Chris Biemann, Felix Bildhauer, Stefan Evert, Dirk Goldhahn, Uwe Quasthoff, Roland Schäfer, Johannes Simon, Leonard Swiezinski, Torsten Zesch

Scalable Construction of High-Quality Web Corpora

Abstract

In this article, we give an overview about the necessary steps to construct high-quality corpora from web texts. We first focus on web crawling and the pros and cons of the existing crawling strategies. Then, we describe how the crawled data can be linguistically pre-processed in a parallelized way that allows the processing of web-scale input data. As we are working with web data, controlling the quality of the resulting corpus is an important issue, which we address by showing how corpus statistics and a linguistic evaluation can be used to assess the quality of corpora. Finally, we show how the availability of extremely large, high-quality corpora opens up new directions for research in various fields of linguistics, computational linguistics, and natural language processing.

1 Introduction

The availability of large corpora is a prerequisite for both empirical linguistic research in many fields such as phraseology, graphemics, morphology, syntax, etc. and for many applications in language technology such as spell checking, language models, statistical machine translation, collocation extraction, or measuring semantic textual similarity. The web is certainly an abundant source of text documents and many projects have are already constructed corpora from web texts. Most of them either provide a final corpus in the form of a collection of sentences, paragraphs, or documents, e. g. COW [85], Gigaword [76], LCC [52], UMBC WebBase [57], and WaCky [11]. Some projects only provide aggregated information like word-level (Google Web 1T 5-Grams [24]) or syntactic n-gram counts [50]. As the documents found on the web can be quite noisy from a linguistic point of view, all the resulting corpora have been criticized (among other reasons) for being biased or for not properly reflecting the research question at hand. In this article, we argue that we can address these challenges by carefully controlling all steps of the corpus creation process, including (i) crawling, (ii) preprocessing, and (iii) an assessment of corpus quality.

It will become clear that there are diverse notions of "corpus quality" (and, consequently, "noise"), which depend on the intended use of the corpus. In empirically oriented theoretical linguistics, carefully selected sampling procedures and non-destructive cleaning is important, while for many tasks in computational linguistics and language technology, aggressive cleaning is fundamental to achieve good results. Technically, however, these differences merely result in different software configurations, but not

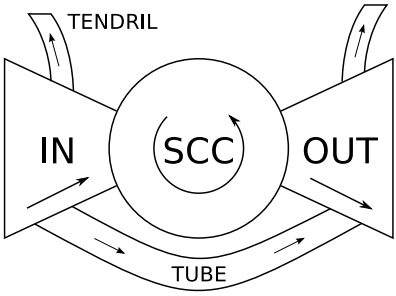

Figure 1: Schematic structure of the web according to [26], as depicted in [86, 9]

in substantially different software architectures. Also, all projects for web corpus construction considered in this paper put an emphasis on improving processing speed and collecting substantial amounts of data.

One major focus of this article is on web crawling, as it might be relatively easy to crawl a large English corpus, but in order to collect a big corpus for "smaller" languages more sophisticated crawling strategies are required. We further describe how the crawled data can be linguistically pre-processed, including ways of parallelization that allow the processing of web-scale corpora. When working with web data, controlling the quality of the resulting corpus is an important issue, which we address by showing how corpus statistics and a linguistic evaluation can be used to assess the quality of corpora. Finally, we show how the availability of extremely large, high-quality corpora opens up new directions for research in various fields of linguistics, computational linguistics, and natural language processing.

2 Web Crawling

Web crawling is the process of fetching web documents by recursively following hyperlinks. Web documents link to unique addresses (URLs) of other documents, thus forming a directed and cyclic graph with the documents as nodes and the links as edges. Each node has an in-degree (the number of nodes linking to it) and an out-degree (number of nodes linked to by it). It is usually reported that the in-degrees in the web graph are distributed according to a power law [7, 72].

Macroscopically, the web graph has been described as having primarily a tripartite structure [26, 87], as shown in Figure 1: Pages with an in-degree of 0 form the IN component, pages with an out-degree of 0 form the OUT component, pages with both in- and out-degree larger than 0 form the strongly connected component (SCC). Within SCC, there is a link path from each page to each other page. Since web crawling relies on pages being either known beforehand or having at least one in-link, it reaches only pages in SCC and OUT and pages in IN which are known at the start.

Besides pages with an in-degree of 0, certain other pages also cannot be reached, such as pages on servers which only serve certain IP addresses, pages requiring a login, form input, etc. There are approaches to access this so-called deep web (e. g. [70]), but we are unaware that any such techniques were used in web corpus construction so far. Besides the problems of accessing the deep web, crawling is complicated by the fact that the web is not static. Pages appear and disappear at very high rates. The decay of web sites is known as *link rot* [10]. Even worse, many web sites are dynamic and generate or remix content for each request based on input parameters, the client's IP, and geolocation.

A major issue is the extremely large size of the web. Even large search engines index only fractions of the web (web pages they classify as relevant), but already in 2008 it was announced that Google had indexed a trillion pages.[1] Thus, building corpora from the web starts by sampling from a population and will usually result in a corpus that is orders of magnitude smaller than the web itself.

2.1 Sampling

The population from which the sample is taken is the set of web documents. There are several options for sampling: We can take random (preferably uniform) samples, cluster samples, stratified samples, or non-random samples like systematic/theoretical samples (an overview of sampling techniques can be found in Chapter 3 of [23]). In most variants, crawling is some form of random sampling, whereas classical balanced corpus construction is a variant of stratified and/or theoretical sampling, cf. Section 5.2. It is rather complicated to do the same stratified sampling with web data, because (i) the relative sizes of the strata in the population are not known, and (ii) it would be required to start with a crawled data set from which the corpus strata are sampled, as web documents are not archived and pre-classified like many traditional sources of text. Web documents have to be *discovered* through the crawling process, and cannot be taken from the shelves. If a crawling procedure favors the inclusion of, for example, highly popular web pages (e. g. pages which are linked to by many other pages), it might be argued that a theoretical or systematic sample is drawn. Still, the basics of the initial discovery procedure would remain the same.

In this section, we have discussed the nature of the web to the extent that it is relevant for crawling. A more detailed overview is given in [86]. Additional information can also be found in [72, 75], although these sources are more concerned with implementation details than with empirically sound sampling and its relation to crawling strategies, which we cover in the next section.

2.2 Crawling Strategies

Crawling strategies can be divided into graph traversal techniques (each node is only visited once) and random walks (nodes might be revisited), cf. [49]. Available off-the-

[1] `http://googleblog.blogspot.de/2008/07/we-knew-web-was-big.html`

shelf crawler software like Heritrix[2] [74] or Nutch[3] commonly applies some kind of graph traversal, mostly a (modified) Breadth-First Search (BFS). In a BFS, the crawler starts with a set of known URLs (the *seeds*), downloads them, extracts the contained URLs and adds them to the queue, then crawls again etc. This strategy leads to the exhaustive collection of a local connected sub-component of the web graph. BFS can thus introduce a bias toward the local neighborhood of the seed URLs, possibly not discovering relevant/interesting material in other parts of the graph. For example, a URL/host analysis for WaCky [11] and COW corpora showed that these corpora contain many documents from only a few hosts, most likely due to BFS crawling [85]. Although there was only one URL from each host in the crawling seed set for deWaC, these hosts account for 84% of the documents in the resulting corpus. In a crawl of the top level domain `.se`, we found that 95% of the corpus documents came from the hosts already represented in the seed set, and one single host alone (`blogg.se`) accounted for over 70% of the documents [85]. Furthermore, in a series of theoretical and experimental papers [3, 65, 64, 71], it was found that BFS is biased toward pages with high in-degrees, and that this bias is impossible to correct for. Whether such biased (and hence not uniform) random sampling is acceptable is an important design decision. It is especially problematic from the perspective of empirical linguistics, where the question of sampling should receive close attention (cf. Sections 2.1 and 5.2). However, an important advantage of BFS is that it allows crawling a huge amount of web pages in a short time.

Alternative strategies, which deliver considerably less yield in the same time compared to BFS, are random walks (RW). A random walk is characterized by recursively selecting a single out-link from the current page and following this link. Pages might be revisited in the process (something which implementations of BFS crawlers go to great lengths to avoid). Sampling based on random walks also delivers samples which are biased, but in contrast to BFS, these biases can be corrected for, for example by rejection sampling [58, 81]. Such approaches have not yet been used in web corpus construction, although, for example, [31] mention unbiased sampling as possible future work. The COW project (see Section 3.2) is currently working on their own crawler (CLARA) which implements diverse bias-free crawling algorithms based on random walks. A long-term goal is linguistic characterization for single languages based on uniform random samples, i. e. the characterization of the distribution of linguistic features in the web graph.

Refinements There is a number of refinements that can be applied to modify the basic crawling strategy. We briefly discuss three such refinements: scoped crawling, focused crawling, and optimized crawling.

Scoped crawling imposes constraints on the kind of crawled documents. The idea is to avoid downloading content which would not be included in the final corpus anyway. A scoped crawl is restricted by accepting only documents from certain URLs, IP ranges, etc. Restricting the scope of a crawl to a national top level domain is

[2] `http://webarchive.jira.com/wiki/display/Heritrix/`
[3] `http://nutch.apache.org/`

standard procedure in monolingual web corpus construction. Since documents are never exclusively written in one language under any top level domain, additional language detection/filtering is required. Also, with the recent proliferation of top level domains, it is no longer guaranteed that the relevant documents in one language can be found under its associated top level domain. A non-scoped but focused crawler (as discussed below) might be the better solution in the long run.

