Goldstandards für Webkorpora

Felix Bildhauer und Roland Schäfer (alphabetisch)
SFB 632/A2 und Deutsche Grammatik (FU Berlin)

Workshop "Webkorpora in Computerlinguistik und Sprachforschung"
IDS Mannheim 27. – 28.09.2012
(Diese Version: 27. September 2012)

Freie Universität Berlin

Freie Universität Berlin

Goldstandards für Webkorpora???

Felix Bildhauer und Roland Schäfer (alphabetisch)
SFB 632/A2 und Deutsche Grammatik (FU Berlin)

Workshop "Webkorpora in Computerlinguistik und Sprachforschung"
IDS Mannheim 27. – 28.09.2012
(Diese Version: 27. September 2012)

Goldstandards für Webkorpora???

Freie Universität Berlin

COW-Projekt:
`http://hpsg.fu-berlin.de/cow/`

`CODS` Downloadsystem (COW im Leipziger Format):
`http://hpsg.fu-berlin.de/cow/download/`

COLiBrI Web-Interface:
`http://hpsg.fu-berlin.de/cow/colibri/`

`texrex-mrvain` (alt) und `texrex-hyperhyper` (in Arbeit):
`http://sourceforge.net/projects/texrex/`

COW Machine-Updated Notification and Information System for Twitter:
`http://twitter.com/cowmunist`

Goldstandards für Webkorpora???

Freie Universität Berlin

Mitarbeit

- **Felix Bildhauer** (SFB 632, FU Berlin)
  - Korpusevaluation
  - linguistische Nachverarbeitung
  - Spanisch, Französisch, Deutsch
- **Roland Schäfer** (Deutsche Grammatik, FU Berlin)
  - `texrex`-Programmierung
  - Crawlertechnik und -entwicklung (`CLARA`/`HeidiX`)
  - Schwedisch, Englisch, Deutsch, (Malay)
- **Adrien Barbaresi** (ENS Lyon/FU Berlin)
  - Crawlerentwicklung, Seed URL-Harvesting
  - automatische Dokumentenklassifikation
  - Französisch, Deutsch
- **Sarah Dietzfelbinger** (Deutsche Grammatik, FU Berlin)
  - Korpusevaluation, Generierung von Trainingsdaten
  - Türkisch, Deutsch

Überblick

Webkorpus Design: offene Fragen

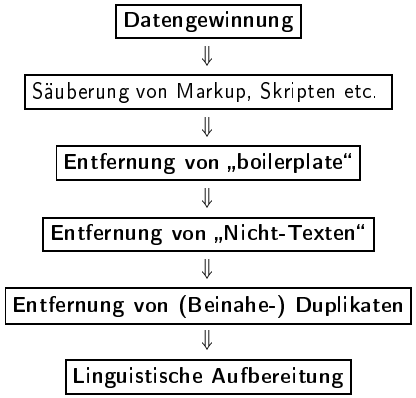
Optimiertes Sampling für Webkorpora

Wir sind hier. . .

Webkorpus Design: offene Fragen
- Korpuszusammensetzung
- Boilerplate
- Dokumentfilterung
- Linguistische Aufbereitung

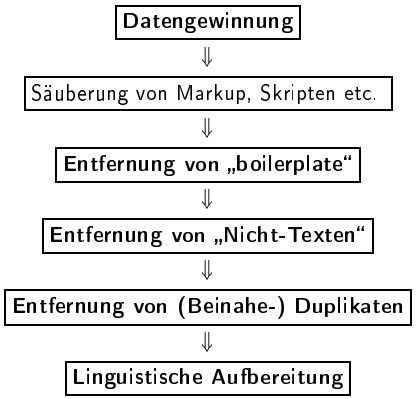
Optimiertes Sampling für Webkorpora

Webkorpuskonstruktion: Workflow



In (fast) allen Schritten werden Entscheidungen fällig, die sich auf Eigenschaften des fertigen Korpus auswirken.

Workflow II



- Wünschenswerte Verteilung von Texsorten im Korpus?
- Genaue Definition von „boilerplate“?
- Ausfilterung „ungeeigneter“ Dokumente?
- Akzeptable Menge von Duplikaten im Korpus?
- Bsp. Tokenisierung: Nicht-standardkonforme Schreibungen,

Datengewinnung / Korpuszusammensetzung

- COW (angestrebt): **Zufallsstichprobe** aus der Population deutschsprachiger Dokumente im (non-deep) WWW
- vgl. Wacky-Projekt (Baroni et al., 2009):
  explizit **keine Zufallsstichprobe** aus dem WWW erwünscht, sondern eine Zusammensetzung, die einem „unbiased/general purpose“ Korpus entspricht
  - Zusammensetzung nicht a priori festgelegt, sondern für das fertige Korpus geschätzt
  - Überprüfung (verschiedene Methoden), ob das Korpus hinreichend „unbiased“ ist

