Avoiding Useless Content While Crawling the WEB

O. Behrendt[1], A. Hierle, Infotiger UG, Munich, Germany

OSSYM 2021 – 3rd International Open Search Symposium, CERN, Oct 11-13

[1]ossym21@infotiger.com

Introduction
Web search engine overview
Useless content
Related work
Adaptive domain limiting (ADL)
ADL algorithm
Distribution of domain sizes
Confidence function
Future work
Conclusion
Why running a crawler?
Lessons learned
References

Outline

Introduction

- Web search engine overview
- Useless content
- Related work

Adaptive domain limiting (ADL)

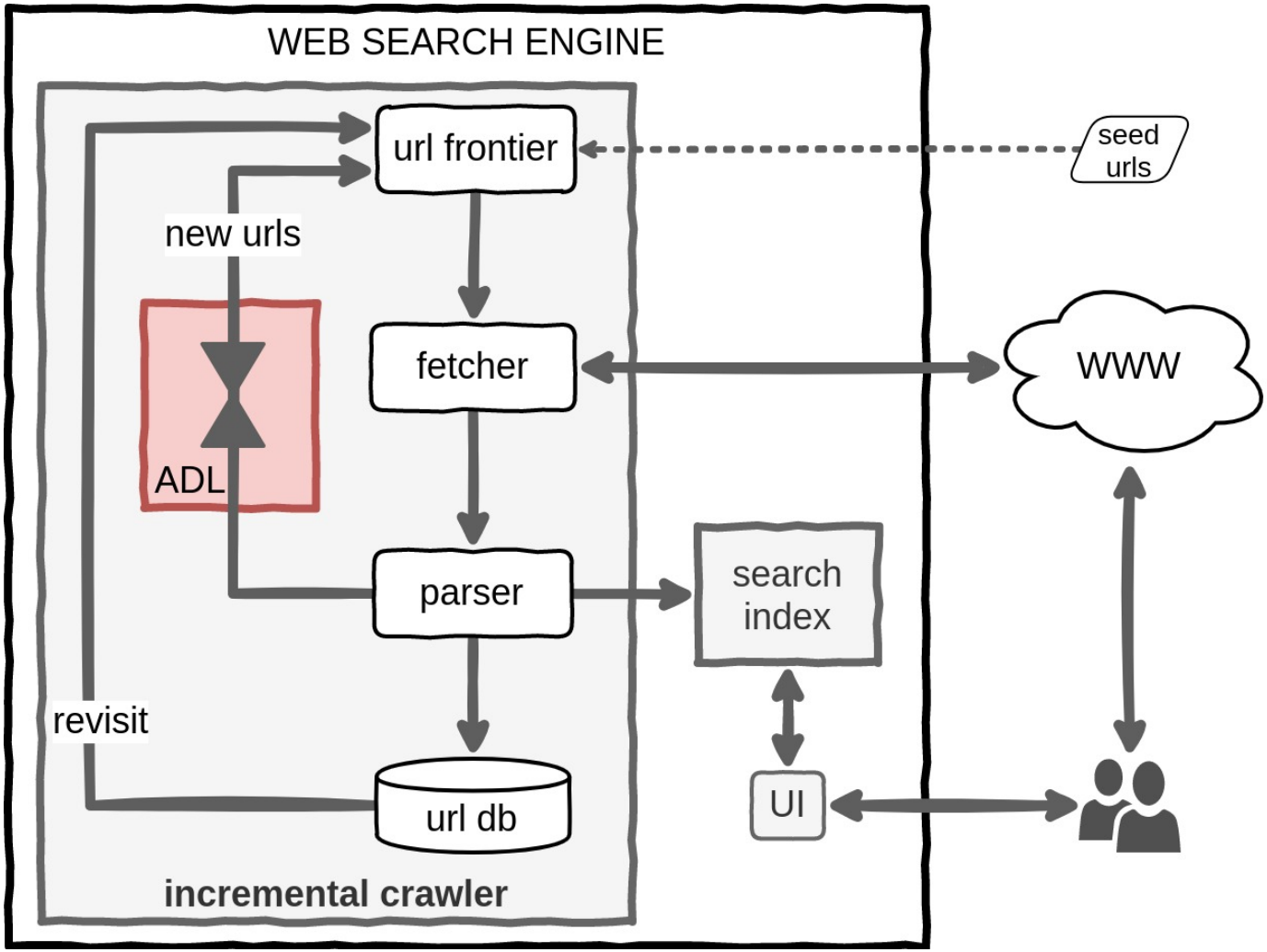
- ADL algorithm
- Distribution of domain sizes follows Zipf's law
- A confidence function based on link analysis
- Future work

