PG A01: SciencePlag

Automatische Suche nach Plagiaten

Johannes Fischer, Sven Rahmann, Dominik Köppl, Florian Kurpicz

Plagiate

**Nutzen von fremden Inhalten,
ohne die Quellen kenntlich zu machen.**

Plagiate

**Nutzen von fremden Inhalten,
ohne die Quellen kenntlich zu machen.**

Wissenschaftliche Arbeit

2.2 Constructing Suffix Arrays by Induced Sorting

As the basis of our new LCP-array construction algorithm is the *induced sorting* algorithm for constructing suffix arrays [18], we explain this latter algorithm in the following. Induced sorting has a venerable history in suffix sorting, see [9, 14, 22]. Its basic idea is to sort a certain subset of suffixes, either directly or recursively, and then use this result to *induce* the order of the remaining suffixes. In the rest of this section, we follow the presentation of Okanohara and Sadakane [19].

Quellenverzeichnis

18. Nong, G., Zhang, S., Chan, W.H.: Linear suffix array construction by almost pure induced-sorting. In: Proc. DCC, pp. 193–202. IEEE Press, Los Alamitos (2009)
19. Okanohara, D., Sadakane, K.: A linear-time burrows-wheeler transform using induced sorting. In: Karlgren, J., Tarhio, J., Hyyrö, H. (eds.) SPIRE 2009. LNCS, vol. 5721, pp. 90–101. Springer, Heidelberg (2009)

Plagiate

**Nutzen von fremden Inhalten,
ohne die Quellen kenntlich zu machen.**

Wissenschaftliche Arbeit

2.2 Constructing Suffix Arrays by Induced Sorting

As the basis of our new LCP-array construction algorithm is the *induced sorting* algorithm for constructing suffix arrays [18], we explain this latter algorithm in the following. Induced sorting has a venerable history in suffix sorting, see [9, 14, 22]. Its basic idea is to sort a certain subset of suffixes, either directly or recursively, and then use this result to *induce* the order of the remaining suffixes. In the rest of this section, we follow the presentation of Okanohara and Sadakane [19].

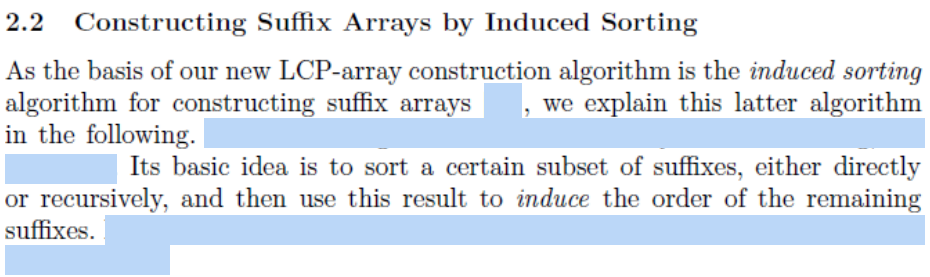
Quellenverzeichnis

18. Nong, G., Zhang, S., Chan, W.H.: Linear suffix array construction by almost pure induced-sorting. In: Proc. DCC, pp. 193–202. IEEE Press, Los Alamitos (2009)
19. Okanohara, D., Sadakane, K.: A linear-time burrows-wheeler transform using induced sorting. In: Karlgren, J., Tarhio, J., Hyyrö, H. (eds.) SPIRE 2009. LNCS, vol. 5721, pp. 90–101. Springer, Heidelberg (2009)

Plagiate

**Nutzen von fremden Inhalten,
ohne die Quellen kenntlich zu machen.**

Wissenschaftliche Arbeit

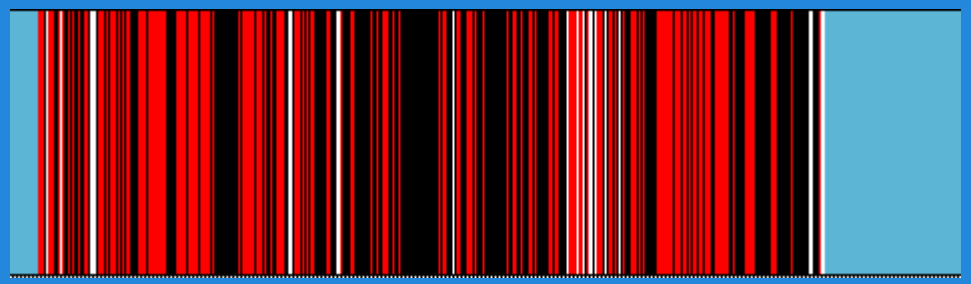


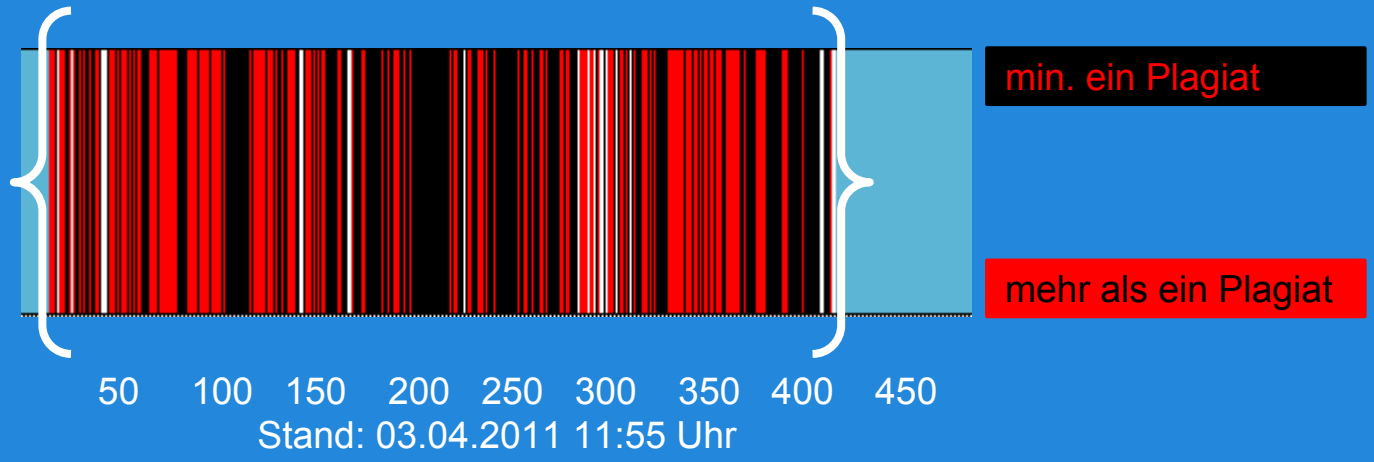
2.2 Constructing Suffix Arrays by Induced Sorting

As the basis of our new LCP-array construction algorithm is the *induced sorting* algorithm for constructing suffix arrays , we explain this latter algorithm in the following. Its basic idea is to sort a certain subset of suffixes, either directly or recursively, and then use this result to *induce* the order of the remaining suffixes.