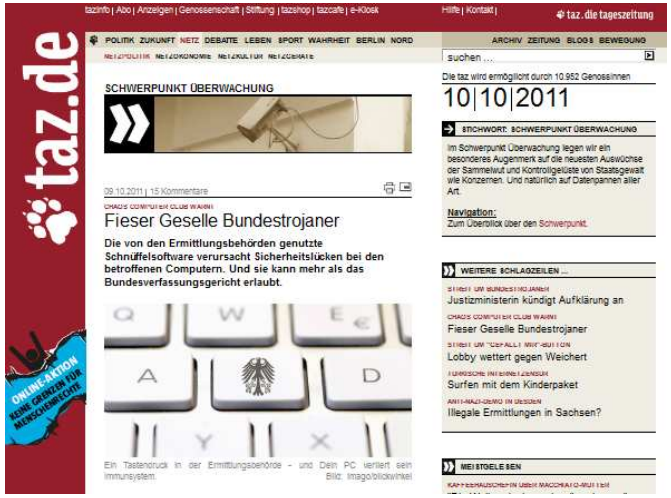
Datengewinnung / Korpuszusammensetzung II

- kein Einvernehmen über die Zusammensetzung eines balanciertes Korpus (z. B. „des Deutschen“; oder auch nur „des Deutschen im WWW“)
- COW (angestrebt): sehr großes „Masterkorpus“ aus Zufallsstichprobe
  - automatische (grobe) Klassifikation der Dokumente
  - dann stratifiziertes sampling aus dem Masterkorpus zur Erstellung großer Korpora mit vordefinierter Zusammensetzung, dem Forschungsgegenstand entsprechend

Entfernung von „boilerplate“

- verschiedene ML-Algorithmen anwendbar
- z. B. Spousta et al. (2008) basierend auf Marek et al. (2007): hohe precision/recall berichtet, aber ressourcenhungrig und relativ langsam
- COW: relativ gute precision/recall mit geringen Ressourcen (ANN)
- CleanEval (Baroni et al., 2008): „Wettbewerb“ zur Säuberung von Webseiten, inklusive Entfernung von boilerplate, anhand von manuell annotiertem Goldstandard
- Das größere Problem (aus unserer Sicht) ist ein konzeptuelles: Definition von boilerplate.

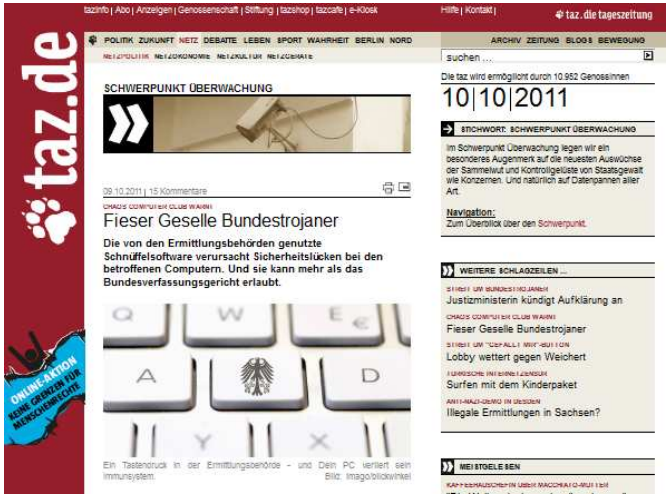
Beispiel



Beispiel (II)

Beispiel (III)

Boilerplate oder „Text“?



Boilerplate oder „Text“?

Boilerplate oder „Text“?

Boilerplate oder „Text“?

Boilerplate oder „Text“?

Boilerplate oder „Text“? (III)

Boilerplate oder „Text“? (IV)

Boilerplate oder „Text“? (V)

Boilerplate oder „Text“? (VI)

Boilerplate oder „Text“? (VII)

Boilerplate: workaround

Alternative zur willkürlichen Kategorisierung von Material als „boilerplate“:

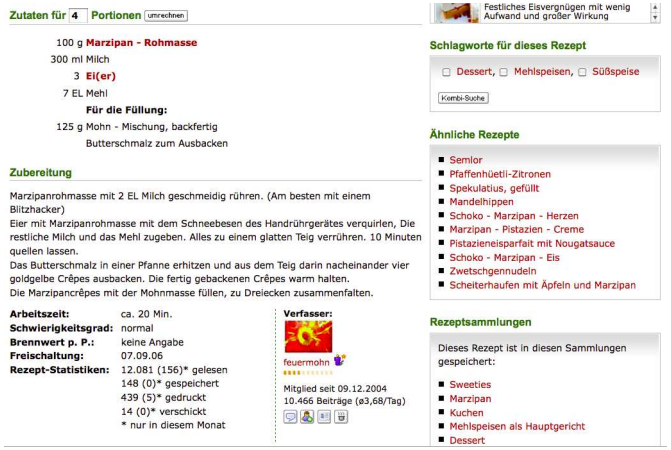
- Boilerplate nicht entfernen, sondern nur markieren
- **Vorteil**: Niemand muss unsere Definition von Boilerplate übernehmen; niemand muss sich auf die Qualität unseres ML-Algorithmus verlassen.
- **Nachteil**: Erhöhung der zu verwaltenden Datenmenge