Focused crawling imposes even stricter constraints and tries to efficiently discover specific types of information (languages, genres, topics, etc.) which cannot simply be inferred from address ranges, domains, or hosts. A focused crawler guesses for each harvested link the kind of document it points to. Normally, the link URL and the surrounding text (in all documents where the link appears) are analyzed in order to predict the contents of the linked document. Various heuristics and machine learning methods are used for the prediction [4, 28, 29, 30, 54, 73, 82, 92]. For language detection based purely on URLs, see [15]. For topic and genre detection by URL analysis, see [2, 14].

What we call *optimized crawling* is similar to focused crawling, but it is merely intended to make the crawl more effective, so that more usable text documents can be downloaded in a shorter time, wasting less bandwidth. Contrary to focused crawling, there is not necessarily a specific restriction on the contents of the crawled documents. The crawl is biased towards documents which, according to some metric, have a high relevance. The bias can be implemented, e. g. by giving promising URLs a better chance of being queued compared to less promising URLs. This is an important aspect in crawling for search engine applications, where usually a bias toward pages with a high PageRank is desirable. In [1], Online Page Importance Computation (OPIC) is suggested, which is basically a method of guessing the relevance of the pages while the crawl is going on.

An interesting experiment concerning linguistically motivated crawl optimization was reported in [93].[4] The authors integrate their post-processing into their own crawler and collect statistics about the final yield from each encountered host after the removal of material which is not good corpus material. Hosts receive penalty for low yield, up to a point where they get effectively blacklisted. Although the optimization effect is reported to be moderate, the method is in principle suitable for collecting more corpus data in a shorter time.

2.3 Strategies Used by Existing Web Corpus Projects

We now give an overview of the crawling and post-processing strategies used by previous web corpus projects.

The WaCky project (*Web as Corpus kool ynitiative*) has released web corpora in a number of European languages [11]. A similar project is COW (*Corpora from the Web*) [85]. Both used the Heritrix crawler, which means that a variant of BFS was applied, restricted to national top level domains. However, the Heritrix strategy is not

[4] The crawler is available as SpiderLing: `http://nlp.fi.muni.cz/trac/spiderling/`

pure BFS, because a system of queues in a multi-threaded implementation is used to optimize the download process. This system prefers downloading all documents from one host exhaustively in one batch. Taking care of one host en bloc makes a number of tasks more efficient, e. g. keeping track of robots exclusion information for the host, obeying delay times between requests to the host, and caching DNS information. This leads to an overall BFS with a preference for host-internal breadth.

The UMBC WebBase corpus [57] consists of 3.3 billion tokens of "good quality English" extracted from the February 2007 crawl of the Stanford WebBase project. The aim was to compile a "large and balanced text corpus" for word co-occurrence statistics and distributional semantic models. Suitable text was selected based on various heuristics and de-duplicated at paragraph level.

The *Leipzig Corpora Collection* (LCC) uses mainly two approaches in parallel for collecting textual data in various languages [52]. On the one hand, the distributed web crawler FindLinks[5] is used, which implements a BFS strategy. There is no restriction to specific top level domains, and it therefore discovers a large number of domains while following the existing link structure of the web. The crawl is split into so-called rounds, where the URLs found in the documents of one round are crawled in the next round. To prevent the size of the rounds from exponential increase, the number of pages per domain is limited to between five and twenty, depending on the top-level domain. On the other hand, news pages are crawled exhaustively using httrack.[6] The directory of AbyZNewsLinks provides a very comprehensive list of URLs for news in about 120 languages which are used as seeds.[7]

The largest publicly available resource derived from a web corpus was compiled by researchers at Google Inc. as a basis for the *Google Web 1T 5-gram* database, or Web1T5 for short [24]. According to the authors, this corpus consists of about 1 trillion words of English text. However, it is not distributed in full-text form, but only as a database of frequency counts for n-grams of up to five words. Throughout, n-grams with fewer than 40 occurrences were omitted from the database. Due to these restrictions, this resource is of limited use for linguistic purposes, but it can be – and has been – applied to certain types of analyses such as collocation identification (cf. also Section 4.2).

So far, we have focused on projects which have created publicly available corpora. Another (not publicly available) project was described in [77], which resulted in a 70 billion token corpus for English based on the ClueWeb09 dataset.[8] According to an informal and otherwise unpublished description on the ClueWeb09 web page, it was crawled with Nutch and "best-first search, using the OPIC metric", which is biased toward documents which are relevant to search engine applications.[9]

Finally, there are completely different kinds of web corpus projects, which use stratified non-random sampling as mentioned in Section 2.1, such as the German DeRiK corpus

[5] `http://wortschatz.uni-leipzig.de/findlinks/`
[6] `http://www.httrack.com/`
[7] `http://www.abyznewslinks.com/`
[8] `http://lemurproject.org/clueweb09/`
[9] `http://boston.lti.cs.cmu.edu/Data/web08-bst/planning.html` on 2012-04-22.

project of computer-mediated communication [16]. Such corpora are designed with a specific purpose in mind, and they are orders of magnitude smaller than the large crawled web corpora we discuss in this paper.

3 Processing Crawled Data

In this section, we describe approaches to web data processing, i. e. the steps taken to transform crawled data into a corpus which can be used for linguistic research or applications in language technology. The purpose of corpus construction is to provide a sufficient amount of data for a given purpose. For some research questions, a small or a medium sized corpus contains enough data, but in many cases the objects of interest are very rare, and a large corpus is needed. Such questions include research on:

- phraseological units (especially those falling out of use)
- neologisms (might also require certain registers not available in traditional corpora)
- statistical data for very infrequent words (as used, for instance, in distributional semantics)
- word co-occurrences for infrequent words
- rare (morpho-)syntactic phenomena of high theoretical importance
- (structured) n-gram counts for language modeling

We will take a detailed look at two projects that perform processing of crawled data for uses in (computational) linguistics: WebCorpus and COW. Both projects share the following high-level architecture: Data from crawls is filtered to retain only textual material in a desired target language, and unified regarding the encoding. Depending on the unit of interest, which could be either a document or a sentence, a de-duplication step removes redundant material. The resulting large raw corpus is subsequently annotated with automatic linguistic processing (such as POS-tagging). Finally, statistics and indices are generated. The projects were designed towards slightly different goals, differ in their implementation, address scaling differently, and target overlapping but somewhat different usage scenarios for the resulting corpora. Regarding technological aspects, there are no fundamental differences between COW and WebCorpus, as they both implement the same high-level design pattern: a pipeline that performs various pre-processing and filtering steps, and results in a corpus.

3.1 WebCorpus Project

This section describes WebCorpus[10] that implements corpus data processing based on the Hadoop MapReduce framework.[11] After motivating this choice of parallelization technology, we flesh out the generic pipeline steps with concise descriptions of techniques and software used in their implementation. The overall framework is entirely written in Java, and is available under the open-source Apache Software License 2.0.[12]

[10] http://sourceforge.net/projects/webcorpus/
[11] http://hadoop.apache.org/
[12] http://code.google.com/p/dkpro-bigdata

As the size of web corpora usually ranges in the order of gigabytes to terabytes, processing requires considerable amounts of time. Sole optimization of code execution speed, e. g. through the use of advanced compilers, is insufficient as runtime will stay proportional to the input size. A better suited approach to reduce runtime is the (massive) use of parallelization through distributing independently processable parts of the data, which in our case can be done by, e. g. dividing data archives into single records or texts into sentences.

One method of distributing such "embarrassingly parallel" [48] problems, which has recently gained popularity, is MapReduce [33]. Its core idea is to perform processing in two consecutive steps: the map phase and the reduce phase. In the map phase, input is read and split in a defined manner appropriate for a specific *input format*, e. g. one that reads text documents and creates one split per line. These are then processed in parallel by multiple *mappers*, which can themselves produce an arbitrary number of intermediate results, e. g. extracted tokens. In a second step, *reducers* will combine these intermediate results, for example by counting all occurrences of identical tokens and producing a sorted list of tokens with frequencies.

A notable advantage of this method is that it scales arbitrarily with increasing amounts of input data: Additional mappers and reducers can run on different cores or on different machines. Since all intermediate data is processed independently in each of the two phases, runtime can be reduced almost linearly by adding more computing hardware. Reducing processing time is therefore merely a matter of available resources.

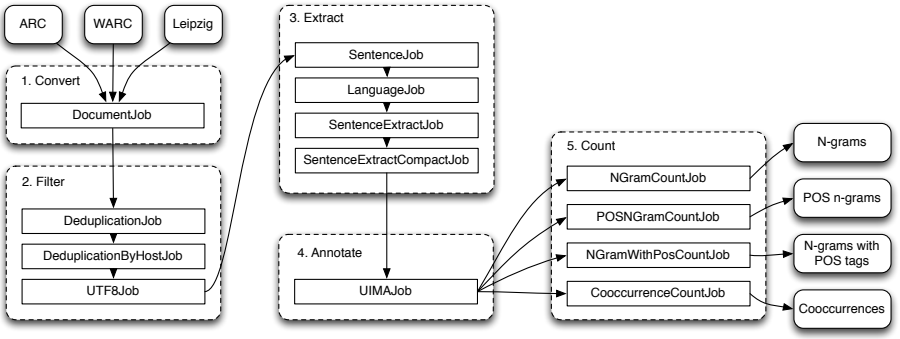

Figure 2: WebCorpus processes data in a pipeline fashion to allow for error recovery and reuse of intermediate data. Lines represent the data flow between jobs.

