

Finding Near-Duplicate Web Pages: A Large-Scale Evaluation of Algorithms

Monika R. Henzinger

Seminar Datenbanken und Information Retrieval

Dozent: Dr. A. Hinneburg

WS 2006/2007

Martin-Luther-Universität Halle-Wittenberg

Vortrag von Heike Stephan

Einführung

Motivation

- Seiten sind oft inhaltlich gleich, aber unterschiedlich formatiert
 - In der Regel interessiert jedoch nur der Inhalt
 - Das Erkennen sehr ähnlicher Internetseiten (Fast-Duplikate) stellt somit eine große Herausforderung für Suchmaschinen dar
 - Es werden zwei Algorithmen vorgestellt, die Fast-Duplikate finden, und ihre Ergebnisse beim Durchlauf auf einer großen Testmenge von Internetseiten verglichen
-
-

Gliederung

- Einführung
 - Motivation
 - Vorbereitende Erklärungen
 - Vorstellung der Algorithmen
 - Experimente
 - Erläuterungen
 - Auswertung von Broder's Algorithmus
 - Auswertung von Charikar's Algorithmus
 - Vergleich beider Algorithmen
 - Der kombinierte Algorithmus
 - Experimente
 - Fazit und Ausblick
-
-

Einführung

Vorbereitende Erklärungen

- Vorverarbeitung der Seiten
 - die meisten HTML-Tags werden entweder ignoriert oder durch Leerzeichen ersetzt
 - jeder alphanumerischen Zeichenkette wird mit einer Hash-Funktion eine Bitfolge (*Token*) zugeordnet, die mit hoher Wahrscheinlichkeit eindeutig ist
 - URLs werden besonders behandelt:
 - gewöhnliche URLs werden an Punkten und Strichen auseinander gebrochen
 - URLs in IMG-Tags werden als ein Term betrachtet, um unterschiedliche Bilder erkennen zu können
-
-

Einführung

Vorbereitende Erklärungen

Definition: **Site**

Die Site einer Seite ist

- (1) der Domain-Name der Seite, wenn er höchstens einen Punkt, d.h. zwei Ebenen hat; oder
- (2) der Domain-Name ohne die Zeichenfolge vor dem ersten Punkt, wenn der Domain-Name mindestens zwei Punkte, d.h. drei oder mehr Ebenen hat.

Bsp: Die Site von www.cs.berkeley.edu/index.html
ist cs.berkeley.edu

Einführung

Vorstellung der Algorithmen

- Die Ergebnisse zweier Algorithmen auf einer Testmenge von Web-Seiten werden untersucht und verglichen
 - 1. Algorithmus: Entwickelt von Broder et al.
 - im Folgenden als Alg. B abgekürzt
 - 2. Algorithmus: Entwickelt von Charikar
 - im Folgenden als Alg. C abgekürzt
 - Beide Algorithmen werden von erfolgreichen Suchmaschinen verwendet
-
-

Einführung

„Algorithmus B“

- Alg. B erstellt zunächst die „Shingles“ („Schindeln“) einer Seite

Definition: Shingle

Ein Shingle ist eine zusammenhängende Teilfolge von Termen/Tokens eines Dokuments D . Für ein Dokument D ist das w -shingling $S(D,w)$ die Menge aller (einzigartigen) Shingles der Größe/Länge w in D .



Einführung

„Algorithmus B“

Bsp.: Das 4-Shingling von
(a,rose,is,a,rose,is,a,rose)



Einführung

„Algorithmus B“

Bsp.: Das 4-Shingling von
(a,rose,is,a,rose,is,a,rose)

ist die Menge

$\{(a,rose,is,a),(rose,is,a,rose),(is,a,rose,is)\}$

(wobei (a,rose,is,a) und (rose,is,a,rose) doppelt
vorkommen, was aber keine Rolle spielt)

Man sieht, dass die Shingles einander
dachziegelartig überlappen.

- Ein Dokument mit n Tokens/Termen erhält $n-k+1$ Shingles der Länge k .
-
-

Einführung

„Algorithmus B“

Definition Ähnlichkeit:

Die Ähnlichkeit $r(A,B)$ zwischen zwei Dokumenten A und B ist der Anteil derjenigen Shingles, die beiden Dokumenten gemeinsam sind, an der Gesamtzahl der Shingles beider Dokumente, also:

$$r(A,B) = |S(A) \cap S(B)| / |S(A) \cup S(B)|$$

- Es werden m für alle Seiten gleiche Hash-Funktionen f_i , $1 \leq i \leq m$, gewählt, die einer Zeichenfolge ein (wahrscheinlich eindeutiges) Token zuordnen. Jedes Shingle einer Seite erhält m Token.
-
-

Einführung

„Algorithmus B“

- Für jede Funktion f_i wird der kleinste Wert pro Seite ausgewählt und in einem m -dimensionalen Vektor zu dieser Seite, dem *Minwert-Vektor*, gespeichert.
 - Es kann gezeigt werden, dass der erwartete Anteil von Einträgen in den Minwert-Vektoren, in denen zwei Dokumente A und B übereinstimmen, gleich demjenigen der einzigartigen Shingles ist, in denen die beiden Dokumente übereinstimmen.
 - Die Ähnlichkeit $r(A,B)$ zwischen Dok. A und Dok. B kann also durch einen Vergleich der Minwert-Vektoren von A und B approximiert werden.
-
-

Einführung

„Algorithmus B“

- Um Platz und Zeit zu sparen, wird der m -dimensionale Min-Vektor zu einem m' -dimensionalen Vektor von *Supershingles* gekürzt:
 - Sei m durch m' teilbar und $\ell = m/m'$. Die Konkatenation der Minwerte $j*\ell, \dots, (j+1)*\ell-1$ ($0 \leq j \leq m'$) wird in eine weitere Token-Funktion eingegeben, das Resultat ist ein *Supershingle*.
-
-

Einführung

„Algorithmus B“

Definition B-Ähnlichkeit:

Die Anzahl identischer Einträge in den Supershingle-Vektoren zweier Seiten ist ihre B-Ähnlichkeit.