Dokumentfilterung

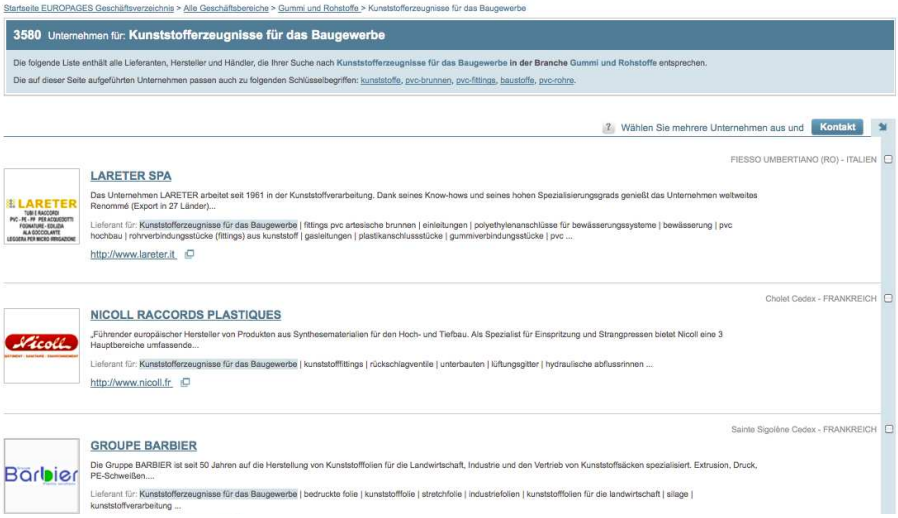
Das Korpus soll (möglicherweise) nur „gute“ Dokumente enthalten.

- „Gut“: Dokument enthält genug zusammenhängenden Text
- Schließt bestimmte Textsorten aus: z. B. Auflistungen, tagclouds etc.

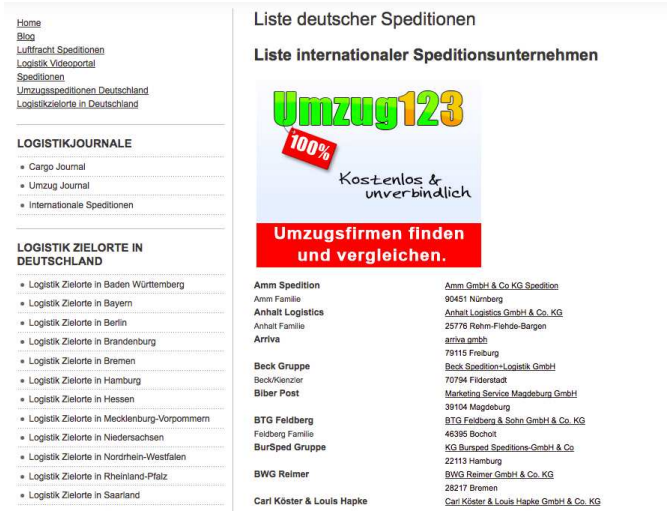
Beispiel „Textkriterium“: Listen



Beispiel „Textkriterium“: Listen II



Beispiel „Textkriterium“: Listen III



Dokumentfilterung in den COW-Korpora

- Mindestlänge: Anzahl der Zeichen *nach* Säuberung (markup & boilerplate)
- „Textkriterium“: Vergleiche rel. Frequenz der X häufigsten Worttypen mit einem vordefinierten Sprachprofil (z. Z. Unigrammmodell generiert aus einer Menge „guter“ Texte)
- Die häufigsten Worttypen sind Funktionswörter: charakteristisch für zusammenhängenden Text
- Sprachfilterung als Nebenprodukt (Grefenstette, 1995)

Duplikatentfernung

COW:

- w-shingling (Broder et al. 1997; ohne clustering)
- fingerprints bestehend aus 100 5-shingles
- def. „Duplikat“: zwei Dokumente haben 5 oder mehr gleiche 5-shingles im fingerprint
- das kürzere Dokument wird entfernt

Potentielles Problem: **Duplikation innerhalb desselben Dokuments**

- wird nicht erkannt (tritt aber massiv auf z. B. bei Zitaten in Diskussionsforen etc.)
- Umgang mit dieser Art von Duplikation unklar (Entfernung von Teilen eines Dokuments kann das Korpus für bestimmte Fragestellungen unbrauchbar machen)

Tokenisierung, Lemmatisierung, POS-tagging

Für die COW-Korpora: Treetagger (Schmid, 1995), out-of-the-box.

- zahlreiche Tokenisierungsfehler (verschiedene Ursachen)
- zahlreiche Fehler bei POS-tagging und Lemmatisierung (Lexikon nicht adäquat für Webdokumente 2011)
- Ergebnis: sehr hohe Zahl von Typen

N Tokens:	9,108,097,177	
N Typen:	63,569,767	
N *hapax legomena*:	39,988,127	
N Typen, instantiiert in > 6 Korpustranchen:	2,808,253	
N Tokens, die einen dieser Typen instantiieren:	8,777,938,146	(=96.4%)

Tokenization (II)

Beispiele aus DECOW2012:

Token	Original Context
!"Hallo	Laykan ?!!"**Hallo** hörst du mich
:rolleyes	**:rolleyes**: ja also für mich wars auch nix.
!"/	"Wie schnell rennst Du dem Geld hinterher?!"/"Join us! Fuck the Entwicklung einer eigenen Persönlichkeit"
u.bellte	Sie blieb an der Tür stehen **u.bellte** ihn recht zaghaft u.unsicher an
übernehmen."Die	alle 48 Arbeitnehmer der Gesellschaft zu **übernehmen."Die** Klägerin
'Achso	Ich: **'Achso**, das waren Sie nicht.
]Und	[...]**Und** die IG denkt darüber nach
-(Unterschiedliche	Beweggründe und Umstände -**(Unterschiedliche**) Rezeption des Werks
-)Irgendwie	;-)**Irgendwie** schon
->schnell	kann's nur weiterempfehlen -**>schnell** und langanhaltende Glätte
Diskussion	die Ich find diese Comics super!**Diskussion**
$-Marke	dass Silber nochmal an der 50 **$-Marke** abprallt
!":D:D	"Wie, du auch??!!"**:D:D**