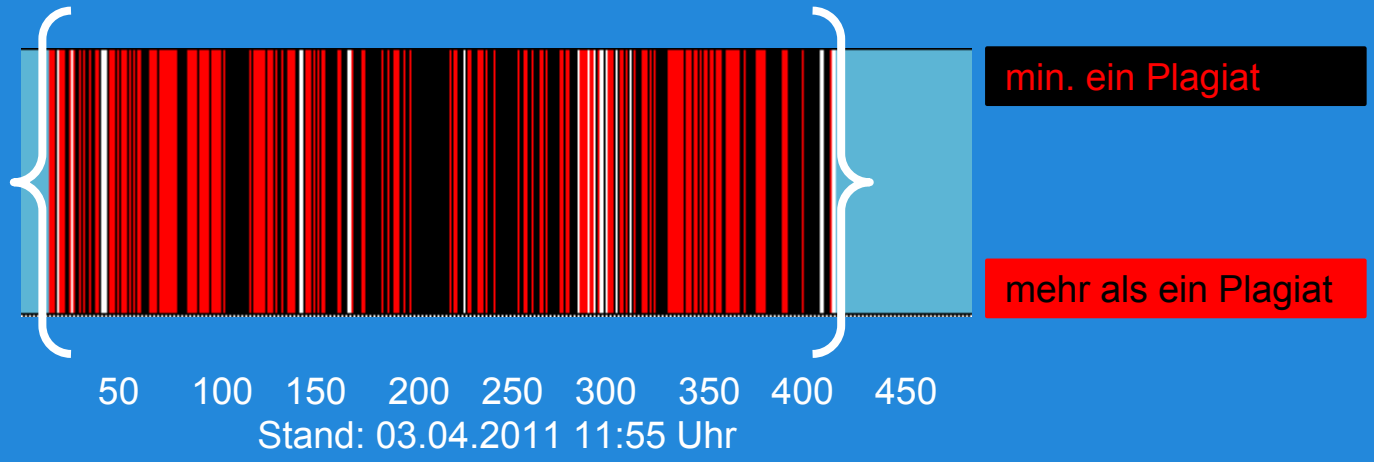
Quellenverzeichnis

http://mathias-middelberg.de/files/2009/09/ZuGuttenbergKommtplakat_neu_jpg.jpg

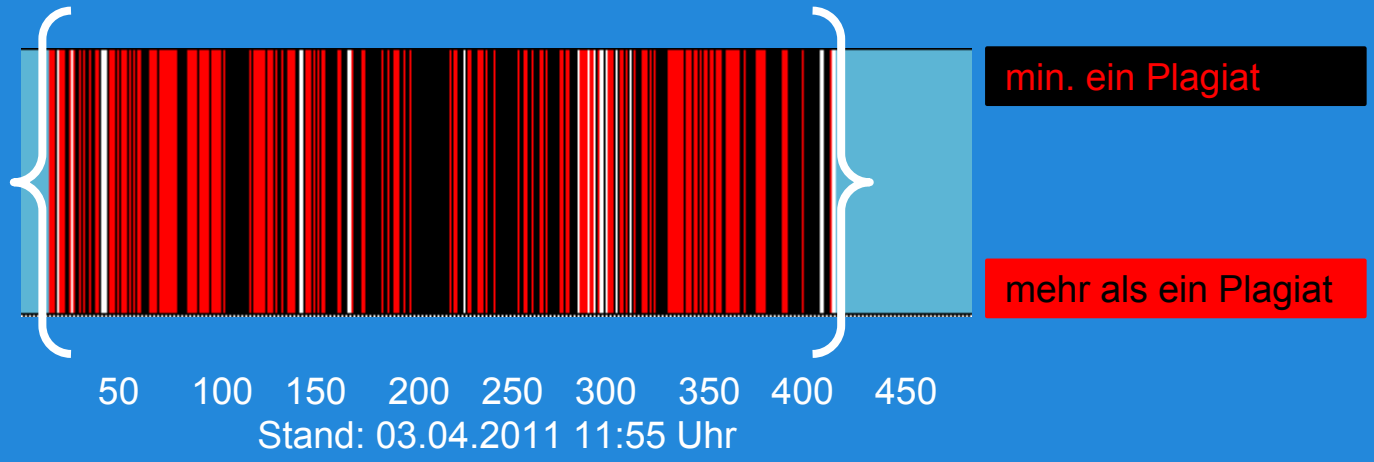

min. ein Plagiat
mehr als ein Plagiat
50
100
150
200
250
300
350
400
450
Stand: 03.04.2011 11:55 Uhr


min. ein Plagiat
mehr als ein Plagiat
50
100
150
200
250
300
350
400
450
Stand: 03.04.2011 11:55 Uhr

- 94,4% der Seiten enthalten mindestens ein Plagiat



- 94,4% der Seiten enthalten mindestens ein Plagiat
- 63,8% der Textzeilen sind plagiiert



135 Quellen auf 371 Seiten mit eigener Farbe

1218 Fragmente mit 10421 Zeilen (63.8%)

Stand: 03.04.2011 11:55 | http://de.guttenplag.wikia.nocookie.net/__cb20110403154534/guttenplag/de/images/8/86/Thumb_xxl.png

135 Quellen auf 371 Seiten mit eigener Farbe
1218 Fragmente mit 10421 Zeilen (63.8%)
Stand: 03.04.2011 11:55 | http://de.guttenplag.wikia.com/wiki/Benutzer:User8

Dr. strg. c. Karl-Theodor (Maria Nikolaus Johann Jacob Philipp Franz Joseph Sylvester) Freiherr von und zu Guttenberg

img2.wikia.nocookie.net/__cb20110403154534/guttenplag/de/images/8/86/Thumb_xxl.png

http://ais.badische-zeitung.de/piece/04/1c/e0/b6/69001398.jpg

- 29,9% der Seiten enthalten mindestens ein Plagiat
- 49,1% der Plagiate wurden aktiv verschleiert

http://schavanplag.wordpress.com/

- 29,9% der Seiten enthalten mindestens ein Plagiat
- 49,1% der Plagiate wurden aktiv verschleiert

http://schavanplag.wordpress.com/

Neben der beschriebenen ontogenetischen Gewissenstheorie gibt Freud auch eine phylogenetische Begründung des Gewissens, die allerdings nur historischen Wert hat. Dazu konstruiert er die Geschichte vom Mord am Vater der Urhorde: Der Vater soll ursprünglich den Besitz aller Frauen beansprucht haben. Aus Haß, der durch dauernden Triebverzicht immer wieder neu verstärkt wurde, töteten die Söhne den Vater. Diese Tat hatte nicht den unbewußt erwarteten Erfolg, weil keiner sich an die Stelle des Vaters setzen konnte.

Freud meint nun, nach der Verwirklichung der Haßbestrebungen sei es zum Wiederauftauchen der unbefriedigten Zärtlichkeitsregungen gegenüber dem ermordeten Vater gekommen. Trauer, Reue und Sehnsucht hatten Schuldgefühle als Urform der Gewissensregung zur Folge. So soll aus dem Ambivalenzkonflikt der Söhne das erste sittliche Gebot "Du sollst nicht töten" entstanden sein. Die Entwicklung weiterer Gebote ist nach Freud auf den fortschreitenden Verzicht der Triebbefriedigung unter dem Druck der Realität zu verstehen [(vgl. dazu: Totem und Tabu.

Diss. Schavan [1980], S. 82

Von dieser ontogenetischen Gewissenstheorie ausgehend, gab FREUD auch eine phylogenetische Begründung für das Auftreten der Sittlichkeit beim Menschen. Aus einer Ära aktiver ethnologischer Forschung heraus […] konstruierte FREUD die Geschichte vom Mord am Vater der Urhorde. Ursprünglich soll der Vater den Besitz aller Frauen beansprucht haben. Die Söhne schritten aus ihrem vom Triebverzicht gespeisten Haß zum Vatermord. Doch hatte diese Tat nicht den unbewußt erwarteten Erfolg, denn keiner konnte sich an die Stelle des Vaters setzen. […] FREUD meint, nach der Verwirklichung der Haßstrebungen sei es zum Wiederauftauchen der unbefriedigten Zärtlichkeitsregungen gegenüber dem ermordeten [Seite 702] Vater gekommen, was Trauer und Reue zur Folge gehabt habe. So soll aus dem Ambivalenzkonflikt der Söhne […] das erste sittliche Gebot "Du sollst nicht töten" hervorgegangen sein. Die weitere Entwicklung der Sittlichkeit stellt sich FREUD als einen fortschreitenden Verzicht auf Triebbefriedigung unter dem Druck der Realität vor. In der gegenwärtigen Psychoanalyse spielt diese phylogenetische Hypothese eine untergeordnete Rolle […].

Original von Häfner [1959], S. 701-702

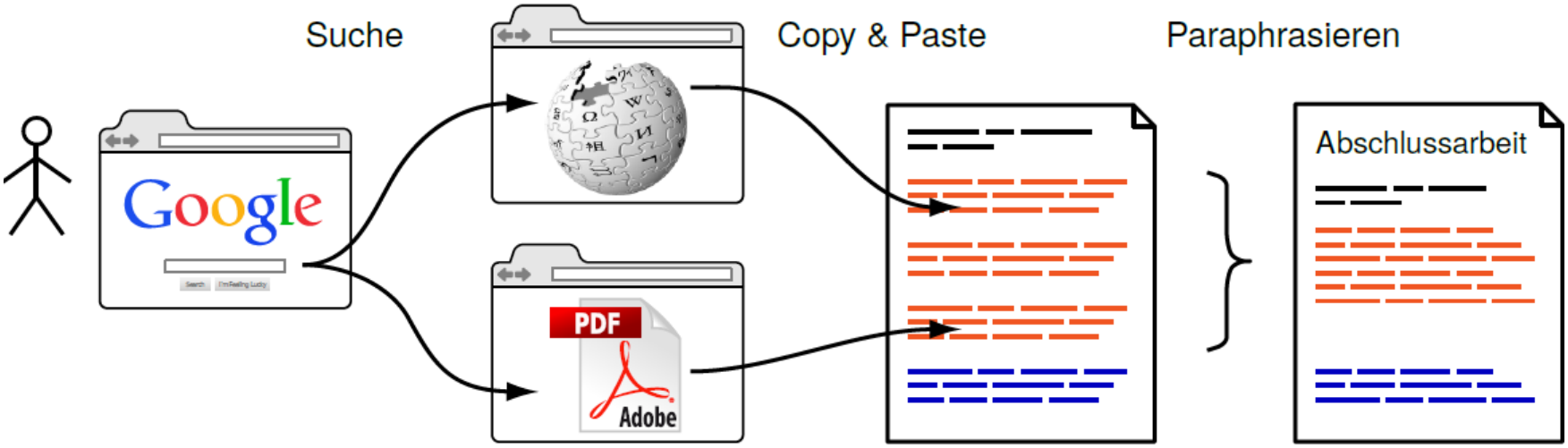
- 29,9% der Seiten enthalten mindestens ein Plagiat
- 49,1% der Plagiate wurden aktiv verschleiert

http://schavanplag.wordpress.com/

"Die Arbeit entsprach damals absolut dem wissenschaftlichen Standard."