Figure 2 illustrates data flow in the WebCorpus project. The data is processed in a pipeline fashion, where results from each job are written to disk before being passed onto the next job, which in turn has to read the data back from disk. Especially for large-scale computations that involve long running jobs and many machines, the scheme of saving intermediary steps makes it possible to use partial results even if an outage

occurs. Also, this makes it possible to reuse output data from any job in other contexts, e. g. to use de-duplicated sentences and their counts to determine frequent boilerplate text. In a production environment, intermediate outputs can be deleted automatically once they are not needed by any further steps in the pipeline.

For the WebCorpus project, the Hadoop framework was chosen, which has become the de-facto standard implementation of MapReduce. Native Hadoop code is written in Java, as is most of the WebCorpus code, although other programming environments exist for Hadoop. One such example is the data-flow oriented scripting language *Pig Latin*.[13] Its capability of performing joins across several MapReduce tables allows the computation of significance measures on word co-occurrences.

Convert Since the input can be given in various formats, such as HTML, ARC[14], or WARC[15] the pipeline starts with a conversion step that extracts documents from the input and stores them in a unified format: One document per record is stored along with its metadata from the crawl, i. e. URL, time of download, etc. In the first pipeline step, those different input formats are read and split into parts that can be processed in parallel. For ARC/WARC, this is done by reading input archives using the open-source library JWAT[16] and generating a split for each archive record. Although web archives can contain text documents in virtually any format such as PDF, word processor, presentations, or even images, only HTML documents are used for processing. In order to extract content from HTML and remove boilerplate text, we use the *html2text* package of the Findlinks project, which itself makes use of the Jericho HTML parser.[17] Encoding of all HTML documents was normalized by utilizing the *encodingdetector* package of the Leipzig ASV toolbox [20].[18] In this step, removal of unwanted (i. e. non-textual) material, DOM-tree based cleaning [12] as well as normalization of white space are also performed. Also, parameters that are irrelevant to the document content can be removed, e. g. session IDs and tracking parameters added by Google WebAnalytics. Once plain text documents have been extracted from the input, we perform additional normalization of white spaces and mark paragraphs in the text with XML-like tags. Additionally, metadata such as URL or time of download are kept and also enclosed in XML tags in the resulting text documents. Normalization also extends to metadata: different URL strings may denote the same location, due to possible permutations of query parameters. For example, `?lang=de&page=2` and `?page=2&lang=de` are equivalent – a simple and effective way to resolve such cases is to sort query parameters alphabetically.

[13] `http://pig.apache.org/`
[14] `https://archive.org/web/researcher/ArcFileFormat.php`
[15] `http://archive-access.sourceforge.net/warc/`
[16] `https://sbforge.org/display/JWAT/`
[17] `http://jericho.htmlparser.net/docs/`
[18] `http://wortschatz.uni-leipzig.de/~cbiemann/software/toolbox/`

Filter In order to reduce the amount of data that has to be processed in later steps, undesirable documents are filtered as early as possible in the pipeline. Such documents may be duplicate documents, documents with a broken encoding, documents that are too long or too short, or documents not written in the target language. In WebCorpus, we tackle de-duplication using an approach that incorporates a Bloom filter [22] for probabilistic detection of duplicates residing on the same host (as defined by the URL), which produced satisfactory results while using very few resources. See Section 3.2 for a more sophisticated method for finding document duplicates.

Encoding detection can be performed with different types of heuristic approaches. Depending on the input format, encoding has either already been normalized (usually to UTF-8) or encoding information is contained in the metadata. We thus only have to catch cases of wrongly converted or non-convertible documents. For this, we do not need to incorporate a full-fledged system for encoding detection such as ICU.[19] It suffices to look for malformed UTF-8 bytes and malformed UTF-8 byte sequences.

While all filtering steps described so far operate on documents, language detection is performed on sentences. Language filtering is performed with the *LanI* language identification system from the ASV Toolbox. Based on word frequencies and the size of the underlying corpus, LanI returns the most probable languages of a given sentence. If LanI does not return the expected language as the most likely one, then the sentence is either filtered from the data, or it is considered for further processing if one of following conditions are met: the sentence is not at the beginning or at the end of a paragraph, and its length is shorter than a configurable value, e. g. 200 characters. In this way, content from mixed-language documents can be used for all target languages. After language detection has been performed, subsequent steps only operate on a predefined target language and ignore material of other languages.

Extract While all subsequent WebCorpus steps can also be run on documents, we describe a sentence-centric pipeline from this point on. It is advantageous for many applications – in particular those based on a statistical analysis of the corpus data – to remove duplicated sentences, i. e. to keep only one copy of every distinct sentence. This removes artifacts that are valid sentences but are vastly overrepresented because of their appearance in boilerplate text, e. g.: "You are not allowed to edit this post."[20] These artifacts lead to unwanted bias in the frequency distribution of contained terms, which hurt linguistic analysis.

In the extraction step, we apply a language-independent, extensible rule-based system for splitting all the documents that passed the filtering step into sentences. For specific languages, this could easily be replaced by a sentence splitter tuned to the target language. Filtering based on sentence length or other metrics can also be performed, cf. [56]. To extract the sentences in a basic format, two Hadoop jobs are employed.

[19] International Components for Unicode: `http://site.icu-project.org/`

[20] Boilerplate is text which is usually not written by humans, but inserted automatically from a template, etc. Typical boilerplate material includes navigational elements, copyright notices, "read more" snippets, date and user name strings.

The first job outputs all sentences along with their source URLs. The second job de-duplicates the sentences in its reducer, computes the sentence frequency and outputs the frequency as well as up to ten URLs per sentence.

Annotate In this step, we use the Unstructured Information Management Architecture (UIMA) [44] in order to allow for arbitrary annotations and to flexibly support future annotation requirements. Since UIMA has recently gained popularity, there is a variety of annotators available that can be reused easily, e. g. from OpenNLP[21] or the DKPro framework.[22] Currently, the OpenNLP tokenizer and POS taggers for German and English are included. The sentences are processed in parallel by Hadoop mappers that pass the sentences through the UIMA pipeline. Each component in the pipeline writes annotations to the Common Analysis Structure (CAS), an efficient in-memory database used by UIMA to store annotations. In the final step of the pipeline, all annotation stored in the CAS are written to disk as XML files. While the produced XML documents tend to be verbose, it was found to be sufficient to compress them with *gzip* to keep required storage space and disk I/O overhead in reasonable ranges.

Count Finally, based on the annotated corpus, statistical analyses are performed. First, patterns are generated from the corpus, e. g. all token unigrams, all verb-object dependencies, or any other pattern that can be defined on top of the annotations. Afterward, the generated patterns are collected, counted, and written in a suitable exchange format for further processing or indexing. For a scalable implementation of co-occurrence significance measures and second order similarities, the reader is referred to the JoBimText project [21].[23]

Figure 3 illustrates the process of annotating and pattern counting using the Hadoop framework: Sentences are tokenized and annotated with POS tags, which are stored in the UIMA CAS. Then, a map step extracts patterns of interest, in this case POS-tagged bigrams. In the reduce step, these are counted across the whole corpus.

With WebCorpus, it was possible to process 600 GB of German web-crawled text (already stripped from non-textual material, as provided by the Findlinks[24] project [59]) on a (comparatively small) Hadoop cluster with 8 nodes providing 64 cores in total. The preprocessing from the raw input to a total of 130 Million de-duplicates sentences (over 2 Gigatokens) took about 8 hours. The annotation with POS tags and the counting of 1-5-grams with and without POS tags required another 11 hours. Since most steps scale linearly with the amount of input data, the speed can be increased easily by adding more machines with sufficient disk space, without ever having to care about RAM limits.

[21] http://opennlp.apache.org/
[22] http://code.google.com/p/dkpro-core-asl/
[23] http://sourceforge.net/p/jobimtext/
[24] http://wortschatz.uni-leipzig.de/findlinks/

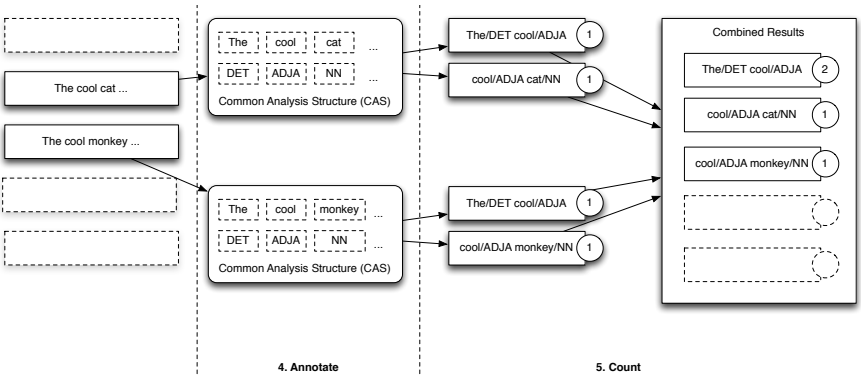

Figure 3: Annotation, pattern gathering and count example

3.2 COW Project

We describe here the most recent (COW2013) workflow, which was first tested on the UKCOW2012 corpus that is already available.[25] In the COW project, everything up to but not including tokenization is implemented in the `texrex` software suite.[26] The software is available under (modified) GNU (Lesser) General Public License(s) and is written in the object-oriented dialect of FreePascal.[27] The latter choice is motivated by the speed and exceptionally good platform support of the compiler. Even integrating the tools into Java-based environments is unproblematic, because the compiler can compile directly (i. e. without code translations) to JVM byte code.