Zwei Seiten sind B-ähnlich, wenn ihre B-Ähnlichkeit mindestens 2 ist.

- Gewählte Parameter: $m=84$, $\ell=14$, $m'=6$, $k=8$.
- Größe eines Tokens: 64 bit → Größe des Supershingle-Vektors pro Dokument: $m' \cdot 64 \text{ bit} = 6 \cdot 8 \text{ byte} = 48 \text{ byte}$

Einführung

„Algorithmus C“

- Jedes Token einer Seite wird mit einer für alle Seiten gleichen Projektions-Funktion auf eine zufällige Folge von b Werten aus der Menge $\{-1, 1\}$ abgebildet.
 - Alle diese Projektionen einer Seite werden addiert.
 - Im resultierenden (ebenfalls b -dimensionalen) Vektor werden alle positiven Einträge auf 1 und alle nicht-positiven auf 0 gesetzt.
-
-

Einführung

„Algorithmus C“

Definition C-Ähnlichkeit:

Die C-Ähnlichkeit zwischen zwei Dokumenten A und B ist die Anzahl der Bits, in denen die Projektionen der beiden Dokumente übereinstimmen.

A und B heißen C-ähnlich, wenn ihre C-Ähnlichkeit einen bestimmten Grenzwert t überschreitet.

- Anmerkung: Die Kosinus-Ähnlichkeit zweier Seiten ist proportional zur Anzahl der Bits, in denen die beiden entsprechenden Projektionen übereinstimmen.

Einführung

„Algorithmus C“

- Gewählte Parameter: $b=384 \rightarrow 48$ byte pro Seite werden gespeichert.
 - Ähnlichkeits-Grenzwert $t: 372$
 - wurde so gewählt, dass beide Algorithmen etwa die gleiche Anzahl korrekter ähnlicher Seiten ausgeben, d.h. etwa gleichen Recall haben
 - Abänderung der Vorgehensweise: jeder Bit-String einer Seite wird in 12 nicht-überlappende 4byte-Stücke aufgeteilt. Es wird die C-Ähnlichkeit aller Seiten berechnet, die mindestens ein Stück gemeinsam haben.
-
-

Einführung

Vergleich

- Beide Algorithmen erzeugen den gleichen Dokumentvektor für die gleiche Token-Folge.
 - B berücksichtigt zusammenhängende Tokens, ignoriert aber die Häufigkeit von Shingles.
 - C berücksichtigt die Häufigkeit von Tokens, nicht jedoch die Reihenfolge.
 - Beide Algorithmen können falsche Positive (nicht-ähnliche Seiten als ähnlich ausgegeben) und falsche Negative (ähnliche Seiten als nicht-ähnlich ausgegeben) erzeugen.
-
-

Experimente

Rahmen und Vorarbeiten

- Dokument-Testmenge: 1,6 Milliarden Seiten, durch den Google-Crawler gesammelt
- Vorverarbeitung: Mengen von exakt gleichen Dokumenten wurden auf je einen Repräsentanten reduziert (Reduktion der Ausgangsmenge um ca. 25-30%)



Experimente

Beurteilung der Präzision

- Jeweils eine Stichprobe von B- und C-ähnlichen Paaren wurde von einem Menschen als korrekt, unkorrekt oder unentschieden gemäß folgender Definition eingestuft:
 - Zwei Dokumente sind korrekte Fast-Duplikate, wenn
 - 1) ihr Text sich nur durch eine Session-ID, ein Timestamp, eine Ausführungszeit, eine Nachrichten-ID, einen Besucherzähler, einen Servernamen und/oder ihre URL oder einen Teil dieser unterscheiden; oder/und
 - 2) der Unterschied für Besucher der Seiten unsichtbar ist; oder
 - 3) die Seiten Eintrittsseiten zur selben Site sind (z.B. bei Pornoseiten).
-
-

Experimente

Beurteilung der Präzision

- Zwei Fast-Duplikate werden als nicht-korrekt angesehen, wenn sie sich im Hauptgegenstand der Seite unterscheiden.
 - Ansonsten werden sie als unentschieden bewertet, wobei manche Seiten nicht ausgewertet werden konnten, z.B. weil sie sich automatisch erneuerten oder auf Chinesisch, Koreanisch oder Japanisch waren
 - Die Seiten wurden sowohl visuell als auch mit Hilfe der diff-Operation von Linux auf der Token-Sequenz untersucht.
-
-

Experimente

Auswertung von Broder's Algorithmus

- Der Algorithmus fand 27,4 Mio Dokumente, die zu mindestens einer Seite ähnlich sind.
- Aus den gefundenen Paaren wurde eine Stichprobe von 96 556 Stück entnommen.
- In 91,9% der Fälle gehörten beide Seiten zur gleichen Site.