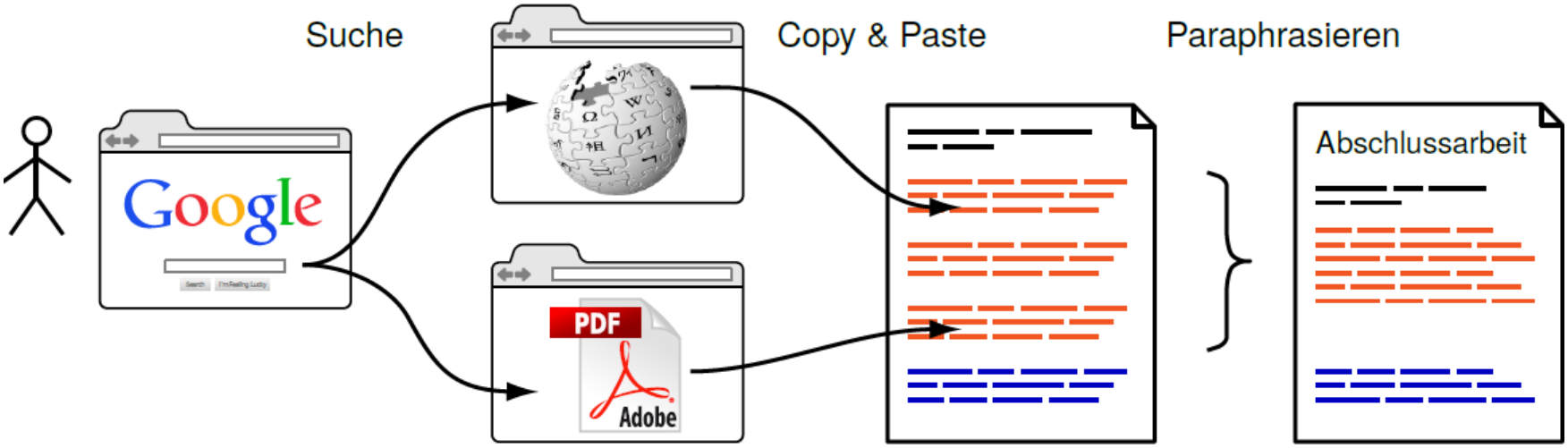
Gerhard Wehle

Wie entstehen Plagiate ?



[Potthast, 2011]

Wie entstehen Plagiate ?

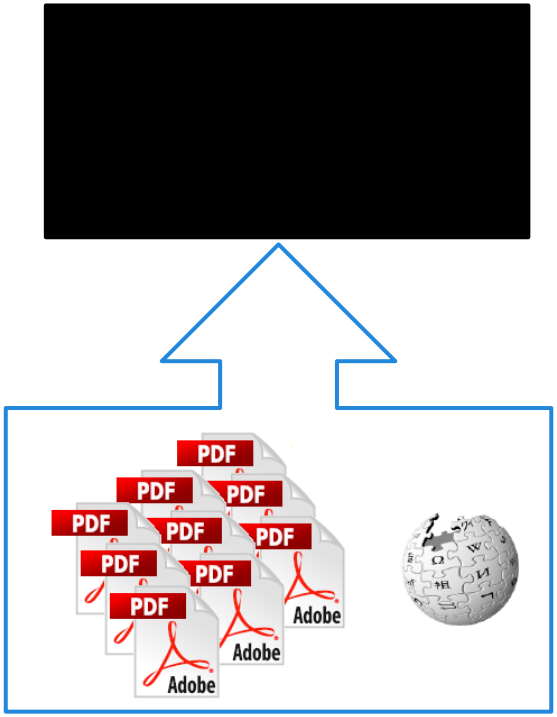


Wie können wir solche Plagiate finden?

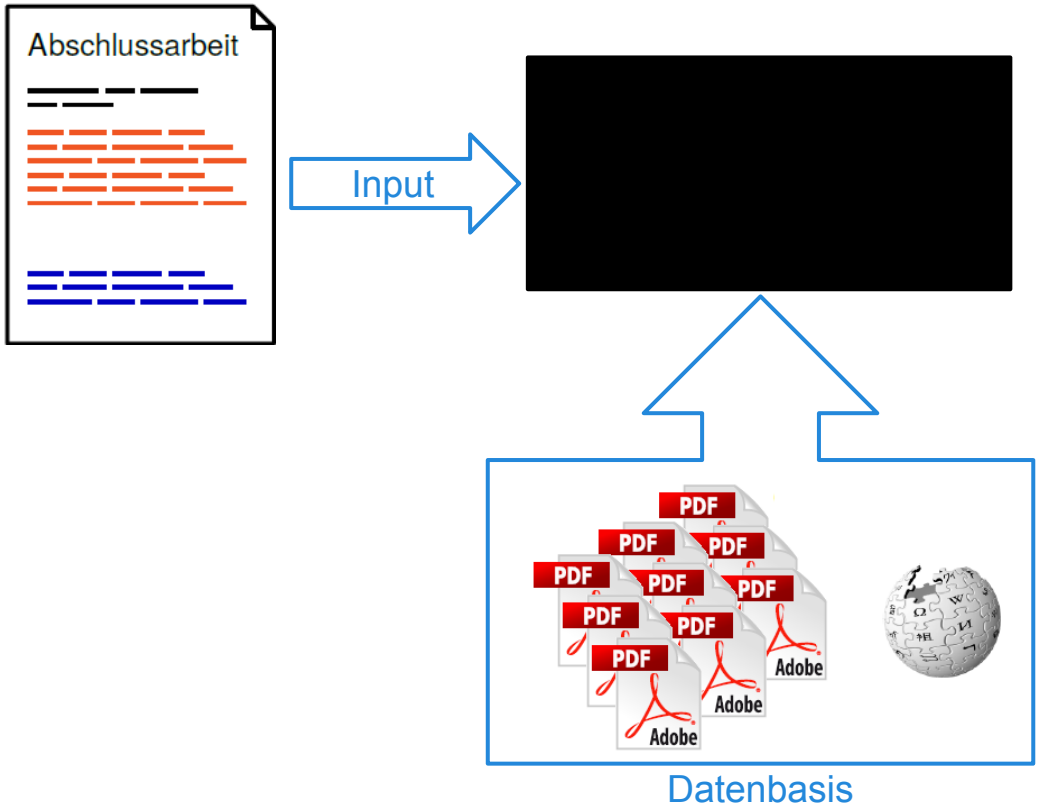
[Potthast, 2011]