First, there is a *conversion* step, which includes encoding normalization to UTF-8 using ICU, as well as HTML stripping. To allow for later analysis of link structures, link relations between the corpus documents are extracted in the HTML stripping process. Then, documents undergo certain quality assessment steps, most importantly boilerplate detection using a Multi-Layer Perceptron (MLP) trained on document-internal features [85] and text quality assessment [84].[28] Instead of simply removing potential boilerplate material, it is preserved and marked as potential boilerplate by storing the output of the MLP as paragraph metadata. Also, the overall text quality, which is measured as the lack of otherwise highly frequent words, only leads to the removal of the document if it is extremely low. Otherwise, the text quality metric is added as document metadata, and the document is preserved. This allows corpus users to perform queries restricted

[25] `http://hpsg.fu-berlin.de/cow/?action=corpora`
[26] `http://sourceforge.net/projects/texrex/`
[27] `http://www.freepascal.org/`
[28] Notice that recent versions of the COW software uses 39 features as input for the boilerplate classifier, not just the 9 features described in the original paper. A comprehensive list of features for this task is described in [91].

to potentially non-noisy regions of the corpus, while still being able to see the noisier regions (cf. Section 5.2 for why this is desirable).

Removal of near-duplicate documents is performed by a conservative implementation of w-Shingling without clustering as described in [25] and using an efficient implementation of Rabin hashes [78]. All important parameters (w, fingerprint size, and sensitivity) are configurable. Duplicated documents are removed, but for future versions of the software, a method of recording the degree to which each document, which was not removed, had duplicates in the original crawl data is being developed.

In the *annotation* step, standard tools for POS tagging, chunking, and named entity recognition are used, but the tools vary from language to language to ensure best possible results. For example, named entity recognition for German turned out to work best using the Stanford NER tool with the deWaC-generalized classifier described in [41], whereas for Swedish, the best option is the Stockholm Tagger, which also performs NER [83].[29,30]

In addition to finding the optimal annotation tools and models, the noisy data in web documents usually requires hand-crafted pre- and post-processing scripts (or even modified rules and models for the software used) in each annotation step. For example, special language-specific rule sets for the Ucto tokenizer were developed to handle documents which contain emoticons, non-standard use of punctuation, etc.[31] An introduction to the problems of linguistic processing of noisy data can be found in Chapter 4 of [86].

Finally, because whole documents cannot be redistributed under German copyright legislation, shuffled versions are produced for public release in the *shuffle* step. Sentence-wise shuffled corpora contain single sentences in random order (unique, with the frequency of the sentence in the original corpus as metadata). However, for tasks like distributional semantics, there will be in-document shuffles (sentences within the documents are shuffled) and windowed shuffles (words are shuffled within windows of n words). For such applications, it is a commendable option to use only those documents classified as being of high quality and the paragraphs which are most likely not boilerplate.

4 Assessing Corpus Quality

Large-scale web corpora as discussed in this article are often designed to replace and extend a traditional general-language reference corpus such as the British National Corpus (BNC) [5]. In contrast to more specialized samples used to study text types unique to computer-mediated communication (e. g. [16]), web corpora are expected to be similar to "traditional" corpora compiled from newspapers, magazines, books, essays, letters, speeches, etc. Their advantages lie in (i) better accessibility (e. g. there is no German reference corpus whose full text can freely be accessed by researchers),

[29] http://www.nlpado.de/~sebastian/software/ner_german.shtml
[30] www.ling.su.se/english/nlp/tools/stagger/
[31] http://ilk.uvt.nl/ucto/

(ii) much larger size, allowing for a more sophisticated and reliable statistical analysis (web corpora are now typically 10 to 100 times larger than the BNC), (iii) inclusion of up-to-date material so that recent trends and developments can be tracked (most of the texts included in the BNC were produced between 1985 and 1993), and (iv) a broader range of authors and genres (such as fan fiction, semi-personal diaries, blogs, and forum discussions) than can be found in traditional corpora.

For these reasons, it is of great importance to control the quality of the material included in a web corpus and to assess its usefulness for linguistic purposes. Section 4.1 describes how general corpus statistics can be used to select high-quality text and detect problematic material in a web corpus. Section 4.2 presents a comparative evaluation of a number of existing web corpora, which are tested on the linguistic task of collocation identification.

4.1 Corpus Statistics

In this section, we briefly describe several straightforward methods for comparing corpora based on quality metrics, and show how these metrics can be used to identify problems with corpus quality.

Corpora can be characterized – and thus compared – by a wide range of statistical parameters. Three types of conclusions can be drawn from such corpus statistics:

- When comparing similar corpora (same language, genre, and size): If they differ in some key parameters, this may indicate quality problems.
- When comparing corpora of the same language, but with different genre or subject area: Parameters that correlate with genre or subject area can be identified.
- When comparing corpora of different languages: Systematic correlations between parameters can be identified. Features that are relevant for typological classifications or the classification into language families can be identified.

Furthermore, such corpus statistics can also be used to assess corpus quality. Quality assurance is a major challenge, especially when dealing with hundreds of corpora, possibly containing millions of sentences each, which can hardly be evaluated by hand [35]. For example, extreme values for certain statistics are possible indicators of problematic/noisy objects which require further inspection. Such words, sentences, or even whole documents are often of dubious quality for most types of linguistic research, and removing such objects can therefore improve corpus quality. In other cases, the distribution of some parameter is expected to follow a smooth curve, and any sharp peak can be a good indicator that more in-depth checking is required. Typical statistics used for assessing corpus quality are:

- distribution of word, sentence, or document lengths;
- distributions of characters or n-grams; and
- agreement with certain empirical laws of language such as Zipf's Law [96].

If anomalies are found by examining such statistics, further cleaning can be applied to rectify the problems which cause the anomalies. In particular, such metrics can be indicative of necessary modifications of the processing chain.

For the Leipzig Corpora Collection, there is a *Technical Report Series on Corpus Creation* that suggests various corpus statistics as indicators of corpus quality.[32] The following features turned out to be good indicators for potential flaws in corpora:

- C1: largest domains represented in the corpus and their size
- C2: number of sources per time period (in the case of continuous crawling)
- W1: graph showing the length distribution of words
- W2: list of n most frequent words in the corpus
- W3: list of longest words among the n most frequent & longest words overall
- W4: words ending in a capitalized stop word
- A1: list of characters (alphabet) and their frequencies
- A2: possibly missing abbreviations (left neighbors of the full stop with additional internal full stops)
- S1: shortest and longest sentences
- S2: length distribution of sentences (measured both in words and in characters)

The inspection of such statistics can reveal different kinds of preprocessing problems:

- problems with crawling (C1, C2),
- wrong language (W3, A1),
- wrong character set (A1, W3, S1),
- near-duplicate sentences (S2),
- problems with sentence segmentation (W4, S1),
- predominance of a certain subject area (C1, W2),
- many malformed sentences (W4, A2, S1, S2).

Figure 4 shows an example for feature S2. We computed the sentence length distribution (in characters) for two corpora. The Hindi news corpus shows the expected shape of the distribution, while the distribution in the Sundanese Wikipedia is highly atypical. On closer examination, the peaks turned out to be the result of boilerplate material and near duplicates, which should have been removed.

4.2 Linguistic Evaluation: The Collocation Identification Task

The linguistic usefulness of web corpora as representative samples of general language (rather than web-specific genres) can be evaluated by comparison with a traditional reference corpus, using frequent general-language words and constructions as test items. The underlying assumption is that an ideal web corpus should agree with the reference corpus on the syntactic and lexical core of the language, while offering better coverage of

[32] http://asvdoku.informatik.uni-leipzig.de/corpora/index.php?id=references

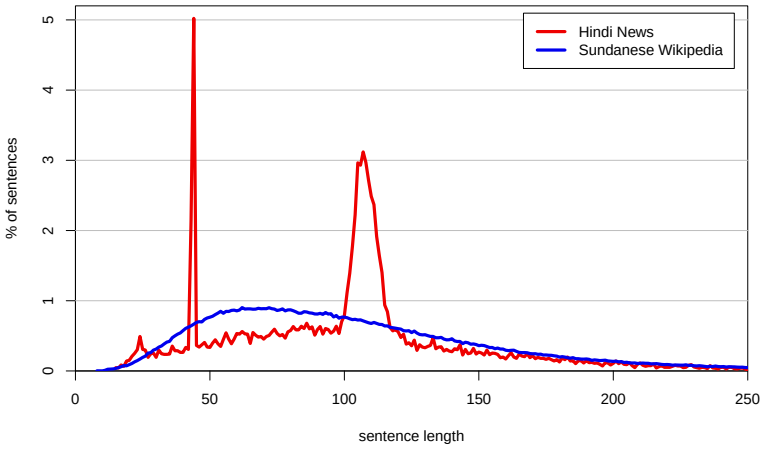

Figure 4: Distribution of sentence lengths (measured in characters) in a corpus of Hindi newspapers vs. a corpus made from the Sundanese Wikipedia.

less frequent words and construction, highly specialized expressions, and recently coined words. Evaluation criteria range from direct correlation of frequency counts [61] to assessing the benefit of web-derived data for a natural language processing application such as measuring semantic textual similarity [9, 57].

For the experiments reported here, we selected a task that plays a central role in the fields of corpus linguistics and computational lexicography: the automatic identification of collocations and other lexicalized multiword expressions (MWE) based on statistical association scores that are computed from the co-occurrence frequency of a word combination within a specified span and the marginal frequencies of the individual words. This task has two important advantages: (i) unlike frequency correlation, it allows us to assess the linguistic quality of the web data, not just its surface similarity to a reference corpus; (ii) unlike complex NLP applications, the evaluation results directly reflect corpus frequency data and do not depend on a large number of system components and parameters.