Experimente

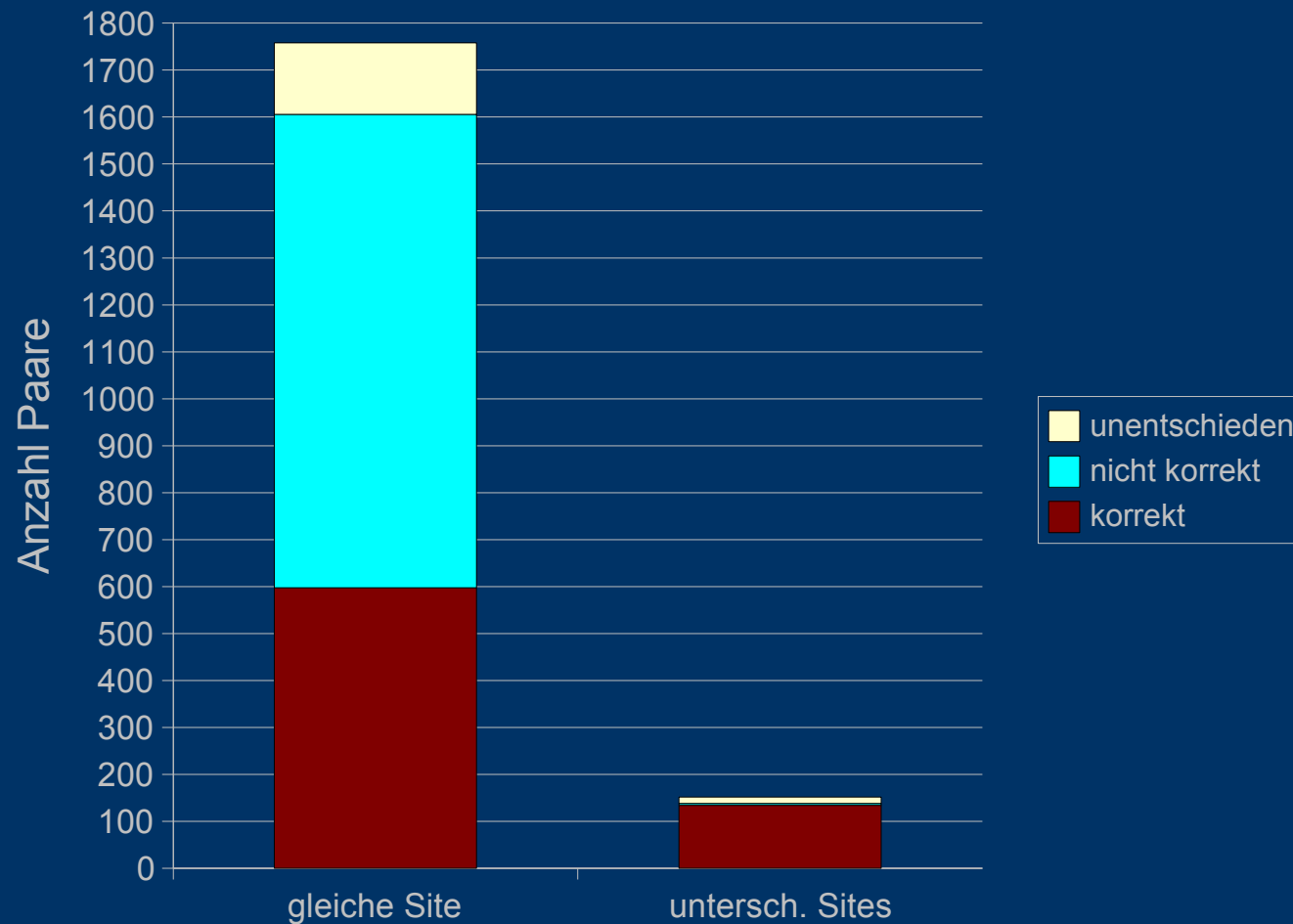
Auswertung von Broder's Algorithmus

- Aus der Stichprobe wurde eine weitere Stichprobe von 1910 Stück entnommen und auf Korrektheit geprüft.
 - Bei Paaren der gleichen Site liegt die Präzision bei 0,34, bei Paaren unterschiedlicher Sites bei 0,86. Grund: Oft benutzen Seiten der gleichen Site den gleichen Text und unterscheiden sich nur durch den Hauptgegenstand in der Seitenmitte.
 - Die Präzision erhöht sich proportional zur B-Ähnlichkeit, wobei weniger als die Hälfte aller Paare eine B-Ähnlichkeit von mindestens 3 haben.
-
-