Umkehr des Workflows

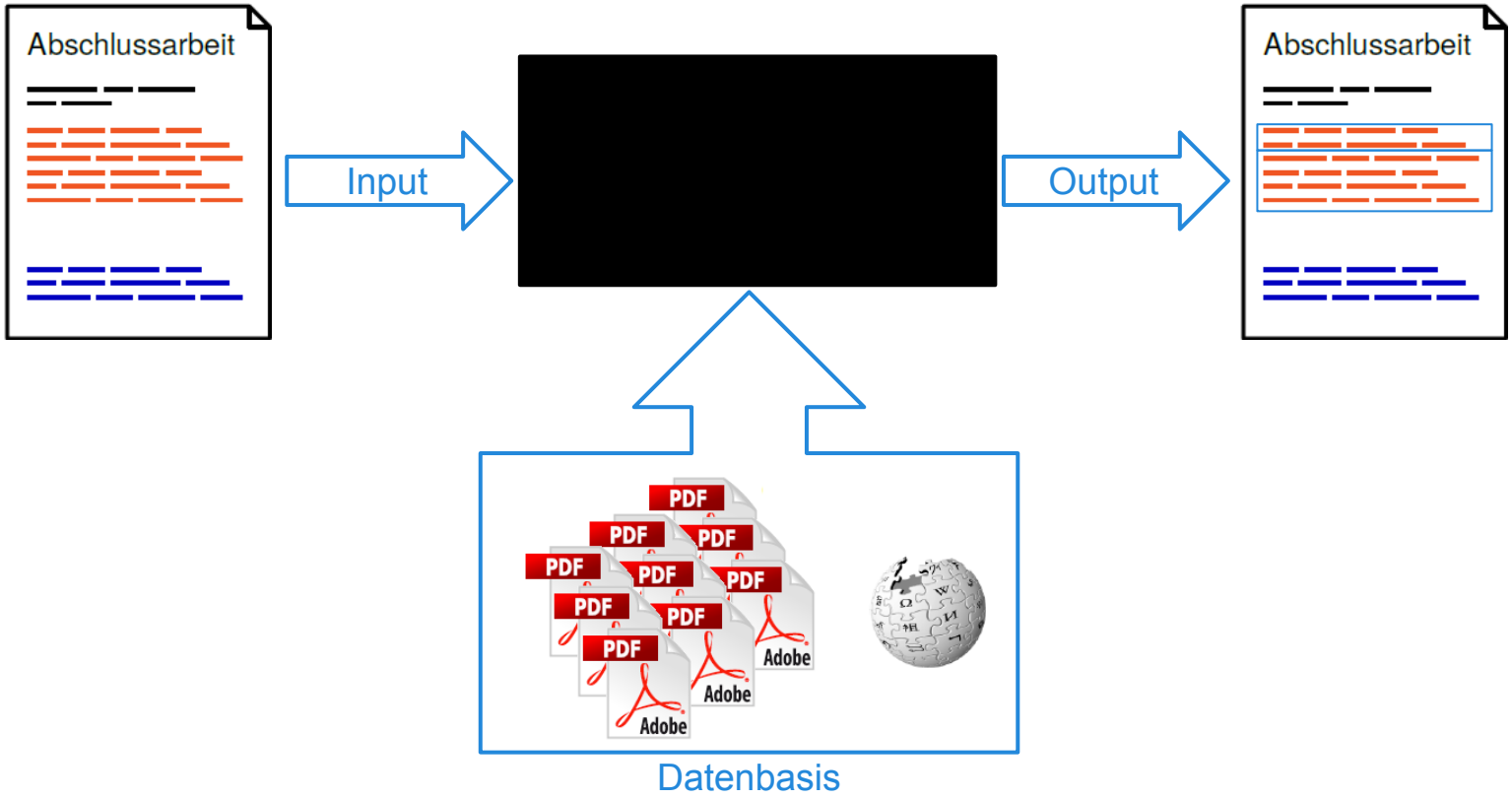
Umkehr des Workflows



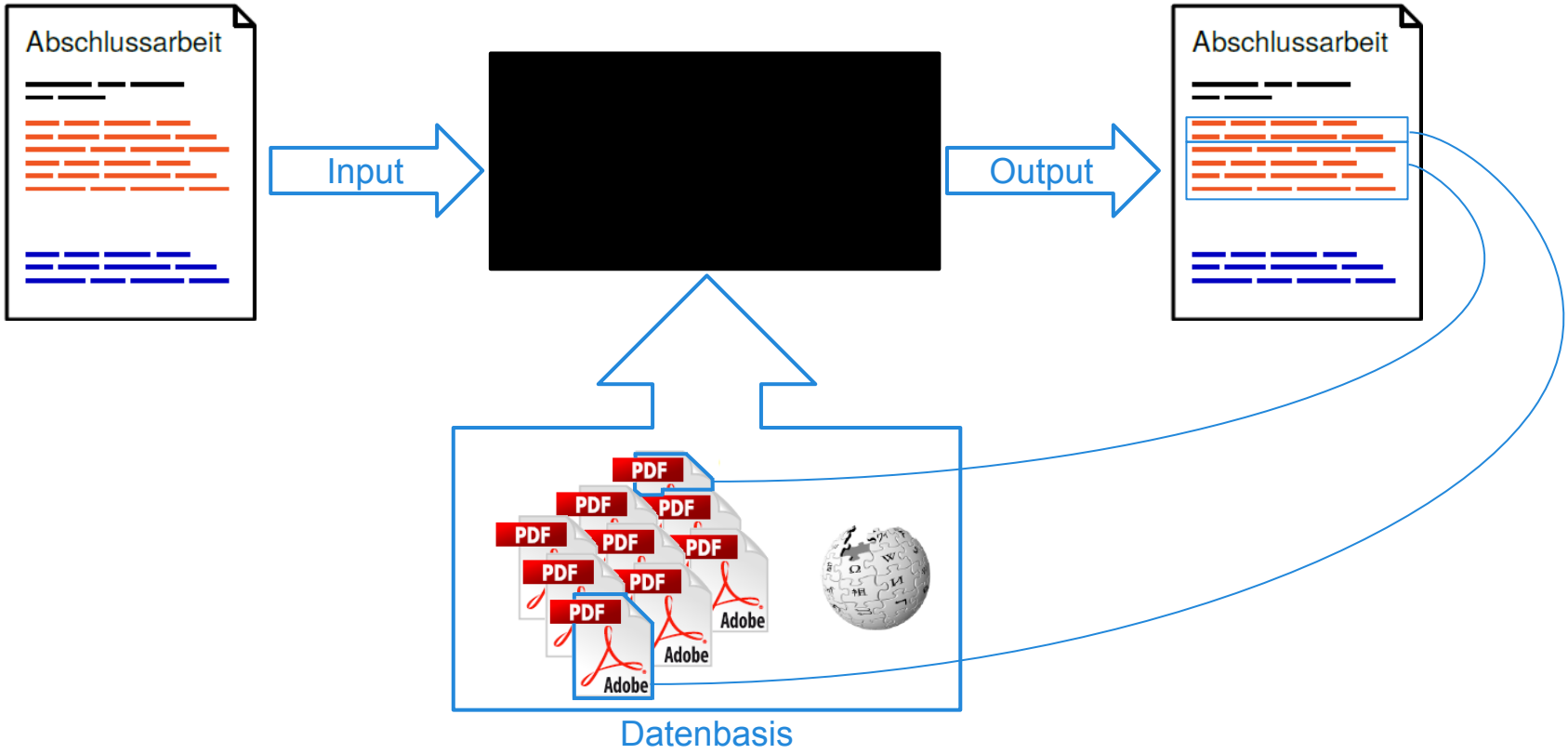
Umkehr des Workflows



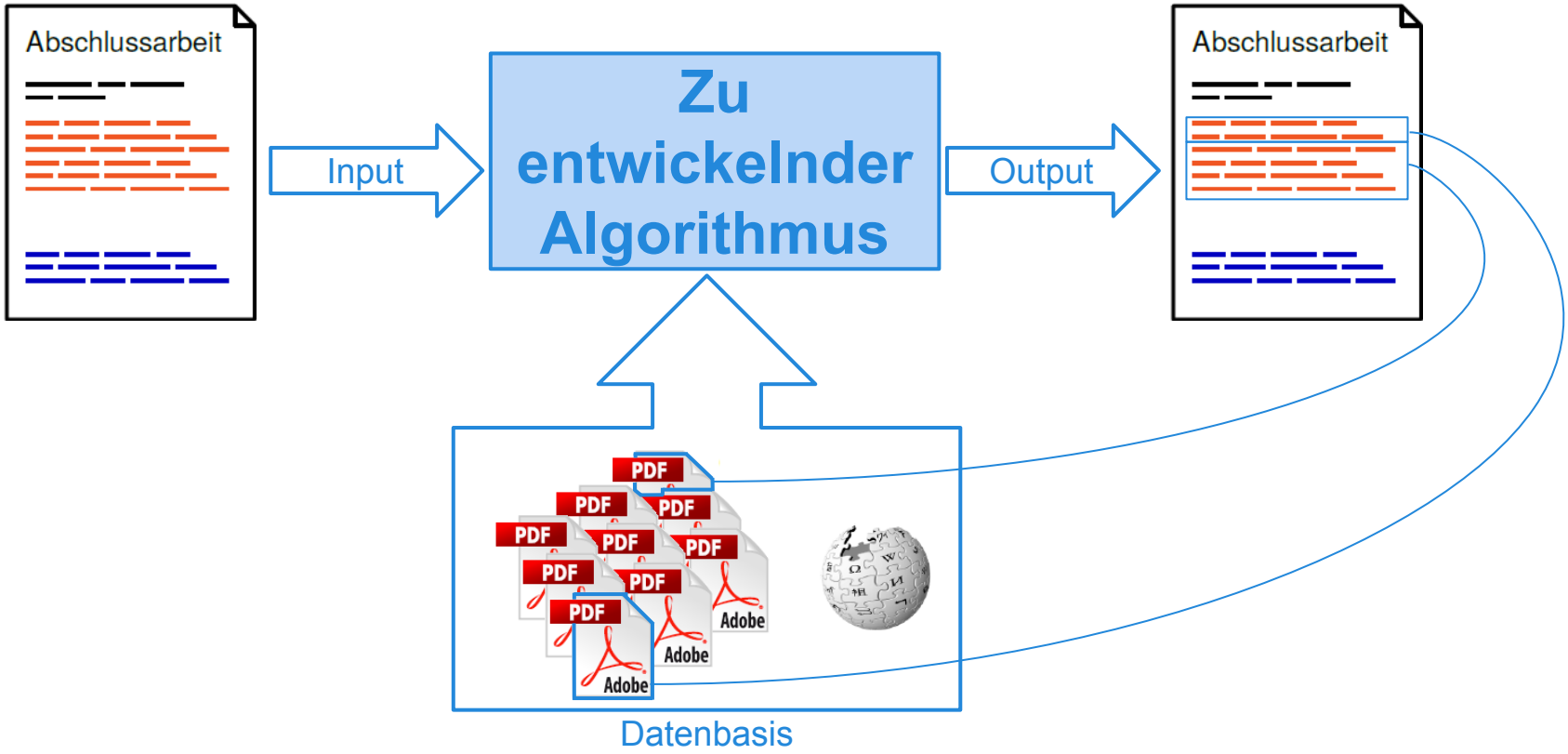
Umkehr des Workflows



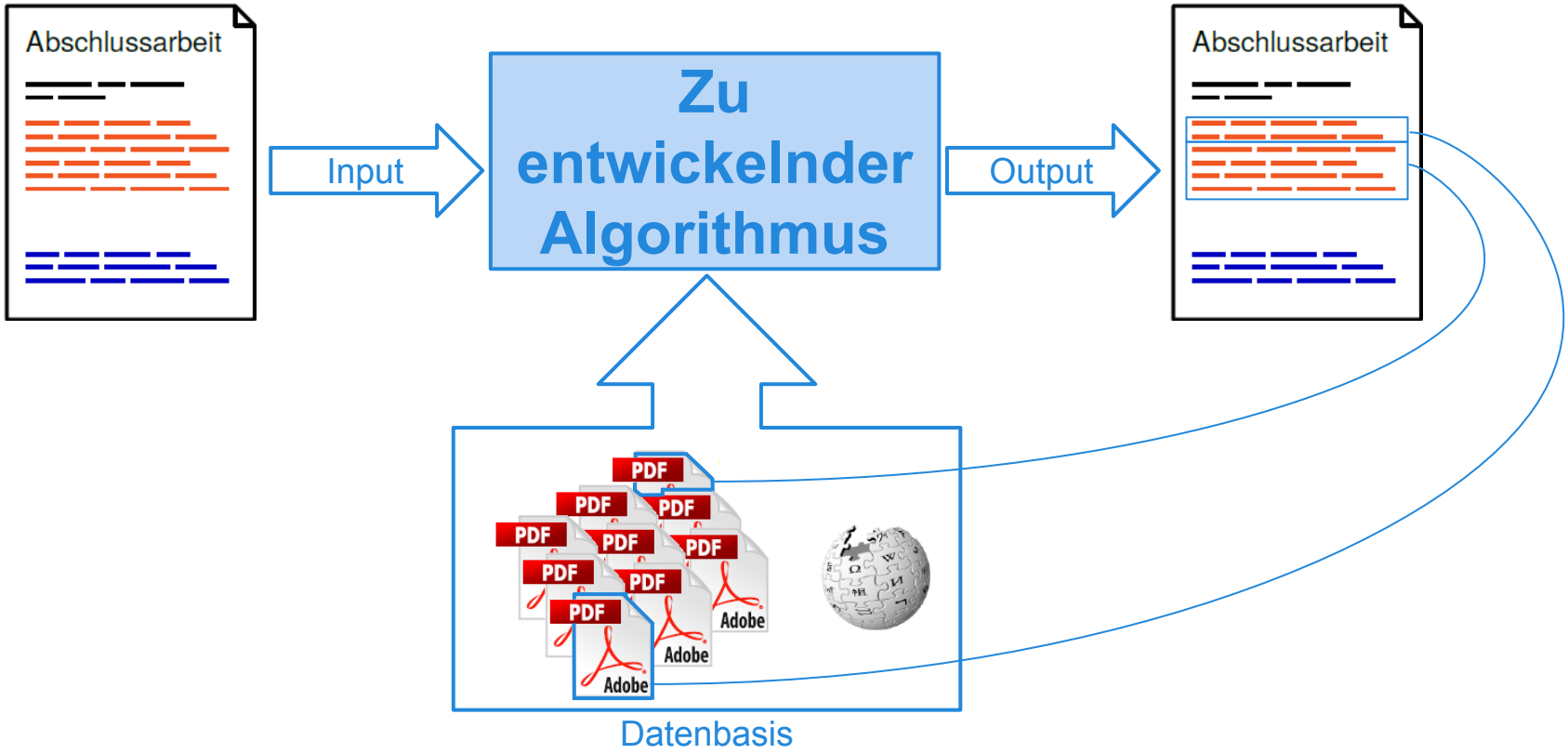
Umkehr des Workflows



Umkehr des Workflows



Umkehr des Workflows



Aber solche Algorithmen gibt es doch schon ?!

Einer der populärsten Ansätze…

…ist **Machine Learning**…

…mit **Information Retrieval**.

Einer der populärsten Ansätze…

…ist **Machine Learning**…

…mit **Information Retrieval**.

Statt Machine Learning setzen wir auf:

Einer der populärsten Ansätze…

…ist **Machine Learning**…

…mit **Information Retrieval**.

Statt Machine Learning setzen wir auf:

Einer der populärsten Ansätze…

…ist **Machine Learning**…

…mit **Information Retrieval**.

Statt Machine Learning setzen wir auf:

Textindexierung

INDEX AND GLOSSARY

Indexes need not necessarily be dry.
— HENRY B. WHEATLEY, *How to Make an Index* (1902)

When an index entry refers to a page containing a relevant exercise, see also the *answer* to that exercise for further information. An answer page is not indexed here unless it refers to a topic not included in the statement of the exercise.