Conclusion

- Why running an own crawler?
- Lessons learned

Introduction
Web search engine overview
Useless content
Related work
Adaptive domain limiting (ADL)
ADL algorithm
Distribution of domain sizes
Confidence function
Future work
Conclusion
Why running a crawler?
Lessons learned
References

Web search engine overview



Introduction
Web search engine overview
Useless content
Related work
Adaptive domain limiting (ADL)
ADL algorithm
Distribution of domain sizes
Confidence function
Future work
Conclusion
Why running a crawler?
Lessons learned
References

Useless content

Definition

- Link spam (link farms)
- Content spam
- Spider traps
- Unwanted content like casino, gambling, porn, ...

Useless content has negative impact

- Relevance of search results (recall and precision)
- Wasted resources (could be used for relevant content)
- Flooding of frontier queues, ...

Adaptive domain limiting (ADL)

ADL helps us to reduce useless content and thereby increase efficient use of limited resources and improve search quality.

Introduction
Web search engine overview
Useless content
Related work
Adaptive domain limiting (ADL)
ADL algorithm
Distribution of domain sizes
Confidence function
Future work
Conclusion
Why running a crawler?
Lessons learned
References

Related Work

- *IRLbot: Scaling to 6 Billion Pages and Beyond* [Lee+09] control the *allowed download rate* for a domain by its *reputation*, defined by the indegree from other PLDs[2]

Rank	Domain	In-degree	PageRank	Pages
1	microsoft.com	2,948,085	9	37,755
2	google.com	2,224,297	10	18,878
3	yahoo.com	1,998,266	9	70,143
4	adobe.com	1,287,798	10	13,160
5	blogspot.com	1,195,991	9	347,613
7	wikipedia.org	1,032,881	8	76,322
6	w3.org	933,720	10	9,817
8	geocities.com	932,987	8	26,673
9	msn.com	804,494	8	10,802
10	amazon.com	745,763	9	13,157

Table 4: Top ranked PLDs, their PLD in-degree, Google PageRank, and total pages crawled.

- Important approaches to identify spam pages are *TrustRank* [GGP04] and *truncated PageRank* [Bec+06] (but many more)