Experimente

Auswertung von Broder's Algorithmus

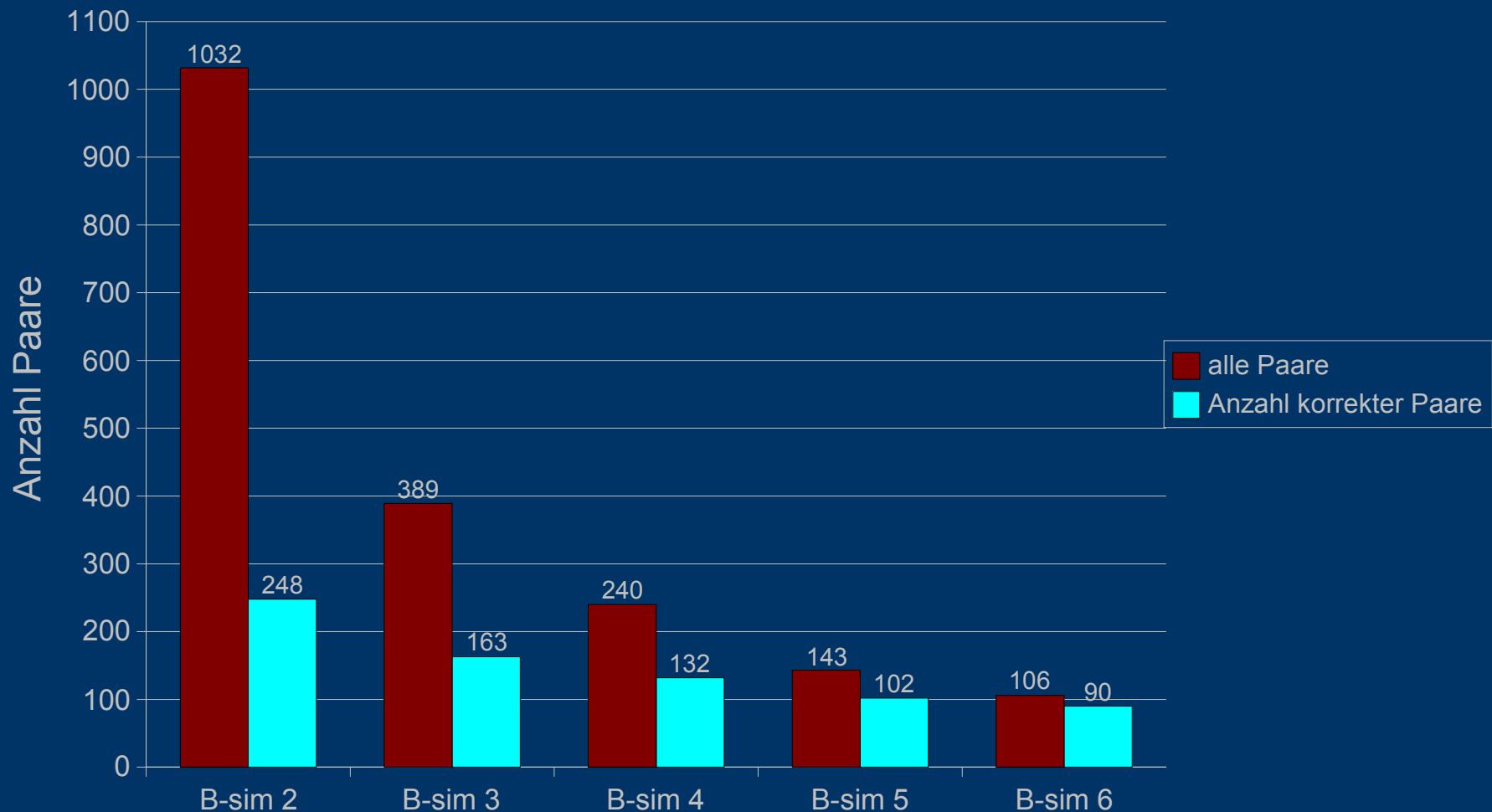
Bewertung aller untersuchten Paare



Experimente

Auswertung von Broder's Algorithmus

Aufschlüsselung der Paare nach B-Ähnlichkeits-Grad



Experimente

Auswertung von Broder's Algorithmus

- Alg. B versagt dann mit höherer Wahrscheinlichkeit, wenn sich zwei Seiten nur um wenige *aufeinander folgende* Token unterscheiden, weil dann mit hoher Wahrscheinlichkeit Supershingles aus dem übereinstimmenden Text gewählt werden.
 - Schätzungsweise 28% aller nicht korrekten Paare stammten von zwei Web-Datenbanken, die sich durch eine kurze zusammenhängende Tokenfolge unterscheiden.
-
-

Experimente

Auswertung von Charikar's Algorithmus

- Der Algorithmus fand 35,5 Mio. Seiten, die mindestens einen Ähnlichkeits-Partner haben, das sind fast 30% mehr als bei Alg. B.
- Es wurde eine Stichprobe von 172 464 Paaren entnommen, davon gehörten 74% zur gleichen Site.

Experimente

Auswertung von Charikar's Algorithmus

- Eine weitere Unterstichprobe im Umfang von 1872 Paaren wurde auf Korrektheit überprüft.
 - Alg. C erreicht eine Präzision von 50%, wobei der Anteil der als unentschieden beurteilten Paare erstaunlich hoch bei 23% liegt (Alg. B: 9%).
 - Für Paare auf unterschiedlichen Sites liegt die Präzision bei 90%, bei gleichen Sites etwa gleich wie die von Alg. B, bei 36% (B: 34%).
-
-