Evaluation des Ansatzes

PAN (http://pan.webis.de/)

- Wettbewerb für Plagiat-Erkennung
- Stellt Testdaten bereit
- Alte Ergebnisse zum Vergleichen
- Mögliche Teilnahme 2015/2016

Was bieten wir?

- Spannendes Thema mit realem Nutzen
  - Neuer Ansatz
  - Teilnahme an der PAN
- Möglichkeit des selbstständigen Forschens
  - Interessante Wege können mit der neuen Herangehensweise erschlossen werden
  - Wie der Ansatz umgesetzt wird liegt bei euch!

Was setzen wir voraus?

- Algorithmische Kompetenz
- gute Programmierkenntnisse in:
  - C/C++ *oder*
  - Java *oder*
  - Python
- Kenntnisse in oder äquivalent zu:
  - Algorithmen auf Sequenzen *oder*
  - Textindexierung und Information Retrieval

ODER die Bereitschaft,
sich diese zur Beginn der PG anzueignen.

Ausführliche Vorstellung...

… mit mehr technischen Details

Morgen (17.12.2014)

14:00 Uhr - 15:00 Uhr

OH14, R202

Ausführliche Vorstellung...

… mit mehr technischen Details

Morgen (17.12.2014)
14:00 Uhr - 15:00 Uhr
OH14, R202

Bis Morgen !