Tokenisierung (III)

Token	Original Context
dbellte	louis erst mal voll erschrocken un **dbellte** sie an
!"#$%&'()*+	Der ASCII umfasst folgende Zeichen: ?!"#**$%&'()*+**,-./0123456789:;<=>?@ABC
$RapperDenIchNichtKenne	"Den einzigen Rapper, den ich abkann, ist **$RapperDenIchNichtKenne**. Aber der hat nen kleinen Pimmel!"
%SystemRoot%\System32	Die Datei befindet sich im Ordner "**%SystemRoot%\System32**"
üüüüüübbbbeerrrraaaaaaallllll	meine blöde Katze. die pisst mir überall hin aus Trotz, **üüüüüübbbbeerrrraaaaaaallllll**, die tickt einfach nicht sauber!

Wir sind hier. . .

⋆ Web Size Estimation und Abdeckung in Korpora I

Hinrich Schütze's Conclusion on web size estimation (Unterrichtsfolien 2008)

- Many different approaches to web size estimation.
- None is perfect.
- The problem has gotten much harder.
- There hasn't been a good study for a couple of years.
- Great topic for a thesis!

http://www2.kbs.uni-hannover.de/fileadmin/institut/pdf/ti1/slides/19web2.pdf

Web Size Estimation und Abdeckung in Korpora II

- Problem: Nur Schätzmethoden, oft via Suchmaschinen (Broder et al., 2006; Bar-Yossef and Gurevich, 2006).
- Irgendwo in einer Größenordnung von deutlich jenseits 10^{10} Seiten...
- Nur als Schätzung der Untergrenze z. B.: `http://www.worldwidewebsize.com/`
- **Auch ein Korpus von 10^6 Dokumenten ist also eine relativ kleine Stichprobe.**
- **Wichtige Frage: Welche Dokumente nehmen wir in die Stichprobe? Und wie finden wir diese am besten?**

Breadth-First-Strategien (FIFO-Queue)

- "The crawls are performed using the Heritrix crawler, with a multi-threaded breadth-first crawling strategy; they are stopped after 10 days of continuous running." (Baroni et al., 2009, 214)
- "[...T]he crawl, using a breadth-first strategy, got stuck trying to exhaustively harvest the few hosts strongly represented in the seeds." (Schäfer and Bildhauer, 2012, 487)
- Grund: Die Beschaffung großer Datenmengen ist mit BFS vergleichsweise einfach möglich.
- Alternativen für Suchmaschinen (SE): Prioritisierung nach **In-Degree** (Cho et al., 1998), **PageRank** (Cho and Schonfeld, 2007) oder anderem **SE-bezogenen Relevanzmaß** (Padney and Olston, 2008).

⋆ Einfache Charakterisierung von BFS

- BFS ist die einfachste URL-Sortierungs-Stategie, bei der URLs gemäß einer FIFO-Queue heruntergeladen werden.
  - Stelle eine Menge von Seed-URLs (=Start-URLs) in die Queue.
  - Beginne mit dem Herunterladen der URLs.
  - Stelle alle auf den heruntergeladenen Seiten gefundenen URLs in die Queue.
  - ...weiter so, bis Festplatte voll.

Implementierungsprobleme für Neusortierung

- Bei COW2012-Korpora gegen Ende der Crawls **200,000,000 URLs** bekannt, gesamte Queue-Größe oft $>$ **100,000,000 URLs**.
- Reordering auf diesen großen Queues ist nicht in-memory möglich und damit erheblich zeitaufwändig.
- Existierende Lösungen in Industriestärke sind prinzipiell für SE-bezogene Abdeckung optimiert, nicht für linguistische Korpora.
- Übersicht in Olston and Najork, 2010, Kap. 4.

Probleme mit Breadth-First I

- Frage: Ist BFS eine brauchbare Sampling-Strategie für linguistische Webkorpora?
- Zwei Aspekte:
  - Bekommen wir technisch bedingte unerwünschte Verzerrungen?
  - Finden und speichern wir möglichst schnell und exhaustiv die Dokumente, die wir aus linguistischer Sicht wollen?

Probleme mit Breadth-First II

- **Maximisieren wir mit BFS die *weighted coverage* *WC*** über unsere Crawl-Zeit t und die Menge der zu t gecrawlten Seiten $\mathcal{C}(t)$ (Olston and Najork, 2010, 29)?

$$WC(t) = \sum_{p \in \mathcal{C}(t)} w(p)$$

- Problem: $w_{Linguistik}$ ist ggf. völlig anders als $w_{Suchmaschine}$.
- Das Problem ist auf keinen Fall nur ein inhaltliches!
- **Schlechte WC-Maximierung führt zu erhöhtem Transfervolumen und Verarbeitungszeit.** Bei den jetzt angestrebten Korpusgrößen ist WC-Maximierung unausweichlich.