Experimente

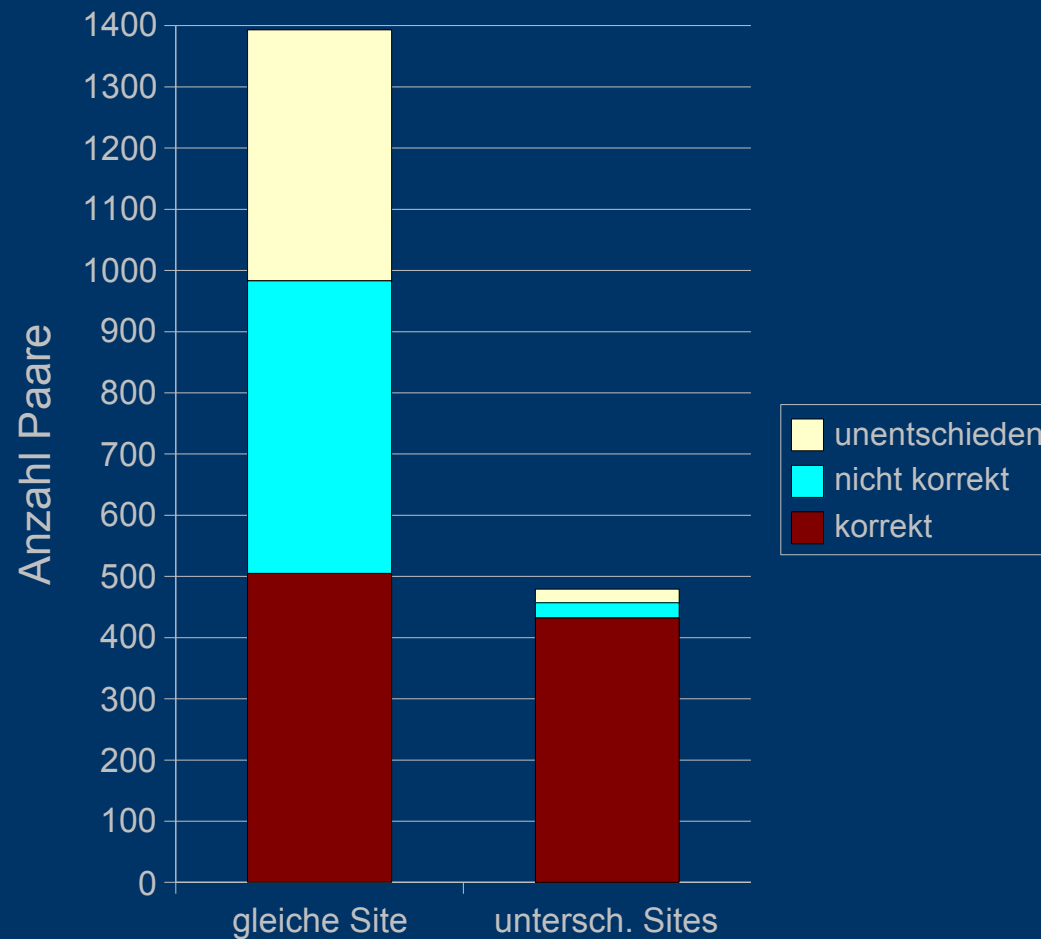
Auswertung von Charikar's Algorithmus

- Die Präzision von Alg. C ist bei den Ähnlichkeits-Schwellenwerten zwischen 372 und 375 am höchsten. Die meisten der Paare unterscheiden sich nur durch die URL, also durch 2-4 Token, wodurch die Ähnlichkeit in den entsprechenden Bereich fällt.
 - Erstaunlicherweise nimmt die Präzision für höhere Schwellenwerte ab.
 - Für die große Zahl unentschiedener Paare ist ein hoher Anteil vorausgefüllter Formulare verantwortlich, die offenbar bei Alg. B keine große Rolle gespielt haben.
-
-