We evaluated a number of English-language web corpora that differ in their size, composition and annotation (cf. Sec. 2.3): 0.9 billion tokens from the English corpus of the *Leipzig Corpora Collection* (LCC) [52]; the ukWaC corpus of British English web pages compiled by the WaCky initiative [11];[33] the aggressively filtered UMBC WebBase corpus [57]; a preliminary version of the public release of UKCOW2012 [85]; and the Google Web 1T 5-gram database [24]. As a point of reference, we used the British

[33] Due to technical limitations of the corpus indexing software, only the first 2.1 billion tokens of the corpus could be used, omitting approx. 5% of the data.

name	size (tokens)	corpus type	POS tagged	lemmatized	basic unit
BNC	0.1 G	reference corpus	+	+	text
WP500	0.2 G	Wikipedia	+	+	fragment
Wackypedia	1.0 G	Wikipedia	+	+	article
ukWaC	2.1 G	web corpus	+	+	web page
WebBase	3.3 G	web corpus	+	+	paragraph
UKCOW	4.0 G	web corpus	+	+	sentence
LCC	0.9 G	web corpus	+	−	sentence
LCC ($f \geq k$)	0.9 G	web n-grams	+	−	n-gram
Web1T5 ($f \geq 40$)	1000.0 G	web n-grams	−	−	n-gram

Table 1: List of corpora included in the linguistic evaluation, with size in billion tokens, type of corpus, linguistic annotation and unit of sampling.

National Corpus (BNC), a balanced sample of written and spoken British English from the early 1990s, comprising a total of 110 million tokens of text [5]. We also included Wackypedia, a corpus derived from a 2009 snapshot of the English Wikipedia.[34] This represents an alternative approach to building large and up-to-date general-language corpora from online sources. While similar in size to the corpora obtained from web crawls, it covers a much narrower range of genres and should consist mostly of standard-conformant edited written English. Finally, WP500 is a subset of the Wackypedia corpus containing the first 500 words from each article, resulting in a much smaller (230 million instead of 1 billion tokens) and more balanced sample.[35] Table 1 summarizes characteristics of the corpora included in the evaluation.

The comparison of five different web corpora allows us to study the influence of different aspects of web corpus compilation with respect to linguistic usefulness. Corpus size ranges from less than 1 billion words (LCC) to 4 billion words (UKCOW), while Web1T5 offers two orders of magnitude more data. All corpora except for Web1T5 were automatically tagged with part-of-speech annotation; ukWaC, WebBase and UKCOW were also lemmatized. LCC, WebBase and Web1T5 were compiled from broad, unrestricted crawls of the web, while ukWaC and UKCOW were restricted to the `.uk` domain. The corpora also represent different strategies for selecting and cleaning web pages, as well as different levels of de-duplication (entire web pages for ukWaC, paragraphs for WebBase, and individual sentences for UKCOW and LCC).

N-gram databases such as Web1T5 are an increasingly popular format of distribution: they circumvent copyright issues, protect the intellectual property of the resource owner, and are usually much smaller in size than the underlying corpus. Such databases record the frequencies of all word forms (or lemmas) in the corpus, as well as those

[34] `http://wacky.sslmit.unibo.it/doku.php?id=corpora`

[35] In addition to truncating each article, disambiguation pages, listings and "messy" pages were removed with heuristic filters.

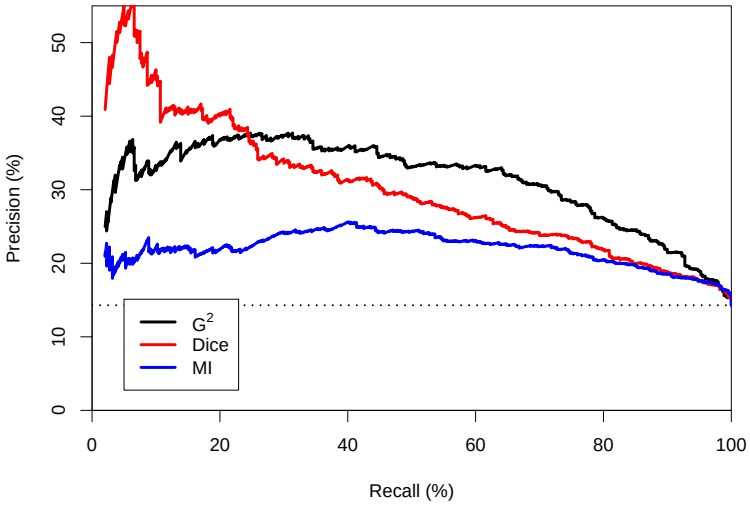

Figure 5: Precision-recall graphs for identification of lexicalized verb-particle combinations (VPC) based on co-occurrence data from British National Corpus.

of bigrams, trigrams, etc. of consecutive words. Co-occurrence frequencies for word pairs can be computed by summing over suitable n-grams of different sizes [38]. With a 5-gram database such as Web1T5, the largest possible span size is 4 words. However, most available databases omit n-grams below a certain frequency threshold in order to keep the amount of data manageable (e.g. $f < 40$ in the case of Web1T5). As a result, co-occurrence frequencies for spans of more than a single adjacent word are systematically underestimated; [38] refers to such data as quasi-collocations. In order to assess the usefulness of quasi-collocations obtained from n-gram databases, we collected all n-grams with $f \geq 5$ and $f \geq 10$ from LCC as described in Sec. 3.1. They are contrasted with the full-text LCC corpus in the evaluation below.

Our evaluation follows the methodology proposed by [39, 40]. It is based on a gold standard of candidate MWE, among which all true positives have been identified by human raters. The candidates are then ranked according to various statistical association measures computed from their observed and expected co-occurrence frequencies in a given corpus. Ideally, all true positives should be ranked at the top of the list (high association scores) and non-lexicalized word combinations at the bottom of the list (low association scores). In order to quantify this intuition, n-best lists of the highest-ranked candidates are evaluated in terms of precision (= percentage of true positives in the n-best lists) and recall (= percentage of true positives in the gold standard that are also found in the n-best list).

Results for many different values of n can be collected in precision-recall graphs as

shown in Fig. 5. Each point on the red curve represents some n-best list according to the Dice association measure. Its position on the x-axis specifies the recall achieved by this n-best list; its position on the y-axis specifies the precision achieved by the n-best list. Different association measures can thus be compared at a glance. For a meaningful interpretation, the precision-recall graphs should always be compared to the baseline precision (i. e. the overall percentage of true positives in the gold standard) indicated by the dashed horizontal line.

Fig. 5 illustrates a fairly typical situation in which no single best association measure can be identified. The red curve achieves highest precision for low recall values, whereas the black curve is considerably better for high recall values. In order to ensure a unique ranking of association measures, we use average precision (AP) as a global evaluation criterion. This measure averages precision values across all recall points from 0% to 100%; it corresponds to the area under the precision-recall graph. High AP indicates a good ranking of the candidates and a well-balanced trade-off between precision and recall. An ideal association measure would achieve an AP of 100% by collecting all true positives at the top of the ranking. A random shuffling of the candidates leads to an AP close to the baseline precision. In Fig. 5, the red curve has an AP of 30.13% and the black curve has an AP of 31.06%; the baseline AP is 14.29%. We would thus conclude that the ranking corresponding to the black curve is slightly better on the whole than the ranking corresponding to the red curve.

We evaluate the web corpora on two collocation identification tasks that focus on different aspects of multiword expressions and different types of data. The first task is concerned with the distinction between compositional and non-compositional verb-particle combinations (VPC). It shows whether the corpus-based measures help to separate semantically opaque word combinations from semi-compositional or habitual collocations, and whether they are suitable for combinations involving highly frequent grammatical words (the particles). The second task is concerned with a more intuitive notion of collocations as habitual word combinations [45]. This task shows how well corpus-based measures correspond to the intuitions of lexicographers collected in a collocations dictionary [18], and whether they are suitable for low-frequency combinations of content words. It is also relevant for applications in language education, where such collocations help to improve the language production skills of advanced learners [17].

English verb-particle combinations Our first evaluation study is based on a gold standard of 3,078 English verb-particle combinations that were manually classified as non-compositional (true positive, e. g. *carry on*, *knock out*) or compositional (false positive, e. g. *bring together*, *peer out*) [8]. This data set has previously been used for evaluation purposes by the multiword expressions community and thus allows a direct comparison. For example, [79] report a best result of 26.41% AP based on a subset of the BNC, which is clearly outperformed by the web corpora in our study (cf. Table 2).

We extracted co-occurrence counts for the word pairs from each corpus, using a 3-word span to the right of the verb (L0/R3 in the notation of [37]), which gave consistently best results in preliminary experiments. POS tags were used to filter the corpus data if

corpus	POS filter	size (tokens)	average precision G^2	t	MI	Dice	MI^2	X^2
BNC	+	0.1 G	31.06	29.15	22.58	30.13	30.97	**32.12**
WP500	+	0.2 G	28.01	25.73	27.81	30.29	29.98	**31.56**
Wackypedia	+	1.0 G	28.03	25.70	27.39	30.35	30.10	**31.58**
ukWaC	+	2.2 G	30.01	27.82	25.76	30.54	30.98	**32.66**
WebBase	+	3.3 G	30.34	27.80	27.95	31.74	32.02	**33.95**
UKCOW	+	4.0 G	32.31	30.00	26.43	32.00	32.96	**34.71**
LCC	+	0.9 G	25.61	24.83	22.14	**26.82**	25.09	26.38
LCC ($f \geq 5$)	+	0.9 G	26.95	26.45	25.54	**27.78**	25.96	27.66
LCC ($f \geq 10$)	+	0.9 G	27.34	26.81	27.13	27.85	25.95	**28.09**
LCC	−	0.9 G	24.67	23.63	21.41	25.36	23.88	**25.63**
LCC ($f \geq 5$)	−	0.9 G	25.45	24.79	23.54	**26.30**	24.55	26.21
LCC ($f \geq 10$)	−	0.9 G	25.84	25.16	25.28	26.49	24.71	**26.63**
Web1T5 ($f \geq 40$)	−	1000.0 G	26.61	26.12	21.67	**27.82**	25.72	27.14

Table 2: Evaluation results (in terms of average precision) for identification of lexicalized verb-particle combinations (VPC). The best result for each corpus is highlighted in bold font. The baseline precision for this task is 14.29%.

available (first word tagged as lexical verb, second word tagged as preposition, particle or adverb). The verbs were lemmatized if possible. For corpora without lemma annotation, morphological expansion was applied, i.e. frequency counts for all inflected forms of each verb were aggregated. Since the performance of an association measure can vary in unpredictable ways for different corpora and evaluation tasks, we present results for a range of standard, widely-used association measures: the log-likelihood ratio (G^2), t-score (t), Mutual Information (MI), the Dice coefficient (Dice), a heuristic variant of Mutual Information (MI^2) and Pearson's chi-squared test with continuity correction (X^2). See [37] for details and references.