Probleme mit Breadth-First III

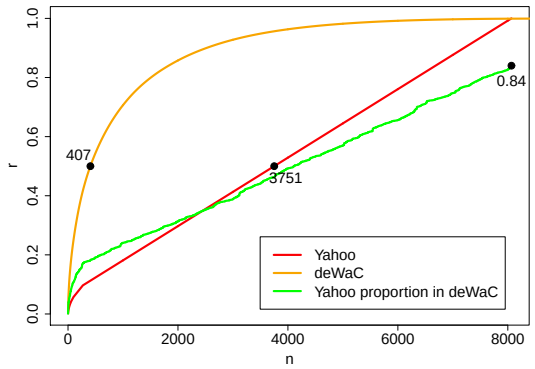
- BFS ist anfällig für **Crawler-Traps**.
- BFS führt als Samplingmethode zu **prinzipiell unbekannten Verzerrungen**, vgl. Kurant et al., 2011, sec. 6 und die dort zitierten Quellen.
- Das bedeutet: **Stichprobenverzerrungen aus BFS-Crawls sind nicht korrigierbar.**
- Vor allem bei kürzeren Crawls **beeinflusst die Auswahl der Seed-URLs die Komposition des Korpus**'.

Diversitätsmaße? Host-Verzerrung

- Die folgenden Plots sind **kumulative Plots** für verschiedene BFS-Webkorpora von:
  - dem Anteil r der Dokumente (vertikal) die von den n frequentesten Hosts (horizontal) des Korpus stammen – im veröffentlichten Korpus (orange)
  - ...in der Seed-URL-Menge (Yahoo oder Bing; rot)
  - dem Anteil der Dokumente im Korpus, die von den n häufigsten Hosts aus der Seed-Menge kommen (grün).
- Sehr krude als Verzerrungsmaß, aber jederzeit auch **ohne Kenntnis des Crawl-Verlaufs** berechenbar.
- Extreme Verzerrungen können Anzeichen von inhaltlichen/registerbezogenen/usw. Verzerrungen sein.

Host-Verzerrung für deWaC

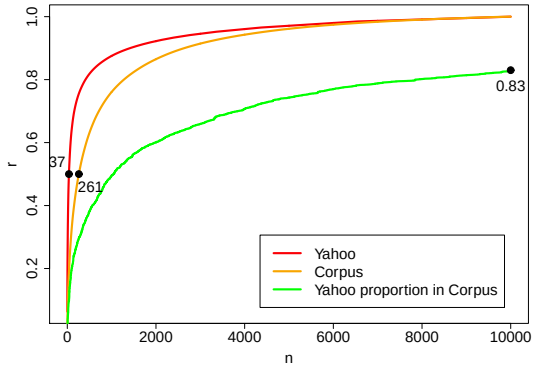
$N_{corpus} = 1,501,076$; $N_{seeds} = 8,631$; Crawldauer: 10d



Anm.: Die Seed-URLs enthielten max. 1 URL per Host.

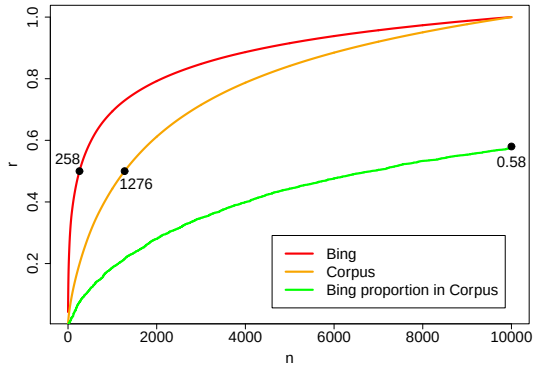
⋆ Host-Verzerrung für ESCOW2011

$N_{corpus} = 656,774$; $N_{seeds} = 229,698$; Crawldauer: 8d



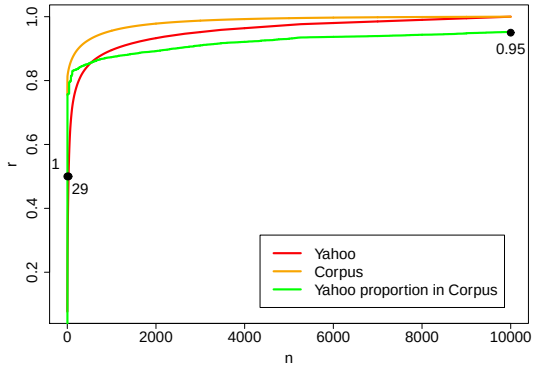
Host-Verzerrung für DECOW2012

$N_{corpus} = 7,759,892$; $N_{seeds} = 912,243$; Crawldauer: 28d



Host-Verzerrung für SECOW2011

$N_{corpus} = 1,912,757$; $N_{seeds} = 204,872$; Crawldauer: 7d



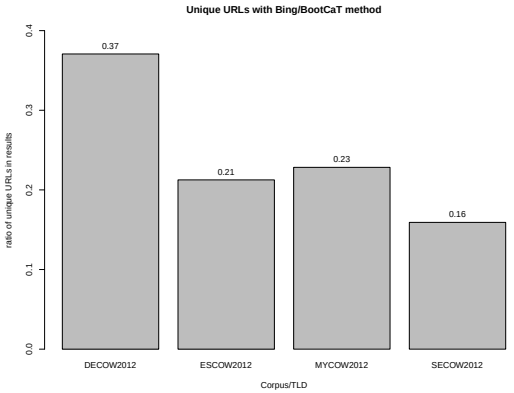
Zusammenhang zwischen Host-Verzerrung und Dokumenttyp-Verzerrung:
75.5% der Dokumente im Korpus von `http://www.blogg.se`