Experimente

Auswertung von Charikar's Algorithmus

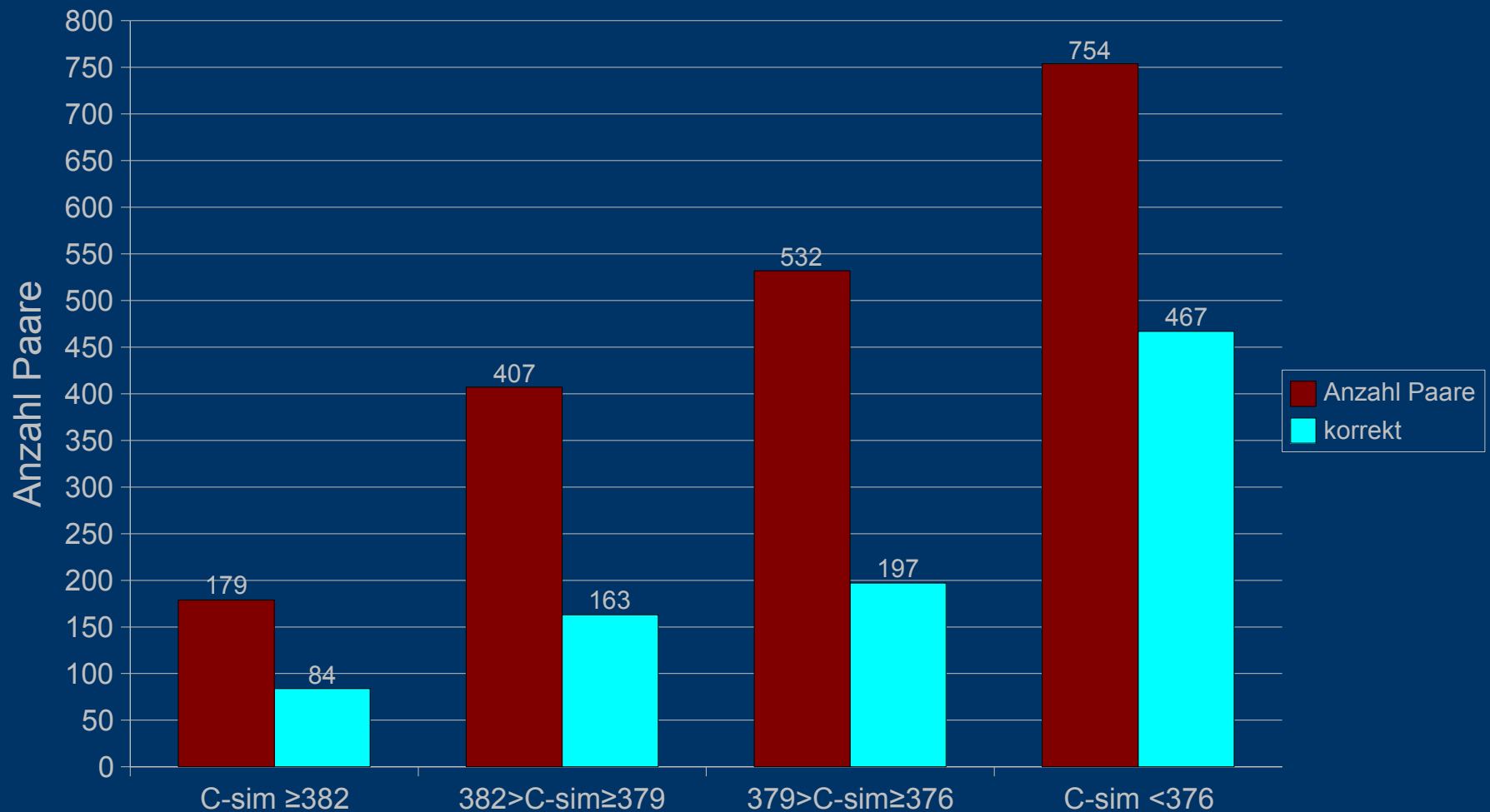
Bewertung aller untersuchten Paare



Experimente

Auswertung von Charikar's Algorithmus

Aufschlüsselung der Paare nach C-Ähnlichkeits-Grad



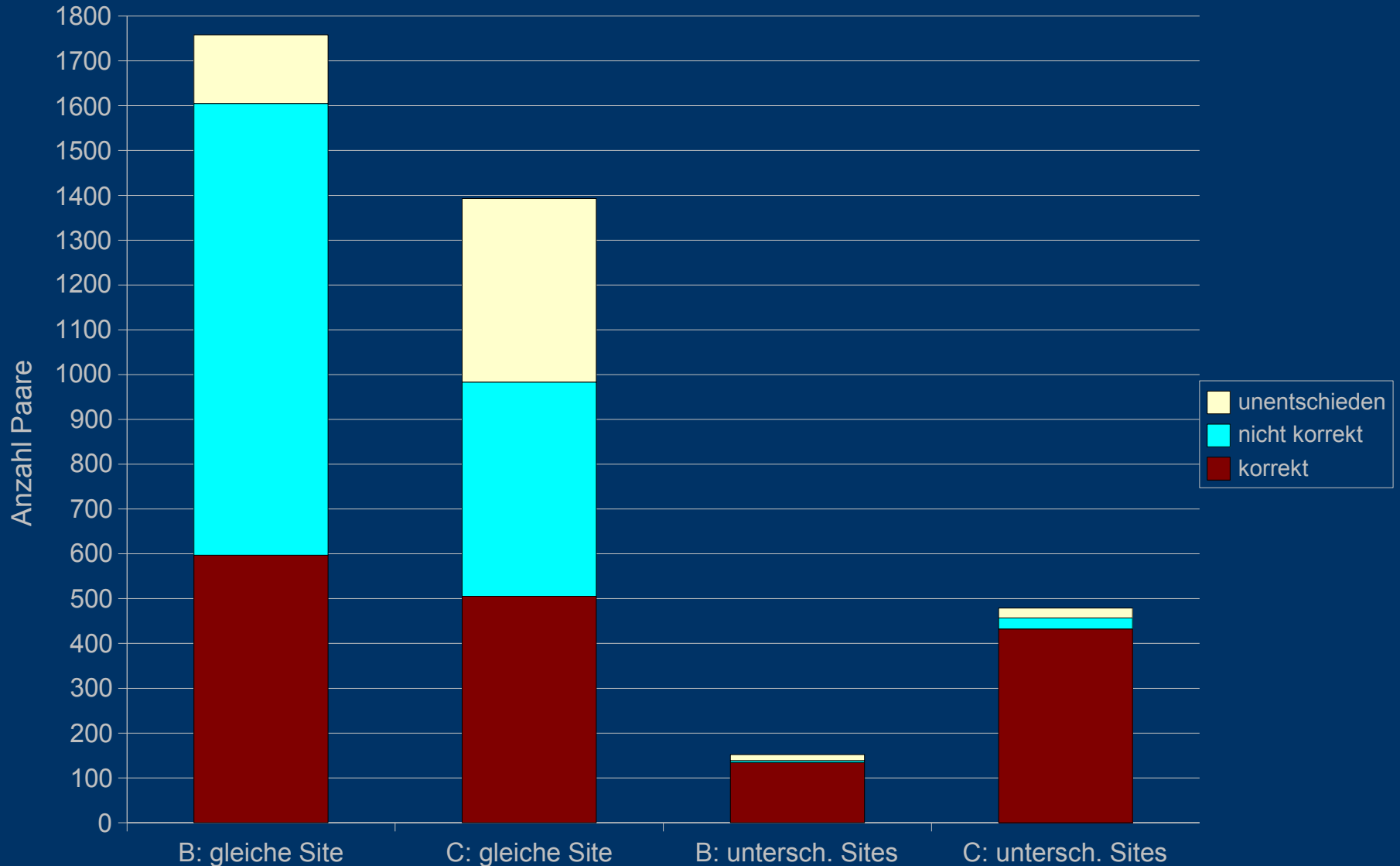
Experimente

Vergleich beider Algorithmen

- Für Paare auf unterschiedlichen Sites ist Alg. C Alg. B deutlich überlegen
 - Für Paare auf der gleichen Site hat Alg. B einen 20% höheren Recall als Alg. C, wohingegen C etwas präziser ist → Kombination beider Algorithmen
 - Alg. C findet sehr viel mehr korrekte Paare, die sich nur durch die URL unterscheiden, und sehr viel mehr Paare, die auf Grund vorausgefüllter Formulare unentschieden bewertet wurden.
-
-