Table 2 summarizes the evaluation results for the VPC task. The average precision of the best association measure for each corpus is highlighted in bold font. The overall best result is obtained from the UKCOW web corpus, with an AP of 34.71%. Comparing Wackypedia, ukWaC, WebBase, and UKCOW, we find that AP increases with corpus size for the web-derived corpora. Scaling up does indeed pay off: the largest corpora clearly outperform the reference corpus BNC. However, they need more than an order of magnitude more data to achieve the same AP of 32.12%.

Surprisingly, Web1T5 achieves only low precision despite its huge size. There are several possible explanations for this observation: distribution as an n-gram database with frequency threshold, poor quality of the corpus (i. e. little filtering and cleaning of web pages), as well as lack of POS tagging and lemmatization. The first hypothesis is clearly ruled out by the LCC results, which show no detrimental effect for n-gram frequency thresholds of $f \geq 5$ and $f \geq 10$.[36] In fact, AP improves by more than 1%

[36] Taken in relation to corpus size, these thresholds are much more aggressive than the $f \geq 40$ threshold of Web1T5.

when the thresholds are applied. Evaluating LCC with and without a POS filter, we found a small benefit from using the annotation. Reliable lemmatization appears to be even more important and might account for the considerably lower AP of LCC compared to the similar-sized Wackypedia.

We can summarize the conclusions from this first evaluation study as follows. Web corpora indeed seem to be a valid replacement for a traditional reference corpus such as the BNC, provided that suitable strategies are used to select and clean up web pages and the resulting corpus is enriched with linguistic annotation. The precise implementation of this procedure and the level of de-duplication do not appear to play an important role: AP grows with corpus size regardless of other parameters. Nonetheless, diversity may be a relevant factor. Web corpora of more than a billion words are needed to achieve the same performance as the 100-million-word BNC, and WP500 discards almost 80% of the Wackypedia text without a noticeable loss of precision. Web1T5 shows that sheer size cannot make up for "messy" content and lack of annotation. The distribution format of an n-gram database is not detrimental *per se* for MWE identification and similar tasks, though.

BBI Collocations The second evaluation study is based on a gold standard of 36,328 two-word lexical collocations from the BBI Combinatory Dictionary [18], which are close to the notion of collocation put forward by J. R. Firth (cf. [13]) and have important applications in language education [17]. In contrast to the first task, there is no fixed list of candidates annotated as true and false positives: a different candidate set is extracted from each corpus and evaluated by comparison with the gold standard. Candidate sets were obtained by collecting all co-occurrences of nouns, verbs, adjectives and adverbs within a span of three words to the left and right (L3/R3), allowing only words that appear in one or more entries of the BBI dictionary.[37] POS tags were used as a filter if available. Since the gold standard does not distinguish collocations between lemmas from collocations that are restricted to particular word forms, all candidate pairs were lemmatized. For corpora without lemma annotation, morphological expansion was used to aggregate frequency counts for all possible inflected forms of each candidate pair, based on word form–lemma correspondences obtained from automatically tagged and lemmatized corpora.

In order to ensure a fair comparison, all candidate lists were truncated to the most frequent 1 million word pairs. While this reduces coverage of the gold standard for the larger Web corpora, our previous experience suggests that high-frequency candidates can be ranked more reliably than lower-frequency data, resulting in better AP scores if frequency thresholds are applied. Preliminary experiments with the WebBase corpus showed quite stable results for candidate lists ranging from 1 million to 10 million word

[37] [13] evaluated different span sizes (3, 5 and 10 words) on a subset of the BBI data, obtaining better results (i.e. higher AP) for smaller spans. For the present experiment, we decided to use a three-word span as a good compromise between precision and coverage. This choice was corroborated by additional preliminary tests on the full BBI data with span sizes ranging from 1 to 10 words.

corpus	size (tokens)	average precision (up to 50% recall) G^2	t	MI	Dice	MI^2	X^2
BNC	0.1 G	22.76	18.48	16.68	20.26	**25.16**	24.89
WP500	0.2 G	21.68	17.08	16.31	19.71	**24.41**	24.33
Wackypedia	1.0 G	21.45	16.90	17.41	19.98	**24.64**	24.56
ukWaC	2.2 G	17.22	13.58	15.63	16.30	**20.39**	20.33
WebBase	3.3 G	19.30	15.06	17.77	18.21	22.77	**22.88**
UKCOW	4.0 G	20.65	16.35	18.28	19.29	**24.15**	24.04
LCC	0.9 G	15.02	12.11	13.58	14.22	**17.78**	17.67
LCC ($f \geq 5$)	0.9 G	15.80	13.35	13.99	15.22	**18.59**	18.50
LCC ($f \geq 10$)	0.9 G	15.72	13.44	14.46	15.30	**18.51**	18.39
Web1T5	1000.0 G	11.95	10.73	11.08	11.38	**13.50**	13.18

Table 3: Evaluation results for identification of BBI collocations. The best result for each corpus is highlighted in bold font.

pairs. Since coverage (and thus the highest recall percentage that can be achieved by a ranking) differs between corpora, it is not meaningful to specify a baseline precision. As a global evaluation criterion, we use average precision for recall points up to 50% (AP50).

The gold standard used here is not without problems. Recent collocations found in the web corpora may not be attested due to the age of the BBI dictionary (published in 1986). Moreover, true positives may be missing due to introspective bias or the space constraints of a paper dictionary. Manual inspection of small samples from the ranked candidate lists revealed hardly any candidates that were marked as false positives despite being clearly collocational in present-day English. We believe therefore that the AP scores of web corpora are underestimated by less than 1%, a margin of error that has no substantial influence on the conclusions drawn from the evaluation. However, a more comprehensive manual validation and quantitative analysis is certainly desirable and is planned for future work.

The results of the second study are shown in Table 3. In this case, corpus composition plays a much more important role than size. Comparing the web corpora LCC, ukWaC, WebBase, and UKCOW, we find a linear increase of AP50 with corpus size that does not seem to depend strongly on other aspects of corpus compilation. However, the 4-billion word UKCOW corpus is still outperformed by the reference corpus BNC. Even a 230-million-word subset of Wikipedia (WP500) is better suited for the identification of lexical collocations than web corpora. The results for LCC confirm our previous observation that n-gram databases with frequency thresholds need not be inferior to full-text corpora, provided that the analysis can be carried out within a window of 5 words. Finally, the extremely low precision of Web1T5 can be explained by its "messy" content and the lack of POS tagging.

In summary, our evaluation study has shown that web corpora can be a valid

replacement for a traditional reference corpus such as the BNC, with much better coverage and up-to-date information. This is not simply a matter of scaling up, though. Unless web pages are carefully selected, cleaned, de-duplicated and enriched with linguistic annotation, even a gigantic text collection such as Web1T5 offers little value for linguistic research. In some tasks, web corpora are still inferior to traditional corpora (and even to other online material such as Wikipedia articles), but this is likely to change with a further increase in corpus size.

5 Working with Web Corpora / Linguistic Applications

5.1 Accessing Very Large Corpora

For most linguistic applications of web corpora, an essential step is to scan the text and annotation with sophisticated search patterns, often based on linguistic annotation such as part-of-speech tags. Ideally, it should be possible to execute simple queries interactively and obtain the search results within a matter of seconds. Several dedicated corpus query engines can be used for this purpose, such as the IMS Open Corpus Workbench, the NoSketch Engine, and Poliqarp.[38,39,40] Most of these query engines will comfortably handle corpora of several billion words even on state-of-the-art commodity hardware, which is sufficient for all publicly available web corpora with linguistic annotation (including, in particular, LCC, UKCOW, ukWaC, and UMBC WebBase). They can easily be scaled up to much larger amounts of data by sharding the text across a grid of servers (with simple wrapper programs to collect and merge result sets). Specialized, highly scalable web/text search engines such as Apache Lucene[41] do not offer the expressiveness and complexity required by most linguistic corpus queries.

For certain applications, it can be convenient to provide web corpora in the form of pre-compiled n-gram databases. Because of their tabular form, such data sets can easily be stored in a relational database, enabling indexed retrieval and flexible aggregation of frequency counts. See [38, 66] for suggestions and a suitable database design. Compact storage and interactive access for full n-gram counts from trillion-word web corpora can only be achieved with specially designed software packages [80, 47].

5.2 Corpus Linguistics and Theoretical Linguistics

Compiling static web corpora in conjunction with an efficient retrieval system and a powerful query language have been argued to be the best way of exploiting web data for linguistic purposes [69]. While large web corpora have unique features to offer to the linguist and are thus particularly attractive in many respects (see Section 4), there are also a number of potential disadvantages (or, more neutrally speaking, characteristics), which have consequences for their use in linguistic research.