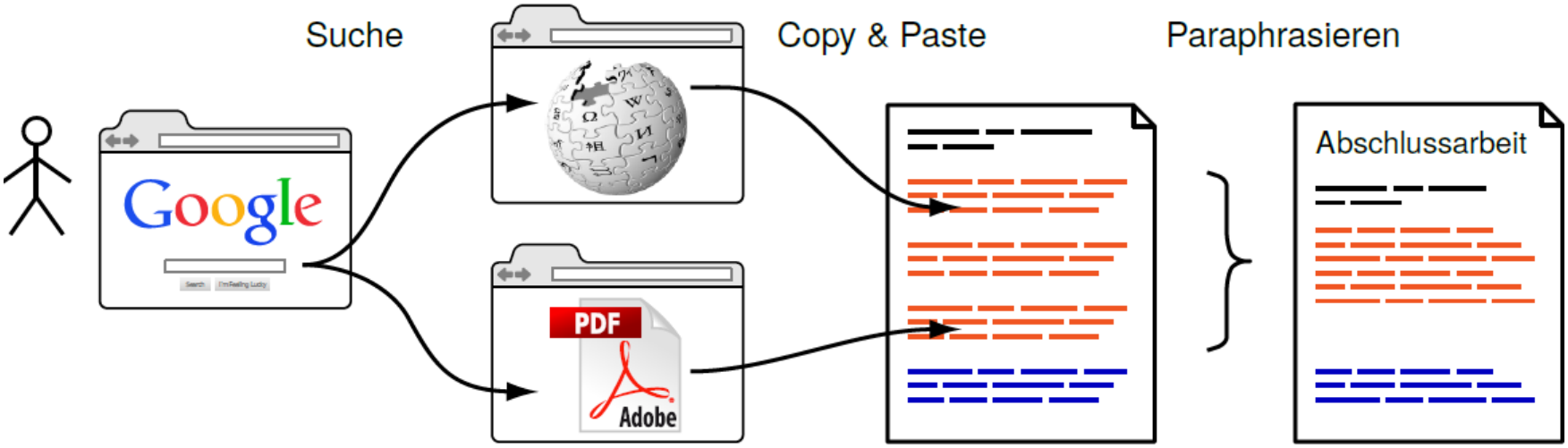
Ausführliche Vorstellung...

… mit mehr technischen Details

Heute (17.12.2014)
14:00 Uhr - 15:00 Uhr
OH14, R202

Willkommen zurück bei
PG A01: SciencePlag

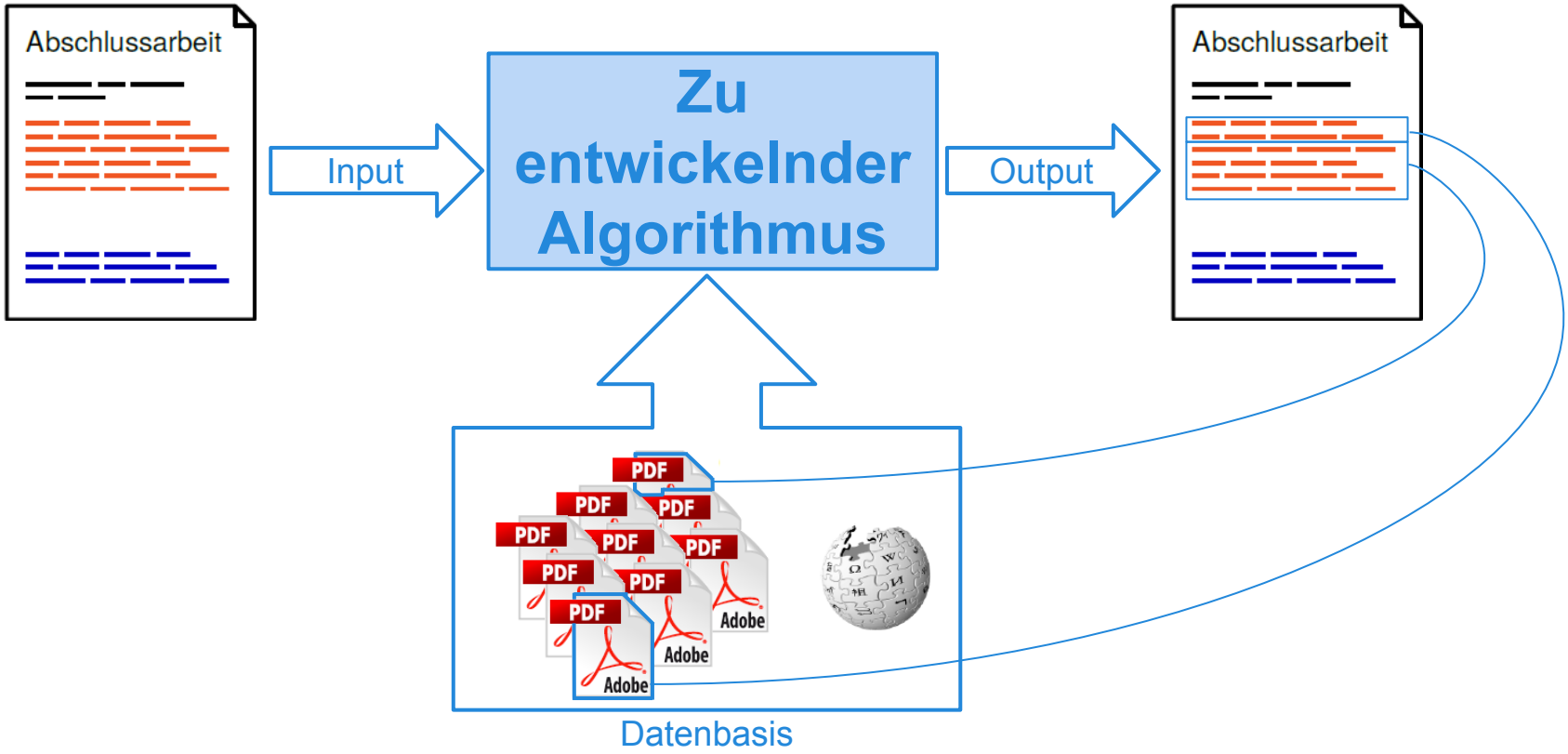
Wie entstehen Plagiate ?



Wie können wir solche Plagiate finden?

[Potthast, 2011]

Umkehr des Workflows



Aber solche Algorithmen gibt es doch schon ?!

Einer der populärsten Ansätze…

…ist **Machine Learning**…

…mit **Information Retrieval**.

Statt Machine Learning setzten wir auf:

Textindexierung

Ein Ansatz für SciencePlag

Information Retrieval

- Vorverarbeitung der Dokumente
  - Stemming
  - Stoppwörter
  - ...

Textindexierung

- Volltext-Index
  - Suffix Array
  - Longest Common Prefix Array
  - ...

Plagiatsprüfung eines Dokuments

- Vorverarbeitung
- “Wo würde das Dokument im Index stehen?”
- Benachbarte Dokumente sind ähnlich.

Das notwendige Übel

i	1	2	3	4	5	6	7	8	9	10	11	12
T =	M	I	S	S	I	S	S	I	P	P	I	$

Sei T ein Text der Länge n.

- Das i-te Suffix (S_i) ist der Teilstring T[i..n]
- Das j-te Präfix (P_j) ist der Teilstring T[1..j]

Suffix Array

i	1	2	3	4	5	6	7	8	9	10	11	12
T =	M	I	S	S	I	S	S	I	P	P	I	$

Das **Suffix Array** enthält die lexikographisch geordneten Indizes aller Suffixe eines Texts.

i	1	2	3	4	5	6	7	8	9	10	11	12
SA	12	11	8	5	2	1	10	9	7	4	6	3

Longest Common Prefix Array (LCP)

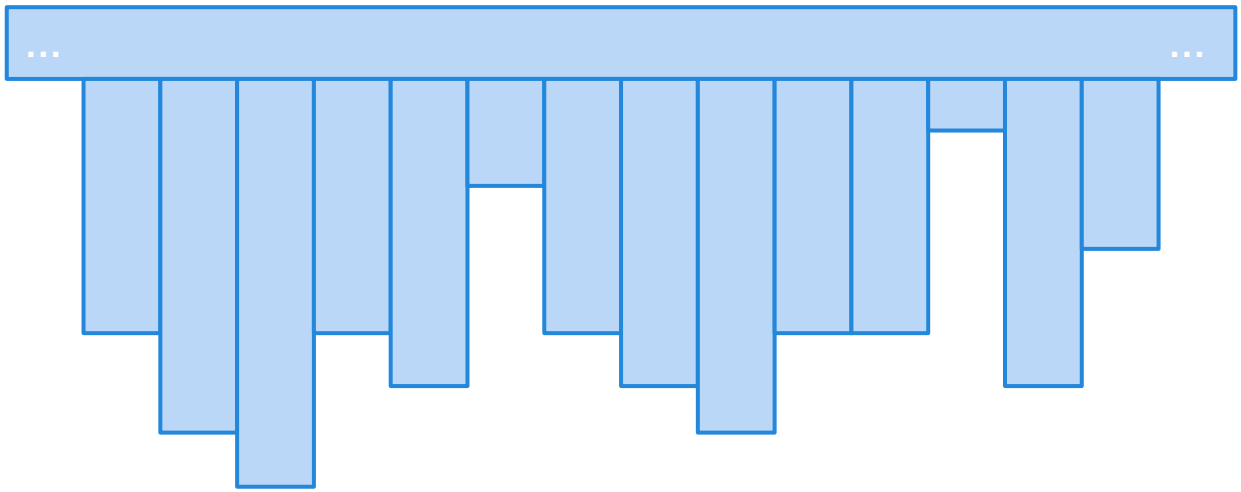
i	1	2	3	4	5	6	7	8	9	10	11	12
T =	M	I	S	S	I	S	S	I	P	P	I	$

Das **LCP** enthält die Größe der LCPs zweier lexikographisch aufeinander folgenden Suffixe.