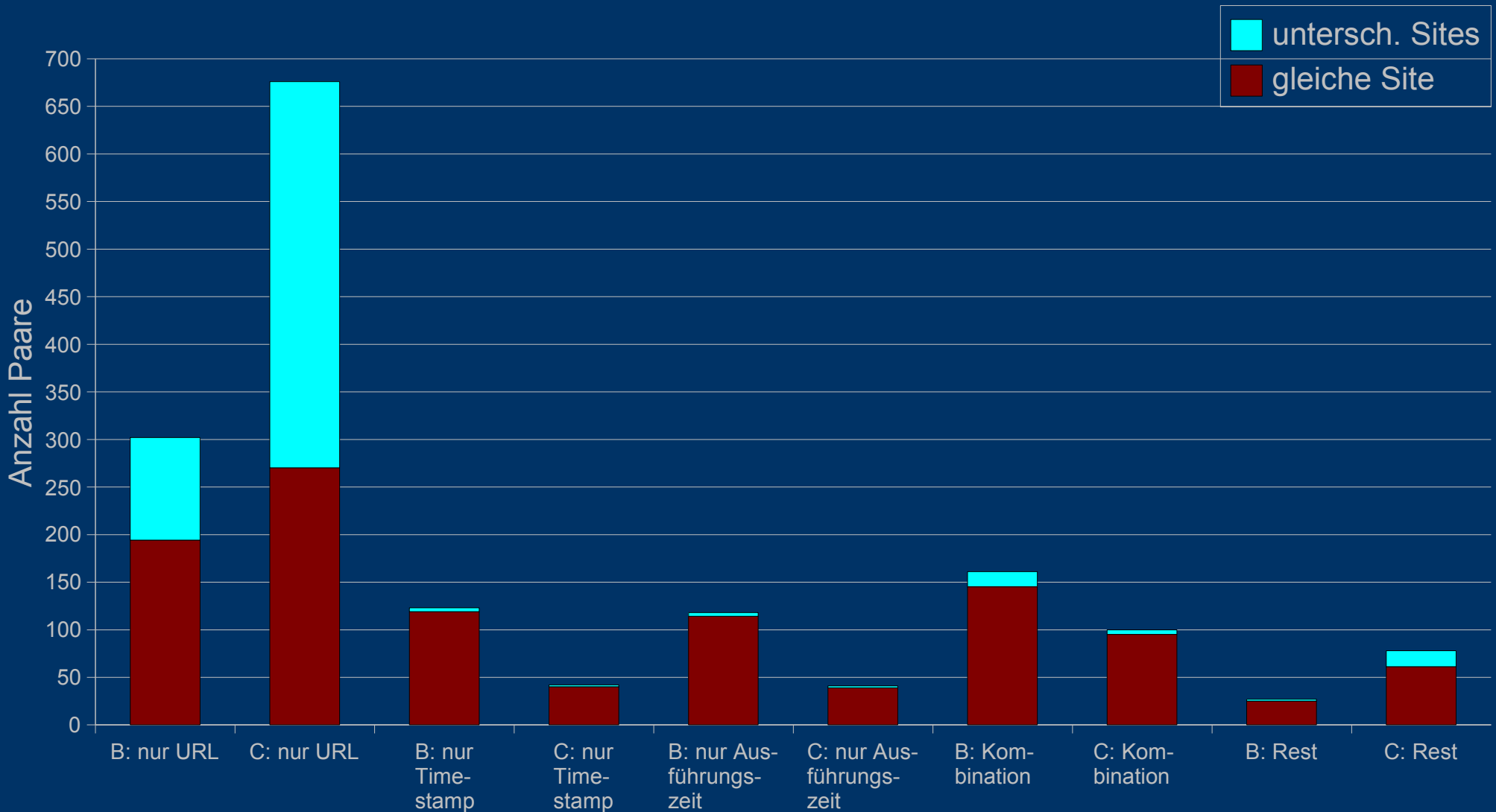
Experimente

Vergleich beider Algorithmen



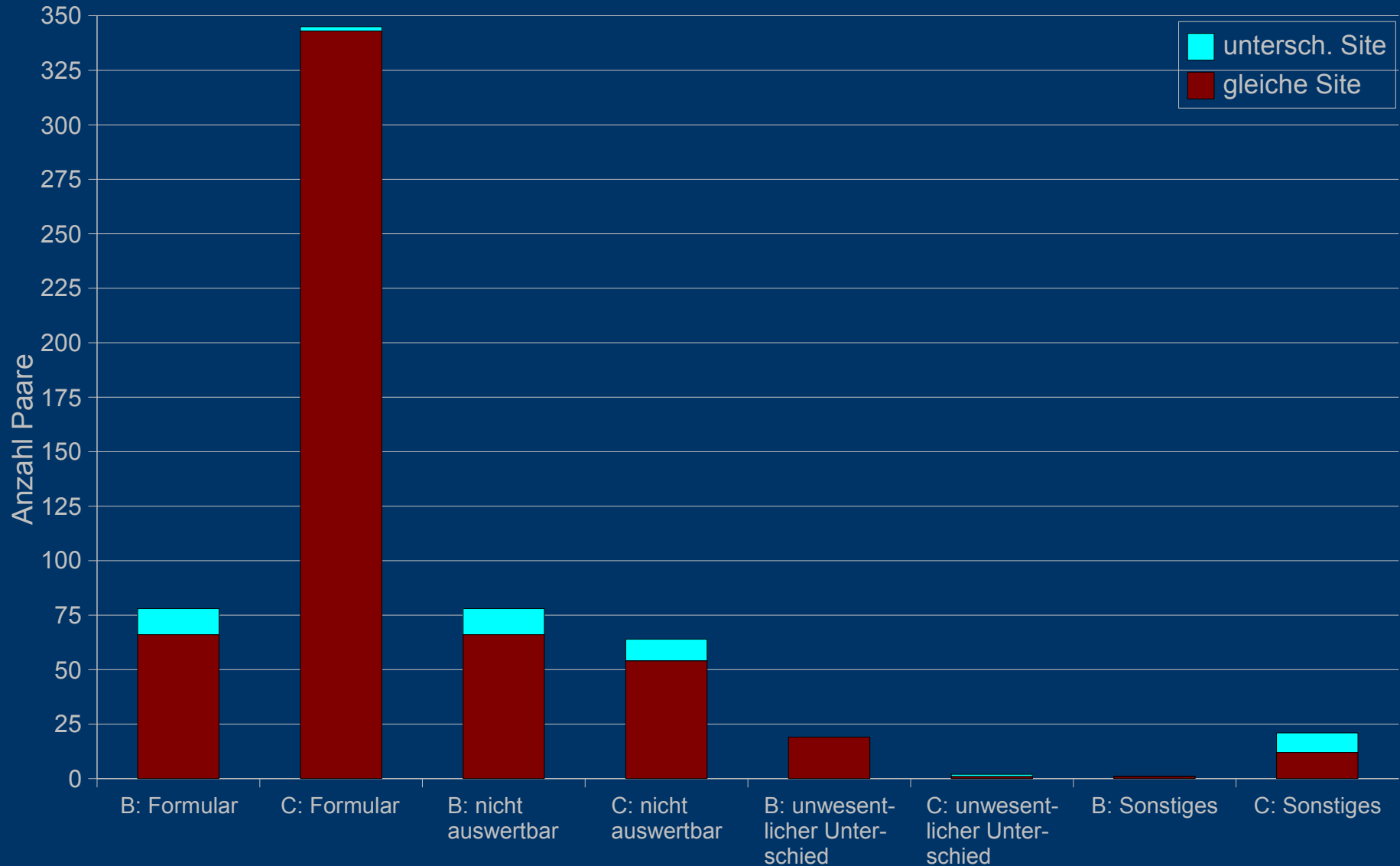
Experimente

Vergleich: Unterschiede bei korrekten Paaren



Experimente

Vergleich: Unentschiedene Paare



Experimente

Vergleich: Korrelation

- Eine Stichprobe von 96 556 B-ähnlichen Paaren wurde auf C-Ähnlichkeit geprüft. Nur etwa 4% hatten eine C-Ähnlichkeit von mindestens 372. Die C-Ähnlichkeit steigt proportional zum Wert der B-Ähnlichkeit.
- Umgekehrt wurden bei einer Stichprobe von 169,757 C-ähnlichen Paaren nur 4% B-ähnliche gefunden, 95% hatten sogar B-Ähnlichkeit 0.

Der kombinierte Algorithmus

- Besteht darin, dass erst Alg. B und auf dessen Ergebnissen Alg. C ausgeführt wird
 - Soll die falschen Positive von Alg. B beseitigen und die Präzision bei Paaren auf der gleichen Site erhöhen.
 - Filtert diejenigen B-ähnlichen Paare heraus, deren C-Ähnlichkeit unter einen bestimmten Grenzwert fällt.
-
-

Der kombinierte Algorithmus

Experimente

- Zunächst wurde experimentell der Schwellenwert 355 für den C-Teil des kombinierten Algorithmus ermittelt, so dass sowohl die Präzision maximiert wurde als auch ein möglichst hoher Anteil der korrekten Ergebnisse von B erhalten blieb
 - Präzision von 0,79, wobei auch 79% der korrekten Ergebnisse von B erhalten blieben
 - 82% der Paare gehörten zur gleichen Site
 - Verbesserung der Präzision für Paare der gleichen Site auf 0,74
-
-

Fazit und Ausblick

- Verbesserungsvorschläge für Algorithmus B:
 - Häufigkeit von Shingles beachten
 - Shingle-Größe verringern
- Vorschläge für Algorithmus C:
 - Algorithmus auf Shingles anwenden
- Vorschläge für den kombinierten Algorithmus:
 - für Paare auf der gleichen Site die Ergebnisse es Kombi-Algorithmus verwenden, für Paare auf unterschiedlichen Sites diejenigen von Alg. C