[38] http://cwb.sourceforge.net/
[39] http://nlp.fi.muni.cz/trac/noske/
[40] http://poliqarp.sourceforge.net/
[41] http://lucene.apache.org/core/

Composition A long-standing and much-discussed problem is the question of representativeness: In order to make valid inferences about a particular language on the basis of corpus data (in terms of fundamental research), corpus linguists often require a general purpose corpus to be representative of that language. For other, more task-oriented linguistic applications (such as extracting collocations or creating thesauri), representativeness might not strictly be required, as long as reasonably good results are obtained from an application perspective. In traditional corpus creation, representativeness is usually approached by following an elaborated stratified sampling scheme (cf. also Section 2.1), resulting in a corpus that has the desired composition in terms of genres, topic domains, and socio-linguistic factors. The difficulty lies in determining the "correct" composition in the first place, and different proposals exist as to how the importance of individual text types should be assessed. Seminal papers include [6, 19]. [68] discusses representativeness in web corpora. For a critical stance on representativeness and balance, see [60]. A recently suggested new approach [63] is characterized by the creation of a very large *primordial sample* with a lot of meta data which is in principle relevant for stratification. From this sample, users can then create specialized *virtual corpora* according to their desired sampling frame. This is an interesting idea for web corpora as well, because they are usually very large. However, the lack of suitable meta data stands in the way of the creation of virtual corpora. Also, if the primordial web document sample itself is heavily biased (cf. 2.2), then it is unclear whether the virtual corpora can adequately model the user's sampling frame.

That said, in the real-world construction of web corpora, there are at two major design procedures with respect to corpus composition:

Random The corpus should be a "random sample" from the population of web texts in a particular language. It should thus reflect the distribution of text types, topics etc. of (a particular segment of) the web. What can we expect then, in comparison to other corpora? An impressionistic estimate is given in [46], according to which legal, journalistic, commercial, and academic texts make up for the major part of prose texts on the web. If true, it would not be unreasonable to expect a web corpus built from a random sample to be comparable in terms of composition to traditional, "general purpose" corpora, since these kinds of texts dominate the composition of traditional corpora as well. On the other hand, [88] finds that, in a given web corpus, texts about the arts, humanities, and social sciences are under-represented compared to the BNC, while texts from technical fields are over-represented. However, this estimate is based on a web corpus made from a breadth-first crawl that was seeded with URLs from search engines. It is thus not likely to be representative of the population of texts on the web, as has been argued in Section 2.2.

Balanced The corpus should be balanced in such a way that no genre or topic is heavily over-represented. Such an approach is taken, e. g. by [31], who argue that the distribution of genres and topics on the web is probably different from what should be contained in a general purpose corpus, and thus a random sample from the population of web pages is not actually desirable. The aim of creating a corpus that is balanced in this sense conflicts with the idea of statistical representativeness.

Building a corpus by random crawling still allows for statements about the composition, however, but they can only be made after the corpus has been constructed, and they have to be based either on estimates from small hand-annotated samples or on automatic classification. See Table 4, reproduced from [85] for an assessment of the genre/text type composition of DECOW2012 and ESCOW2012 (Spanish) based on a small manually annotated sample. The classification scheme is based on the scheme suggested in [88] with only a few added categories. Notice the high number of documents for which authorship cannot be determined as well as the (surprisingly) low proportion of truly fictional texts. The *Audience* classification is a simple evaluation of a document's readability. The *Quasi-Spontaneous* classification for *Mode* refers to forum discussions and the like, whereas *Blogmix* refers to web pages containing a mix of written text plus a forum-like discussion. The amount of documents containing potentially non-standard language is thus estimated at 27% for DECOW2012. Interestingly, the substantial difference to the Spanish ESCOW2012 in this regard (only 11.5%) could be an interesting result related to the linguistic and socio-linguistic characterization of national top-level domains. However, given that biased crawling algorithms and potentially biased search engine-derived seed URLs were used, this must be taken with a grain of salt.

Metadata One of the most serious drawbacks of web corpora is their almost complete lack of document metadata, which is essential for a thorough linguistic interpretation. Such metadata includes information about date of publication/utterance, age, sex, etc. of the author/speaker, text type, and topic domain. Linguists often adjust their hypotheses according to the distribution of such categories in a given corpus. However, no such data is encoded in web documents in a reliable and standardized way. As an additional challenge, some web documents resist a straightforward classification along such traditional categories because of their complex editing history (e. g., multiple authors producing subsequent versions of a text over a period of time). In any event, the only feasible way to gather such metadata for each document in a crawled web corpus is automatic document classification. This can be done with high accuracy for some dimensions of classification, but it is never free of errors. Although some corpus linguists might find this problematic, the size of web corpora usually allows users to extract concordances large enough to make any hypotheses testable with high reliability despite such error rates, and automatic generation of metadata is not just the only option, but in fact a valid one. Furthermore, it is worth pointing out that the lack of metadata is not unique to web corpora. Consider, for instance, corpora containing mostly newspaper articles (like the German DeReKo [63]), where authorship cannot always be attributed to specific individuals.

Document structure The suitability of a web corpus for a specific area of linguistic research depends to a considerable extent on the methods of non-linguistic post-processing applied to the raw data, cf. Section 3. For instance, if we remove boilerplate or treat sentences containing emoticons as noise and delete them, then we run the risk

	DECOW2012		**ESCOW2012**	
Type	%	CI ±%	%	CI ±%
Authorship				
Single, female	**6.0**	2.8	**5.0**	2.5
Single, male	**11.5**	3.7	**16.5**	4.3
Multiple	**36.0**	5.6	**16.5**	4.3
Corporate	**21.0**	4.7	**20.5**	4.7
Unknown	**25.5**	5.0	**41.5**	5.7
Mode				
Written	**71.0**	5.0	**86.0**	4.0
Spoken	**1.0**	3.0	**2.5**	1.8
Quasi-Spontaneous	**22.5**	4.9	**3.5**	2.1
Blogmix	**4.5**	2.4	**8.0**	3.2
Audience				
General	**75.5**	5.0	**94.0**	2.8
Informed	**17.0**	4.4	**2.5**	1.8
Professional	**7.5**	3.0	**3.5**	2.1
Aim				
Recommendation	**12.5**	3.8	**7.0**	3.0
Instruction	**4.5**	2.4	**6.0**	2.8
Information	**36.0**	5.5	**41.5**	5.7
Discussion	**47.0**	5.8	**44.5**	5.8
Fiction	**0.0**	0.0	**1.0**	1.2
Domain				
Science	**2.5**	1.8	**5.0**	2.5
Technology	**14.0**	4.0	**6.5**	2.9
Medical	**4.5**	2.4	**4.0**	2.3
Pol., Soc., Hist.	**21.5**	4.8	**21.0**	4.7
Business, Law	**10.0**	3.5	**12.5**	3.8
Arts	**8.5**	3.2	**8.5**	3.2
Beliefs	**5.0**	2.5	**3.0**	2.0
Life, Leisure	**34.0**	5.5	**39.5**	5.7

Table 4: Text category/genre distribution in DECOW2012 and ESCOW2012 with 90% confidence interval ($n = 200$)

of removing material (e. g. headings or even entire paragraphs) that would have been crucial in the linguistic analysis of discourse-related phenomena, such as co-reference, information structure, and rhetorical structure. Similarly, research on text types and genres is likely to be affected by the removal of such material. Even studies on syntax may require the context of individual sentences to be available, for instance when examining the syntax of sentence connectors. The same is true for some computational linguistics tasks such as distributional semantics based on larger units than sentences (like LSA [67]). Corpora of single sentences are inadequate for such research. Unfortunately, even corpora containing full documents like the COW corpora have often to be distributed in a shuffled form (randomly sorted single sentences) to avoid legal problems with copyright claims. At least for tasks in computational linguistics, corpora containing full documents within which the order of the sentences has been randomized might be a viable solution, as mentioned in Section 3.2.

Duplication Web corpora consisting of complete documents usually contain a certain amount of duplicated material, even if some form of de-duplication was applied in post-processing. Again, this kind of problem concerns non-web corpora as well, albeit to a much lesser extent. For example, newspaper corpora sometimes contain multiple instances of news agency material that was reprinted by several newspapers (see, e.g. 62 on near-duplicate detection in large, non-web corpora). As described in Section 3, in a web corpus (containing only individual sentences) duplication can be dealt with by removing all but one instance of each sentence. The publicly available versions of the COW corpora are distributed in a similar way. The frequency of each sentence in the full corpus is also recorded as metadata, allowing users to reconstruct statistical information for frequent sentences that are not boilerplate, like *Hello!* or *Thank you.*

5.3 Quantitative Typology and Language comparison

The availability of corpora for a large variety of languages is a necessary requirement for performing typological studies. Creating high-quality web corpora, as described in this article, opens up further possibilities for this field.

Linguistic typology is concerned with the classification of the world's languages into types based on phonological, morphological, syntactic and other features [55]. This allows researchers to gain insights into the structure of language by studying possible or preferred types [32]. A main objective of quantitative typology is the search for systematic patterns of variation of language and the linguistic features they are based on. Thus it aims at finding, for example, absolute or statistical language universals [90, 36]. Absolute universals are rules which necessarily hold for all languages. Therefore, they are rare and typically of a very general nature. Statistical universals, on the other hand, describe preferences on a global or local scale. Among them are conditional universals (or implicational universals), which make assumptions about combinations of types or features that are typically preferred or avoided. Many studies in quantitative typology are concerned with parameters of language and their relations. Some works

are concerned with systematic interlingual correlations between the different parameters measured on the text resources [42, 43]. Correlations between measured parameters and typological parameters are also analyzed [43, 53].