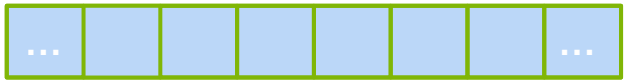
i	1	2	3	4	5	6	7	8	9	10	11	12
SA	12	11	8	5	2	1	10	9	7	4	6	3
LCP	-1	0	1	1	4	0	0	1	0	2	1	3

Plagiate mit SA und LCP finden

Index



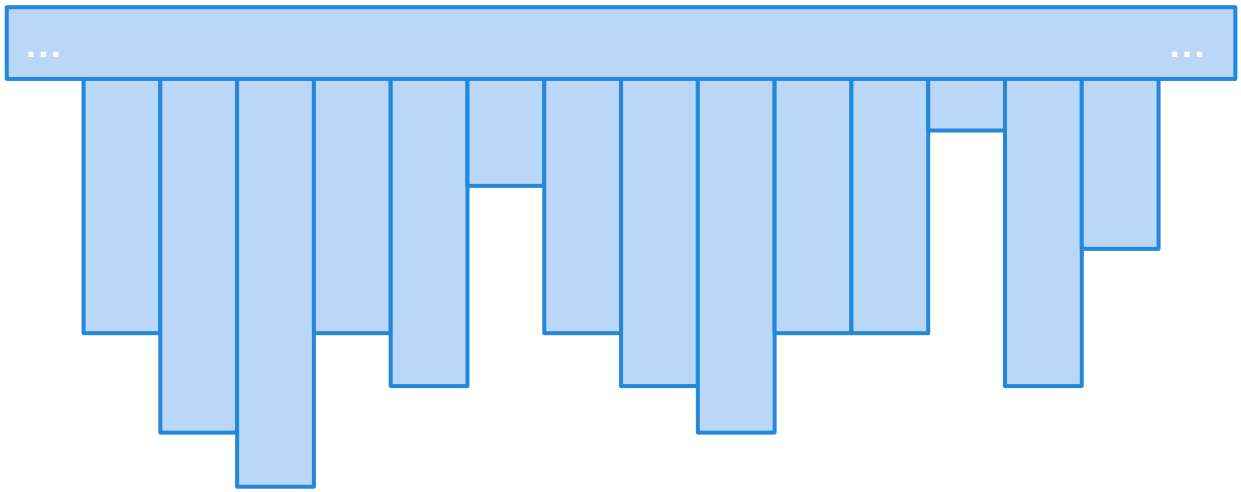
Dokument



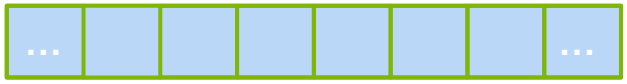
1. Erstelle Index (einmalig)
2. Vorverarbeitung neues Dokument

Plagiate mit SA und LCP finden

Index

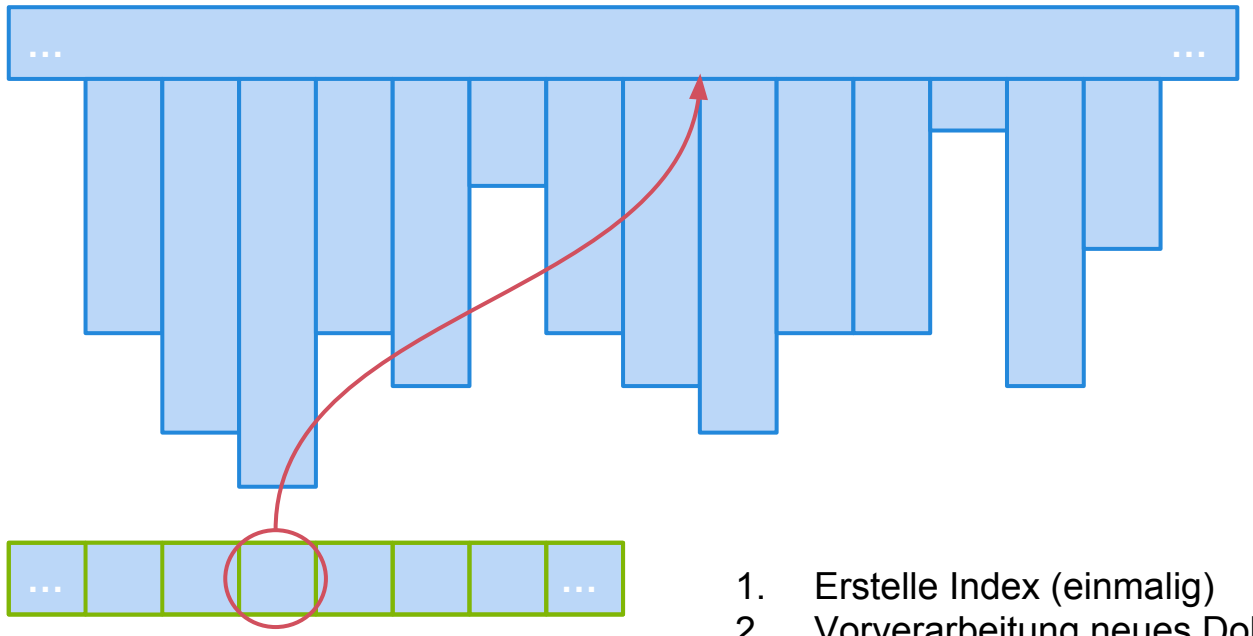


Dokument



1. Erstelle Index (einmalig)
2. Vorverarbeitung neues Dokument
3. An welcher Stelle steht Suffix im Index

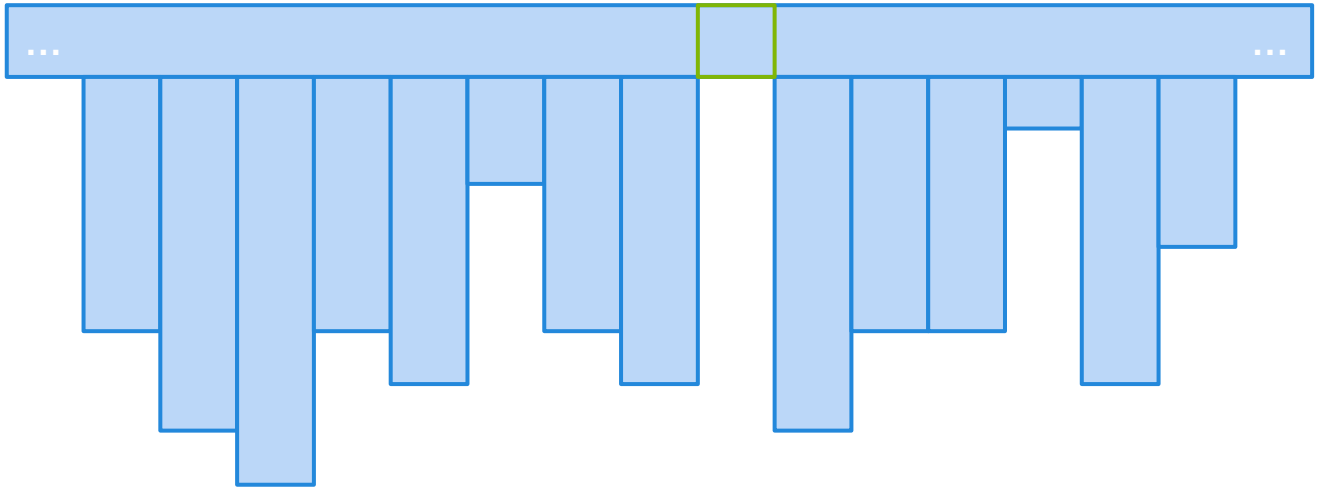
Plagiate mit SA und LCP finden



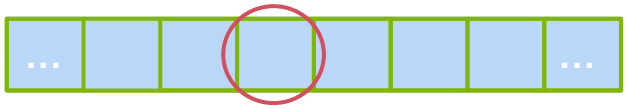
1. Erstelle Index (einmalig)
2. Vorverarbeitung neues Dokument
3. An welcher Stelle steht Suffix im Index