[2] pay-level domain $\approx$ 2nd-level domain

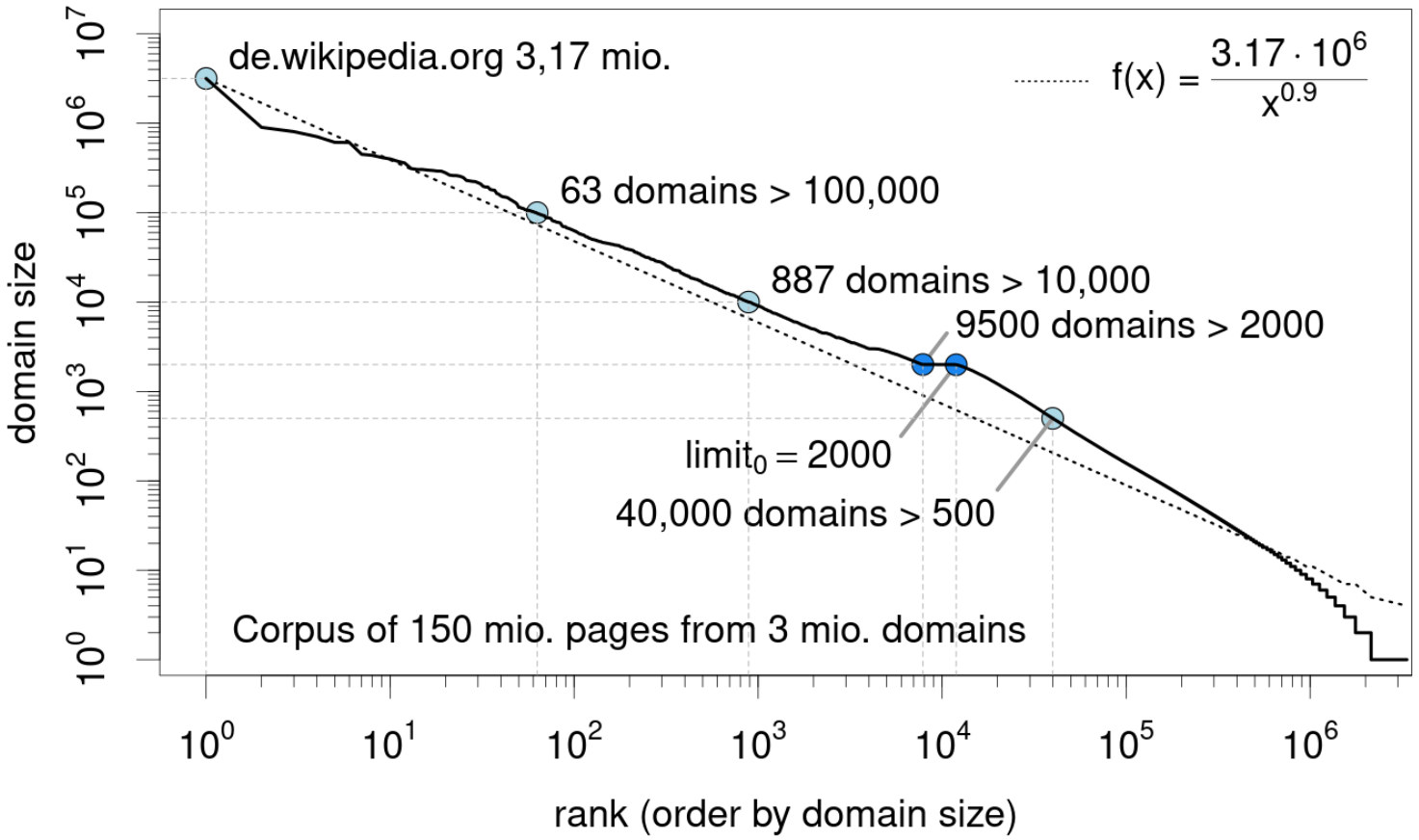
Adaptive domain limiting algorithm

Algorithm 1 ADL algorithm outline

1: $\text{limit}_t(d) :=$ maximum allowed domain size $|d|$ at time t
2: $\text{limit}_0 :=$ *const.* // *default limit e.g.* 2000
3: $\text{conf}(d) : D \rightarrow \{1, 0\}$ // *confidence in usefullness*
4: // *check new urls and ignore low-confidence domains*
5: **for all** *url* $\in$ {extracted links} **do**
6: **if** $|d| < \text{limit}_t(d)$ for $d = \text{domain}(url)$ **then**
7: add *url* to frontier // *else url is ignored*
8: **end if**
9: **end for**
10: // *periodically increase limits for high-confidence domains*
11: **for all** $d \in \{d \,|\, \text{conf}(d) = 1\}$ **do**
12: increase $\text{limit}_t(d)$
13: **end for**

Introduction
Web search engine overview
Useless content
Related work
Adaptive domain limiting (ADL)
ADL algorithm
Distribution of domain sizes
Confidence function
Future work
Conclusion
Why running a crawler?
Lessons learned
References

Distribution of domain sizes follows Zipf's law



Most domains are small

ADL impacts only a small percentage of web sites exceeding the default maximum domain size limit_0.

Introduction
- Web search engine overview
- Useless content
- Related work

Adaptive domain limiting (ADL)
- ADL algorithm
- Distribution of domain sizes
- Confidence function
- Future work

Conclusion
- Why running a crawler?
- Lessons learned

References

A confidence function based on link analysis I

Let aggPR the aggregated PageRank and aggTR be the aggregated TrustRank, both converted to quantiles. Then we can define following confidence rule:

$$\mathrm{conf}(d) := \begin{cases} 0 & \text{if } \mathrm{aggPR}(\mathrm{d}) > 99.9\% \wedge \mathrm{aggTR}(d) < 1\% \\ 1 & \text{otherwise} \end{cases}$$

domain	**PR qtl**	**TR qtl**
leqatus.com	0.999029	0.0
isabellaknoll.com	0.999376	0.0
gospellivemusic.com	0.999414	0.0
www.web-milan-hotels.com	0.999739	0.0
resisdancelondon.com	0.999361	0.0

Table: Randomly selected domains with $\mathrm{conf}(d) = 0$, $|d| = 2000$.

Introduction
Web search engine overview
Useless content
Related work
Adaptive domain limiting (ADL)
ADL algorithm
Distribution of domain sizes
Confidence function
Future work
Conclusion
Why running a crawler?
Lessons learned
References

A confidence function based on link analysis II

Five of 441 domains directly linking to leqatus.com

domain	PR qtl	TR qtl
professorsmoke.com	0.998517	0.0
yogastoponthe101.com	0.998910	0.0
bullet-group.com	0.998693	0.0
jugendzeltplatz.com	0.998222	0.345790
kidsportsinc.com	0.998047	0.0

Many domains with zero confidence are link farms

The above definition for a confidence function based on the difference of aggregated PageRank and TrustRank is a good indicator for link spam [GGP04].

