Overall: looks good, just needs a bit more motivation in the first section, and a short conclusion. Also say where tool is (or will be) available.

LWAC: ~~CoolName2k~~: Longitudinal Web Sampling ~~for Fun and Profit~~ as Corpus

Stephen Wattam **Paul Rayson** **Damon Berridge**

Need departments here

Lancaster University

f.lastname@lancs.ac.uk

1 Sampling

Many sampling efforts for linguistic data on the web are heavily focused on producing results comparable to conventional corpora. These typically take two forms: those based on URI lists (~~typically~~ e.g. derived from search engine results, as in (Sharoff, 2006)), and those formed through crawling (Need example ref). Though initial efforts in WaC first focused on the ~~former~~ first type, in part due to concerns over the balance of samples returned, many projects are now focused on constructing supercorpora, which may themselves be searched with greater precision than the 'raw' web, in line with Kilgarriff's vision of linguistic search engines (Kilgarriff, 2003). This has led to the proliferation of crawlers such as those mentioned in (Schäfer and Bildhauer,) and (Renouf, 2003).

This approach, with its base in a continually-growing supercorpus, parallels the strategy of a monitor corpus (Sinclair, 1982), and is applicable to a linguistic inquiry concerned with diachronic properties (Kehoe, 2006). Indeed, we could conclude that the web is mature enough to require date-based lookup when retrieving articles, and such tools are increasingly being included in consumer search engines such as Google.

This repeated sampling approach tells us about the state of language change online in a manner that is immediately comparable to other diachronic corpora, however, it omits subtler technical aspects that govern consumption of data online, most notably the URI of the data, and variation as to where this points over time. Low publishing costs online, paired with increasing corporate oversight and reputation management, leads to a situation where this content is being updated frequently, often without end users even noticing.

This URI-oriented change has been studied from a technical perspective by those interested in managing and maintaining network infrastructure, and optimising the maintenance of search engine databases (Koehler and others, 2004). The needs of these parties are quite aside from those of corpus researchers, however, since they focus around a best-effort database of information, rather than a dependable sample (which must have known margins for error).

We present here a tool for this form of longitudinal sampling, designed to maximise the comparability of documents downloaded in each sample in terms of their URI rather than content. To accomplish this, we use a batch-mode sampling strategy, as illustrated in Figure 1, to get full coverage over a list of URIs, at the expense of sampling new content.

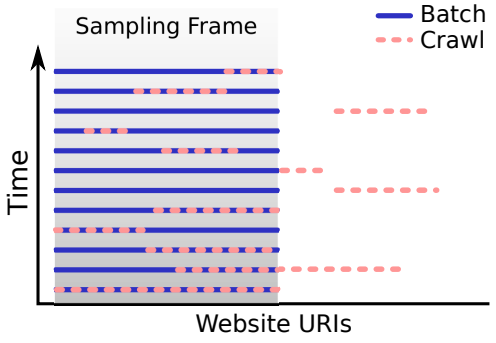

Figure 1: URI coverage for batch and crawl.

2 Aims

The tool is designed to construct longitudinal samples from URI lists, using only commodity hardware. It is designed with 'full storage' in mind, that is, recording everything about each HTTP session in such a way that it may later be exported and accessed in a parsimonious manner.

3 Architecture

In order to maximise the simultaneity of a given sample, a parallel, distributed architecture was selected (Figure 2). This also yields technical ben-

Which type does BootCat fit into? "web as corpus"

define this

Also refer to the TenTen Corpora. See AK's papers on his website. Are you also including UKWaC and others in this type?

Ref so, use them as egs.

Could also mention the library perspective here.

et al?

because

longitudinal

LWAC,

Say something about where the initial URI list comes from.

Describe motivation here from your document attrition research and reference the previous papers. This helps to motivate the regular re-downloading

explain what you mean here, and why it is important.

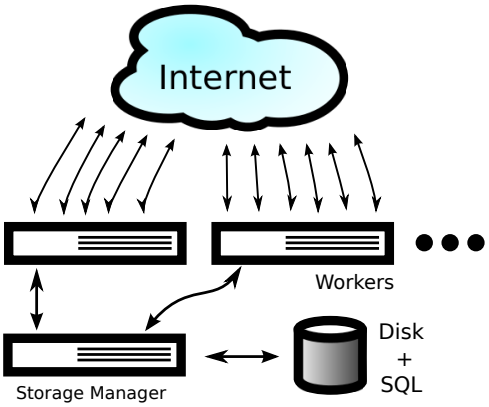

Figure 2: System Architecture

efits of throughput (especially where the internet connection is a bottleneck), and the ability to differentiate between websites that are blocked for a given area of the internet and those that are offline 'proper'.

Data storage in the system is split between metadata, stored in an SQLite database, and website sample data itself, which is stored as raw HTTP response data in a versioned structure, compressed using hard links[1]. The storage format is optimised for large samples, and is nested in order to avoid common filesystem limits.

The download process itself is managed by a central server, which co-ordinates storage and metadata access in order to provide atomicity and recording of sample data. This central server distributes batch jobs, according to policies governing reliability and throughput, to worker servers, which compete for the opportunity to download websites.

Workers imitate, as far as possible, the behaviour of real users. They retain cookies and present typical user-agent and referrer strings in their request headers.

4 Performance

In order to obtain the most simultaneous samples, the system was designed to maximise the parallel number of connections on each client. This eventually led to exceeding the limits of the underlying operating system, which in our tests showed a practical maximum of 120 simultaneous downloads[2].

In practice, throughput is defined both by the external servers and this parallelism limit—reducing timeouts for failed DNS and HTTP connections leads to significant improvements later on in a sample where many hosts have fallen offline. With low failure rates, the system is capable of downloading millions of pages in a 24-hour period.

As with many downloaders, it is possible to exceed polite limits of server usage with an internet connection of even modest throughput.

5 Applications

Corpora built using this strategy offer insights into the properties of language as it is used and maintained on a daily basis, yielding particular value to epistemic problems regarding web sampling:

- The proportions and areas of web pages that typically change as boilerplate and templating systems;
- The impact of social feedback and user generated content on page content;
- How censorship, redaction and revision affect website contents;
- Website resource persistence relative to content (link rot/document attrition).

References

Andrew Kehoe. 2006. Diachronic linguistic analysis on the web with WebCorp. *Language and Computers*, 55(1):297–307.

Adam Kilgarriff. 2003. Linguistic search engine. In *proceedings of Workshop on Shallow Processing of Large Corpora (SProLaC 2003)*, pages 53–58.

Wallace Koehler et al. 2004. A longitudinal study of web pages continued: a consideration of document persistence. *Information Research*, 9(2):9–2.

Antoinette Renouf. 2003. WebCorp: providing a renewable data source for corpus linguists. *Language and Computers*, 48(1):39–58.

Roland Schäfer and Felix Bildhauer. Building large corpora from the web using a new efficient tool chain. In *Proceedings of LREC*, volume 8.

Serge Sharoff. 2006. Creating general-purpose corpora using automated search engine queries. *WaCky!*, pages 63–98.

John Sinclair. 1982. Reflection on computer corpora in English language research. *Computer Corpora in English Language Research*, pages 1–6.

[1] Linked in a similar manner to `rsync`'s `-H` option.

[2] Using the Linux 2.8 kernel