⋆ Problem schon im Vorfeld: Seed-URLs

- Seed-URL-Generierung aus Suchmaschinen (entspricht "BootCaT"-Methode, e. g., Baroni and Bernardini, 2004) ist kaum noch möglich, weil alle freien APIs geschlossen wurden.
- Suchmaschinen-Ergebnisse sind aber absichtlich verzerrt, z. B. weil Precision über Recall gestellt wird (Manning et al., 2009, 429ff), oder diverse und gesponsorte Relevanzmaße zum Einsatz kommen.
- Für viele TLDs liefert eine SE oft nur mittelmäßige Ergebnisse:
  - In einem **Bing**-Experiment in `.my` mit 100,000 4-Tupeln (10 URLs pro Tupel angefragt) aus einem malayischen Untertitel-Korpus im Februar 2012 näherte sich die **Zahl der bekannten Hosts stabil einem Wert knapp unter 4,000 an**.

⋆ Bing und einzigartige URLs in großen URL-Mengen

Anteil der einzigartigen URLs in Anfragen nach 1,000,000 URLs in verschiedenen TLDs:



Ziele für "Goldstandard" Sampling

- Vermeidung technisch durch Crawler bedingter Verzerrungen
- **Scoped Crawl** bzw. **WC-Maximierung** für linguistische Zwecke
- trotzdem große Datenmengen
- trotzdem Schnelligkeit/hoher Durchsatz
- also: technisch bedingte Deckelung der "Goldigkeit" der Samplingprozedur

Random Walks

- Random Walks sind tiefenorientiert statt breitenorientiert (explorativ statt exhaustiv).
- Einfache Implementierung:
  - Lade eine Start-Seite herunter,
  - extrahiere die Links,
  - folge einem Link zufällig,
  - . . . repeat . . .
  - (In OUT oder einem Tendril zu einer möglichst zufälligen Sprung-URL springen.)
- Über OUT und Tendrils vgl. Broder et al., 2000.

Entfernen ungewollter Verzerrungen

- Ein Random Walk kann unter Setzung einer generellen Sprungwahrscheinlichkeit (typischerweise 0.1..0.15)...
- ...**kalkulierbar nach dem PageRank (Brin and Page, 1998) der Seiten** verzerrt sein (Henzinger et al., 1999).
- Henzinger et al., 2000 schlagen eine einfache Korrektur der PR-Verzerrung durch post-hoc-Sampling aus dem Crawl-Ergebnis vor.
- Avanciertere Verfahren z. B. in Bar-Yossef et al. (2000); Rusmevichientong et al. (2001); Gjoka et al. (2011)

Gewollte Verzerrungen

- Warum laden wir für 8 Mio. Dokumente 200 Mio. herunter?
- Vieles können wir beim Crawl vermeiden oder gleich nach dem Crawlen aussortieren.
- **Individuell gewichtbare Metriken** für die Laufzeit-Errechnung von *w* in der *WC*-Maximierung und die **Vorhersage von *w* für entdeckte ungecrawlte Seiten** in einem **skopalen RW-Crawl**:
  - Sprache (statt TLD-Crawl),
  - Textkriterium,
  - Textlänge, Satzlänge, Menge Boilerplate usw.
  - auffällige Keywords,
  - Linktext (Vorhersage),
  - Per-Host-Sprache- und Textqualität (Vorhersage),
  - Geolokalisierung, ...

Zusammenfassung: Integration von RW-Crawler und Nachverarbeitung

- PageRank-Verzerrung effizient (teilweise) korrigieren
- linguistisch motivierte Verzerrung im Sinne einer WC-Maximierung beim Crawl berücksichtigen
- Möglichkeit der Vorhersage der Güte für ungesehene Seiten
- unmittelbare Entfernung schlechter Seiten
- Crawls von Einzelsprachen ohne Restriktion auf TLD
- technische Erleichterung der Beachtung von Politeness-Restriktionen

Language-classified Seed Directory: LSD

- Reservoir von Seed-URLs/Sprungadressen für RW-Crawl
- nur teilweise aus (Meta-)Suchmaschinenergebnissen
- ohne Redirects und tote Seiten
- Quellen: etools.ch (mit Tricks), dmoz.org, Wikipedia Dumps, identi.ca, Twitter, etc.
- Level-1- und Level-2-Expandierung der URLs (experimentell)
- **alle URLs werden mit hoher Qualität auf die Dokument-Sprache geprüft** (Lui and Baldwin, 2012).
- für sich genommen bei stark repräsentierten Sprachen groß genug für Korpuskonstruktion (auch statt BootCaT)

Crawler for Linguistic Applications with Random Access: CLARA

- experimenteller RW-Crawler (in ObjectPascal), **begonnen (10%)**
  - benutzt `texrex-hyperhyper` (70%) als Bibliothek
  - interne Konvertierung auf UTF-8
  - PageRank-Schätzung und Verzerrungskorrektur
  - nicht optimiert, single-threaded, reines Forschungstool
  - LSD als Datenbank für Sprungadressen
- Ziel: Experimente, um Auswahl der Metriken für skopalen Crawler zu optimieren.