Introduction
Web search engine overview
Useless content
Related work
Adaptive domain limiting (ADL)
ADL algorithm
Distribution of domain sizes
Confidence function
Future work
Conclusion
Why running a crawler?
Lessons learned
References

Future Work

Things we would like to do

- Evaluate performance of ADL by running two identically seeded batch crawls one with active, the other without. Compare ratio of

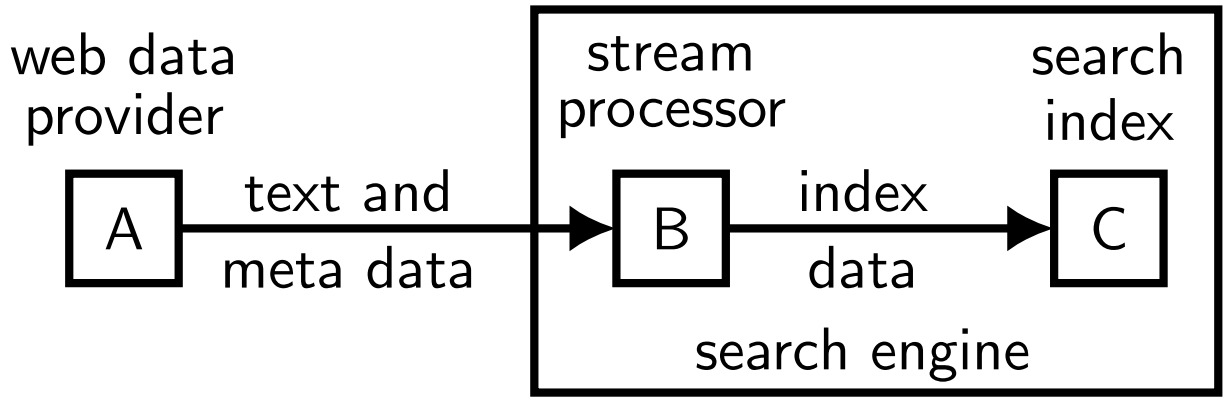
$$\frac{\text{useless pages (domains)}}{\text{total pages (domains)}}$$

- Use different or extended definitions of *confidence funcions* in order to increase effectiveness
- Implement algorithms to detect and remove highly interconnected link farms

Introduction
Web search engine overview
Useless content
Related work
Adaptive domain limiting (ADL)
ADL algorithm
Distribution of domain sizes
Confidence function
Future work
Conclusion
Why running a crawler?
Lessons learned
References

Why running an own crawler?

Scenario to use an open web data provider



Common requirements of web search engines

- Revisiting of frequently changing pages → freshness
- Spam detection and removal
- Boilerplate removal from HTML page
- "Deep" HTML parsing (JSON-LD, microdata, RFDa)
- Near-duplicate detection
- Reliability of web data provider (dependency)

Introduction
Web search engine overview
Useless content
Related work
Adaptive domain limiting (ADL)
ADL algorithm
Distribution of domain sizes
Confidence function
Future work
Conclusion
Why running a crawler?
Lessons learned
References

Lessons learned

Things we learned

- There are many useless pages which we do not want to crawl and index
- Adaptive domain limiting helps us to avoid useless content
- Crawler design significantly influences search results

Introduction
Web search engine overview
Useless content
Related work
Adaptive domain limiting (ADL)
ADL algorithm
Distribution of domain sizes
Confidence function
Future work
Conclusion
Why running a crawler?
Lessons learned
References

Thank you for listening!

Olaf

ob (at) infotiger.com

Alex

ah (at) infotiger.com

Introduction
Web search engine overview
Useless content
Related work
Adaptive domain limiting (ADL)
ADL algorithm
Distribution of domain sizes
Confidence function
Future work
Conclusion
Why running a crawler?
Lessons learned
References

For Further Reading

Hsin-Tsang Lee et al. "IRLbot: scaling to 6 billion pages and beyond." In: *ACM Transactions on the Web (TWEB)* 3.3 (2009), pp. 1–34.

Zoltan Gyongyi, Hector Garcia-Molina, and Jan Pedersen. "Combating web spam with trustrank." In: *Proceedings of the 30th international conference on very large data bases (VLDB)*. 2004.

Luca Becchetti et al. "Using rank propagation and probabilistic counting for link-based spam detection." In: *Proc. of WebKDD*. Vol. 6. 2006.

Introduction

Web search engine overview

Useless content

Related work

Adaptive domain limiting (ADL)

ADL algorithm

Distribution of domain sizes

Confidence function

Future work

Conclusion

Why running a crawler?

Lessons learned

References