Plagiate mit SA und LCP finden

Index



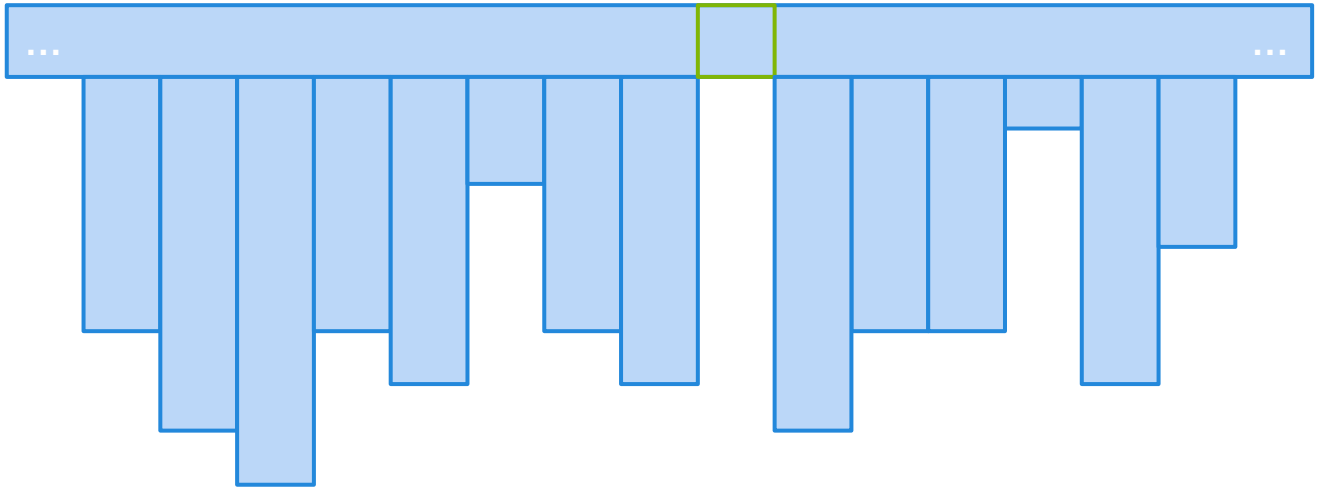
Dokument



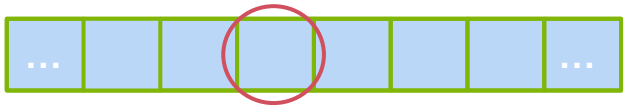
1. Erstelle Index (einmalig)
2. Vorverarbeitung neues Dokument
3. An welcher Stelle steht Suffix im Index

Plagiate mit SA und LCP finden

Index



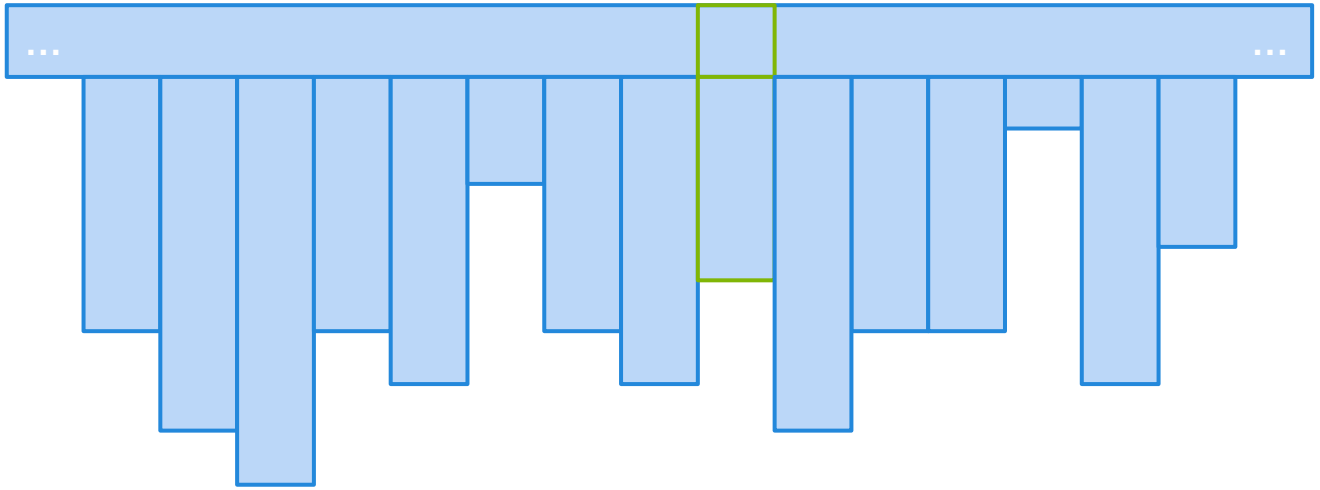
Dokument



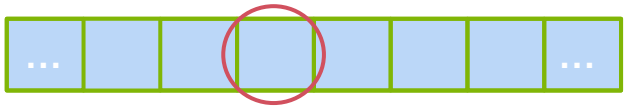
1. Erstelle Index (einmalig)
2. Vorverarbeitung neues Dokument
3. An welcher Stelle steht Suffix im Index
4. Berechne den neuen LCP Wert

Plagiate mit SA und LCP finden

Index



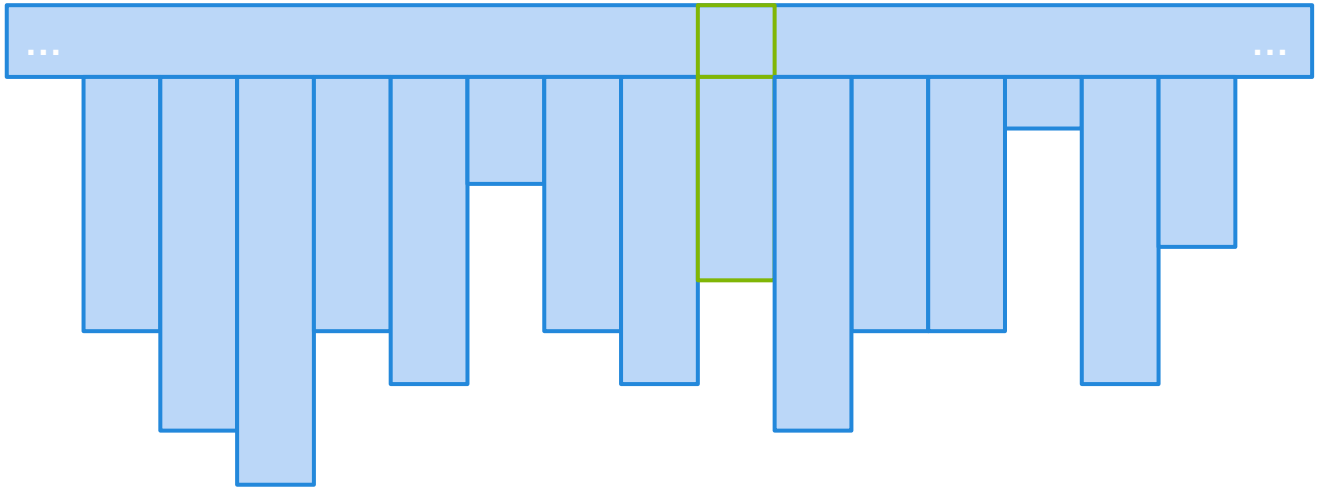
Dokument



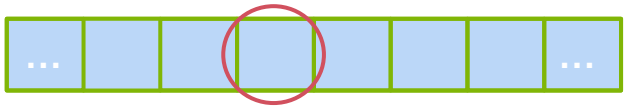
1. Erstelle Index (einmalig)
2. Vorverarbeitung neues Dokument
3. An welcher Stelle steht Suffix im Index
4. Berechne den neuen LCP Wert

Plagiate mit SA und LCP finden

Index



Dokument



1. Erstelle Index (einmalig)
2. Vorverarbeitung neues Dokument
3. An welcher Stelle steht Suffix im Index
4. Berechne den neuen LCP Wert
   - Je größer desto mehr Übereinstimmung

Weitere Probleme

- Unterschiedliche Schreibweisen
- Füllwörter
- Konjunktionen
- “Aktives verschleiern”

Lösung: Methoden des **Information Retrieval**

Stemming

Rückführung des Wortes auf den Wortstamm

n-Gram

- Zerlege den Text in Teilwörter der Länge n
  - T=mississippi, n=2
  - 2-grams: mi, is, ss, si, ip, pp, pi
- Welche Teilwörter können gelöscht werden
  - Im Englischen z.B.: “ed”, “ing”, “ly”, …

Text aufbereiten

Ursprungstext	Hier habe ich plagiiert.
String in Wörter trennen	(Hier, habe, ich, plagiiert)
Stopwörter löschen	(Hier, plagiiert)
Stemming	(Hier, plagiieren)

Diese und viele weitere Probleme
wollen wir zusammen
in der **PG** lösen!

Diese und viele weitere Probleme
wollen wir zusammen
in der **PG** lösen!

Jetzt: **Q&A zu SciencePlag**