HeidiX: Heidi is a Crawler System

CLARA als parallelisiertes Produktivsystem, finanzierungsabhängig.

Erreichtes und Erreichbares

- Ziel in erster Näherung erreicht: **Sehr große möglichst qualitativ hochwertige und linguistisch nachbereitete Korpora ausgewählter Sprachen.**
- Fazit für Qualitätssicherung: Die **linguistischen Entwurfsentscheidungen** überwiegen die technischen Probleme ML-Algorithmen usw.
- De Kette von Prozeduren und Werkzeugen zur Qualitätssicherung (inkl. Lemmatizer-Rewrites, Tagger-Retraining, Spellchecking, Dehyphenation, etc.) ist noch nicht vollständig für Korpora dieser Größe.
- Zusätzliche Forderung: neue Konzentration auf die **Sampling-Methode**. . . Sonst besteht die Gefahr von **Größe bei Unbrauchbarkeit**.

Referenzen I

Baeza-Yates, Ricardo, Castillo, Carlos, Marin, Mauricio and Rodriguez, Andrea. 2005. Crawling a country: better strategies than breadth-first for web page ordering. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, WWW '05, pages 864–872, New York, NY, USA: ACM.

Bar-Yossef, Ziv, Berg, Alexander C., Chien, Steve A., Fakcharoenphol, Jittat and Weitz, Dror. 2000. Approximating aggregate queries about web pages via random walks. In *Proceedings of the 26th International Conference on Very Large Data Bases*, pages 535–544.

Bar-Yossef, Ziv and Gurevich, Maxim. 2006. Random Sampling from a Search Engine's Index. In *Proceedings of WWW 2006*, pages 367–376, Edinburgh.

Baroni, Marco and Bernardini, Silvia. 2004. BootCaT: Bootstrapping Corpora and Terms from the Web. In *Proceedings of LREC 04*, pages 1313–1316.

Baroni, Marco, Bernardini, Silvia, Ferraresi, Adriano and Zanchetta, Eros. 2009. The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-Crawled Corpora. *Language Resources and Evaluation* 43(3), 209–226.

Baroni, Marco, Chantree, Francis, Kilgarriff, Adam and Sharoff, Serge. 2008. CleanEval: A Competition for Cleaning Webpages. In *Proceedings of LREC 06*, pages 638–643, ELRA, Marrakech.

Brin, Sergey and Page, Lawrence. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proceedings of the 7th International World Wide Web Conference*, pages 107–117, Elsevier Science.

Broder, Andrei, Fontura, Marcus, Josifovski, Vanja, Kumar, Ravi, Motwani, Rajeev, Nabar, Shubha, Panigrahy, Rina, Tomkins, Andrew and Xu, Ying. 2006. Estimating corpus size via queries. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, pages 594–603, New York, NY, USA: ACM.

Broder, Andrei, Kumar, Ravi, Maghoul, Farzin, Raghavan, Prabhakar, Stata, Raymie, Tomkins, Andrew and Wiener, Janet L. 2000. Graph structure in the Web. In *In Proceedings of the 9th International World Wide Web conference on Computer Networks: The International Journal of Computer and Telecommunications Networking*, pages 309–320, North-Holland Publishing Co.

Bryan, Kurt and Leise, Tanya. 2006. The $25,000,000,000 Eigenvector: The Linear Algebra behind Google. *SIAM Review* 48(3), 569–581.

Cho, Junghoo, García-Molina, Hector and Page, Lawrence. 1998. Efficient Crawling through URL ordering. In *Proceedings of the 7th International World Wide Web Conference*.

Cho, Junghoo and Schonfeld, Uri. 2007. RankMass crawler: A Crawler with High Personalized PageRank Coverage Guarantee. In *Proceedings of the 33rd International Conference on Very Large Data Bases*.

Gjoka, Minas, Kurant, Maciej, Butts, Carter T. and Markopoulou, Athina. 2011. A Walk in Facebook: a Case Study of Unbiased Sampling of Facebook. In *Proceedings of IEEE INFOCOM 2010*, IEEE, San Diego.

Grefenstette, Gregory. 1995. Comparing two language identification schemes. In *Proceedings of the 3rd Internation conference on Statistical Analysis of Textual Data (JADT 1995)*, pages 263–268, Rome.

Henzinger, Monika, Heydon, Allan, Mitzenmacher, Michael and Najork, Marc. 1999. Measuring Search Engine Quality using Random Walks on the Web. In *Proceedings of the 8th International World Wide Web Conference*, pages 213–225.

Henzinger, Monika R., Heydon, Allan, Mitzenmacher, Michael and Najork, Marc. 2000. On near-uniform URL sampling. In *In Proceedings of the 9th International World Wide Web conference on Computer Networks: The International Journal of Computer and Telecommunications Networking*, pages 295–308, North-Holland Publishing Co.

Kurant, Maciej, Gjoka, Minas, Butts, Carter T. and Markopoulou, Athina. 2011. Walking on a graph with a magnifying glass: stratified sampling via weighted random walks. In *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, SIGMETRICS '11, pages 281–292, New York, NY, USA: ACM.

Lui, Marco and Baldwin, Timothy. 2012. langid.py: An Off-the-shelf Language Identification Tool. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*.

