ^b 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. **Jump up**[^] 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. **Jump up**[^] 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. **Jump up**[^] 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. **Jump up**[^] 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. **Jump up**[^] 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. **Jump up** 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. **Jump up** 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. **Jump up** 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. **Jump up** Gipp, Bela (2014), [Citation-based Plagiarism Detection](#), Springer Vieweg Research, [ISBN 978-3-658-06393-1](#)
 22. **Jump up to: a b c d** 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. **Jump up** Gipp, Bela; Meuschke, Norman; Breiting, 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. **Jump up to: a b** 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. **Jump up to: a b** 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. **Jump up** 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. **Jump up** Holmes, David I. (1998), "The Evolution of Stylometry in Humanities Scholarship", *Literary and Linguistic Computing*, **13** (3): 111–117, [doi:10.1093/lilc/13.3.111](#)
 28. **Jump up** Juola, Patrick (2006), ["Authorship Attribution"](#) (PDF), *Foundations and Trends Information Retrieval*, **1**: 233–334, [doi:10.1561/15000000005](#), [ISSN 1554-0669](#)
 29. **Jump up** [Portal Plagiat - Softwaretest 2004](#) (in German), HTW University of Applied Sciences Berlin, retrieved 6 October 2011
 30. **Jump up** [Portal Plagiat - Softwaretest 2008](#) (in German), HTW University of Applied Sciences Berlin, retrieved 6 October 2011
 31. **Jump up to: a b** [Portal Plagiat - Softwaretest 2010](#) (in German), HTW University of Applied Sciences Berlin, retrieved 6 October 2011
 32. **Jump up to: a b** 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. **Jump up to: a b** 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. **Jump up** 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. **Jump up^** 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. **Jump up^** 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. **Jump up^** "Plagiarism Prevention and Detection - On-line Resources on Source Code Plagiarism". [Higher Education Academy](#), [University of Ulster](#).
38. **Jump up^** Roy, Chanchal Kumar; Cordy, James R. (26 September 2007). "[A Survey on Software Clone Detection Research](#)". School of Computing, [Queen's University](#), Canada.