Manning, Christopher D., Raghavan, Prabhakar and Schütze, Hinrich. 2009. *An Introduction to Information Retrieval*. Cambridge: CUP.

Marek, Michal, Pecina, Pavel and Spousta, Miroslav. 2007. Web Page Cleaning with Conditional Random Fields. pages 155–162, Louvain: Presses universitaires de Louvain.

Olston, Christopher and Najork, Marc. 2010. *Web Crawling*, volume 4 of *Foundations and Trends in Information Retrieval*. Hanover, MA: now Publishers.

Padney, Sandeep and Olston, Chris. 2008. Crawl Ordering by Search Impact. In *Proceedings of the 1st International Conference on Web Search and Data Mining*.

Rusmevichientong, Paat, Pennock, David M., Lawrence, Steve and Giles, C. Lee. 2001. Methods for sampling pages uniformly from the world wide web. In *In AAAI Fall Symposium on Using Uncertainty Within Computation*, pages 121–128.

Schmid, Helmut. 1995. Improvements in Part-of-Speech Tagging with an Application to German. In *Proceedings of the EACL SIGDAT-Workshop*, Dublin, Ireland, ftp://ftp.ims.uni-stuttgart.de/pub/corpora/tree-tagger2.pdf.

Schäfer, Roland and Bildhauer, Felix. 2012. Building Large Corpora from the Web Using a New Efficient Tool Chain. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk and Stelios Piperidis (eds.), *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 486–493, Istanbul: ELRA.

Spousta, Miroslav, Marek, Michal and Pecina, Pavel. 2008. Victor: The Web-Page Cleaning Tool. In Stefan. Evert, Adam Kilgarriff and Serge Sharoff (eds.), *Proceedings of the 4th Web as Corpus Workshop*, pages 12–17, Marrakech.

Wir sind hier. . .

Anhang

PageRank I

- Introduced by Brin and Page (1998), and it was one of the the key initial ideas behind Google.
- The original paper is reasonably accessible, but try also Manning et al. (2009) or other papers with summaries/discussion of PageRank like Baeza-Yates et al. (2005).
- The original PageRank (and many refinements) is the attempt to provide a better measure of a page's **relevance** than its mere in-degree:
  - A PageRank of a page is **high if it is linked to by many pages with a high PageRank**.
  - Pages with **few outlinks contribute more to the PageRanks of the pages they link to**.

PageRank II

- The PageRank R of a page p is the equilibrium probability that a Markovian random walker through the web graph is on page p.
- From each page p the walker follows an out-link from p with a fixed probability $1 - d$, or jumps to a random page with probability d.
- Reasonable values for d are usually assumed to be between 0.10 and 0.15.
- The random jumps avoid accumulation of PageRank by pages (in OUT) without re-distribution of that PageRank, and they take the walker out of sinks.

PageRank III

- The definition of PageRank R of page p with fixed jumping probability d, where N is the total number of web pages, and $p_{1..k}$ are the pages that link to p, and $C(p_n)$ is the number of out-links on p_n:

$$R(p) = \frac{d}{N} + (1 - d) \sum_{i=1}^{k} \frac{R(p_i)}{C(p_i)}$$

- Actual calculations of PageRanks involve intermediate-level linear algebra, cf. Manning et al. (2009) or Bryan and Leise (2006).

Correcting for biases in random walks I

Henzinger et al. (2000) made an early suggestion to correct for the PageRank bias in a random walk and obtain better near-uniform samples (= samples where each page p has equal probability of being sampled). The idea is to **take a corrected sample from the results of a random walk**.

- They perform a random walk (RW) of a portion of the web.
- During the RW, pages with a high page rank had a high probability of being crawled, so their probablity of being sampled from the crawled pages must be lowered.
- So first, the PageRank of a crawled page $R(p)$ is estimated via its **visit ratio** $VR(p)$ during the RW:

$$R(p) \approx VR(p) = \frac{\text{number of times } p \text{ was visited}}{\text{length of the walk}}$$

Correcting for biases in random walks II

- From the (PageRank-biased) crawl, a sample is taken using a **correction** which is **inversely proportional** to the PageRank of a page. Mathematically:

$$Pr(p \text{ is sampled}) = Pr(p \text{ is crawled}) \cdot Pr(p \text{ is sampled} | p \text{ is crawled})$$

- $Pr(p \text{ is sampled})$ should be the same for all pages.
- They estimate the following for the first term on the right-hand side, where L is the length of the crawl:

$$Pr(p \text{ is crawled}) \approx L \cdot R(p) \approx L \cdot VR(p)$$

- For $Pr(p \text{ is sampled} | p \text{ is crawled})$, they only state that:

$$Pr(p \text{ is sampled} | p \text{ is crawled}) \propto R(p)^{-1}$$

Correcting for biases in random walks III

- **They sample from the results of the random walk using a skewed probability distribution over all crawled p_n such that the probability of drawing p_n is inverse to its (estimated) PageRank.**
- In the end, we get:

$$Pr(p \text{ is sampled}) = L \cdot \frac{VR(p)}{VR(p)}$$

A simple example

Visits of $p_1..p_6$ in a crawl with $L = 10$:

Page	Visits	VR	VR^{-1}	Pr (sampled from crawl)
p_1	4	0.4	2.5	0.053
p_2	2	0.2	5	0.106
$p_3..p_6$	1	0.1	10	0.213