In most cases, the starting point for research in quantitative typology is collections of textual resources, which are usually manually created. By using automatically collected web corpora instead, the necessary manual work can be greatly reduced. Based on these corpora simple, features like sentence, word, syllable, morpheme, etc. can be determined. While for many properties nearly exact values can be computed, some can only be approximated - especially when independence of language is aspired [51]. Possible features are:

- average word length in characters,
- average sentences length in words or characters,
- text coverage of the most frequent words,
- slope of Zipf's Law,
- different measurements of vocabulary richness,
- entropy on word and character level,
- average number of syllables per word or sentence,
- average syllable length,
- amount of affixes, or
- ratio of prefixes and suffixes.

In order to reliably compute these features, high quality web corpora are required [34], as properties such as size, text type, or subject area can have strong influence on typological investigations [51]. By keeping the main properties of corpora constant, statistical significance of typological analyses can be increased. Table 5 illustrates this influence. For each feature, it shows the ratio of cross-language standard deviation to standard deviation across other properties of corpora. Let us consider the example of average sentence length in words. First, we calculate the standard deviation of this feature across languages. In addition, we determine the standard deviation when varying corpus size (from 10,000 to 10,000,000 sentences), text type (random web text, Wikipedia, ...), or subject area (culture, science, economy, politics, ...). Since for most features a high variation between languages is expected, the ratio of the standard deviations should typically be larger than one. In the case of average sentence length in words, we measure a cross-language standard deviation which is 13 times larger than when modifying the subject area of the texts analyzed. Compared to the standard deviation dependent on corpus size it is even 107 times larger.

Utilizing methods like tests of significance, relationships with classical typological parameters - some kind of language universals - can be discovered. The following examples were identified by analyzing corpora in 700 languages and comparing measured features to available classical typological parameters taken from the World Atlas of Language Structures.[42] We found significant relations between:

[42] http://wals.info

measurement	language / corpus size	language / text type	language / subject area
average sentence length in words	107.41	8.65	13.20
average sentence length in characters	77.03	6.23	7.67
ratio of suffixes an prefixes	18.78	17.69	25.84
syllables per sentence	30.25	8.22	7.33
Type-Token Ratio	1.16	8.21	6.13
Turing's Repeat Rate	238.95	6.37	8.69
slope of Zipf's Law	3.27	11.35	11.25
text coverage of the top 100 words	530.85	7.93	8.75

Table 5: Comparison of standard deviations of corpus-based measurements. Quotient of cross-language standard deviation and other properties of corpora such as corpus size. Values larger than 1 imply a higher cross-language standard deviation.

- ratio of suffixes and prefixes and position of case marking (end of word vs. beginning of word): $p<0.001\%$, mean values of 10.48 and 0.7 and sample sizes of 57 and 11,
- average length of words of a language and its morphological type (concatenative vs. isolating): $p<1\%$, mean values of 8.43 and 6.95 and sample sizes of 68 and 8,
- measured amount of affixation of a language and its morphological type (concatenative vs. isolating): $p<0.5\%$, mean values of 21.20 and 10.06 and sample sizes of 68 and 8,
- average number of syllables per sentence and word order (SOV vs. SVO): $p<0.001\%$, mean values of 56.95 and 45.27 and sample sizes of 111 and 137,
- average number of syllables per word and morphological type (concatenative vs. isolating): $p<5\%$, mean values of 2.06 and 1.64 and sample sizes of 68 and 8 and
- average number of syllables per sentence and morphological type (concatenative vs. isolating): $p<5\%$, mean values of 21.76 and 27.47 and sample sizes of 68 and 8.

Typological parameters can also be predicted based on measurements on corpora. Using methods such as supervised machine learning, knowledge about different features of corpora can be combined to determine typological properties of a language. Table 6 shows results of the prediction of morphological type.

In addition, the branch of quantitative typological work concerned with vocabulary-based language comparison uses orthographic or phonetic similarity of words or letter n-grams to measure the similarity or relatedness of language pairs. Word lists for such analyses are usually compiled manually [94, 95, 27], but can also be extracted from corpora [89]. However, this comes with high demands for quality and comparability of the corpora used. One the one hand contamination of corpora with text in other languages can lead to unusually high similarity values. For web corpora especially

features used	accuracy
Baseline	50.0%
Words per sentence	74.4%
Number of word forms	87.8%
Words per sentence, number of word forms, syllables per word	91.8%

Table 6: Probability of correct prediction of morphological type of a language (concatenative vs. isolating) using different features and equal sample sizes for both classes.

English remnants are a common problem. On the other hand properties of corpora such as subject area can have influence on the analysis. For very similar languages such as the North Germanic genus, usage of varying subject areas across languages can lead to changes in resulting genus internal ranking of similarities.

6 Summary and Outlook

As discussed in this article, the availability of large corpora is a prerequisite for research in empirical linguistics as well as language technology. We showed how very large, high quality corpora can be constructed from the web despite of all the noise that makes the construction process a challenging enterprise. We focused on three main problems: crawling, processing, and quality control. Crawling is especially important if web corpora for a variety of languages, not just for English, should be created. We then introduced two projects, WebCorpus and COW, that both allow creating large-scale, linguistically annotated corpora from crawled data. We argued that when working with web data, controlling the quality of the resulting corpus is an important issue that we addressed with corpus statistics and a linguistic evaluation based on two collocation identification tasks. Finally, we showed how the availability of extremely large, high-quality web corpora fits in with research in various fields of linguistics, computational linguistics, and natural language processing.

Taking it all together, several desiderata for web corpora creation clearly emerge. Crawling procedures can be optimized in diverse ways, depending on corpus design goals. Focused and optimized crawling should be implemented to discover more interesting data using less bandwidth, storage, and time resources. This is especially true for smaller languages, for which documents are hard to find using non-selective crawling. Bias-free crawling should be used to derive truly representative samples from the web for fundamental research and linguistic web characterization.

The selection of optimal tools and models for linguistic post-processing and their integration into automatable tool chains (for example the UIMA architecture) is not yet completed for many languages. Also, the evaluation of the accuracy of such tools on web data is important under any corpus design goal, and researchers in this area can greatly benefit from joining efforts. For example, to make the COW processing

chain use the advantages of the WebCorpus technology, the COW annotators would just have to be added to the UIMA framework. Further linguistic annotations (like chunking and parsing) as well as automatic text classification and metadata generation need to be added in a similar fashion to improve the quality of web corpora. This is a prerequisite for their acceptance within (theoretical, empirical, and computational) linguistic research communities, as well as for more detailed quantitative analysis of linguistic phenomena.

Continued work on massive parallelization using architectures like Hadoop is the only way of meeting the computational requirements for corpora in the region of 100 billion or even trillions of tokens. As we have learned from the evaluation studies described in Section 4.2, quality increases with corpus size: a high-quality web corpus of 100 billion tokens should outperform traditional reference corpora in most application tasks.

Acknowledgments

The second evaluation study reported in Section 4.2 is based on joint work with Sabine Bartsch.

References

[1] Serge Abiteboul, Mihai Preda, and Gregory Cobena. Adaptive on-line page importance computation. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 280–290, New York, NY, USA, 2003. ACM.

[2] Myriam Abramson and DavidW. Aha. What's in a URL? Genre Classification from URLs. Technical report, AAAI Technical Report WS-12-09, 2009.

[3] Dimitris Achlioptas, Aaron Clauset, David Kempe, and Cristopher Moore. On the Bias of Traceroute Sampling: or, Power-law Degree Distributions in Regular Graphs. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 694–703, New York, NY, USA, 2005. ACM.

[4] George Almpanidis, Constantine Kotropoulos, and Ioannis Pitas. Combining Text and Link Analysis for Focused Crawling – An Application for Vertical Search Engines. *Inf. Syst.*, 32(6):886–908, 2007.

[5] Guy Aston and Lou Burnard. *The BNC Handbook*. Edinburgh University Press, Edinburgh, 1998.

[6] Sue Atkins, Jeremy Clear, and Nicholas Ostler. Corpus design criteria. *Literary and Linguistic Computing*, 7(1):1–16, 1992.

[7] Ricardo Baeza-Yates, Carlos Castillo, and Efthimis N. Efthimiadis. Characterization of national Web domains. *ACM Trans. Internet Technol.*, 7(2), 2007.

[8] Timothy Baldwin. A resource for evaluating the deep lexical acquisition of English verb-particle constructions. In *Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 1–2, Marrakech, Morocco, 2008.

[9] Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. UKP: Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures. In *Proceedings of First Joint Conference on Lexical and Computational Semantics (*SEM), Montreal, Canada*, pages 435–440, Montreal, Canada, 2012.

[10] Ziv Bar-Yossef, Andrei Z Broder, Ravi Kumar, and Andrew Tomkins. Sic transit gloria telae: towards an understanding of the web's decay. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 328–337, New York, NY, USA, 2004. ACM.

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

[12] Marco Baroni, Francis Chantree, Adam Kilgarriff, and Serge Sharoff. Cleaneval: a Competition for Cleaning Web Pages. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008*, Marrakech, Morocco, 2008.

[13] Sabine Bartsch and Stefan Evert. Towards a Firthian notion of collocation. In Andrea Abel and Lothar Lemnitzer, editors, *Network Strategies, Access Structures and Automatic Extraction of Lexicographical Information. 2nd Work Report of the Academic Network Internet Lexicography*, OPAL – Online publizierte Arbeiten zur Linguistik. Institut für Deutsche Sprache, Mannheim, to appear.

[14] Eda Baykan, Monika Henzinger, Ludmila Marian, and Ingmar Weber. Purely URL-based Topic Classification. In *Proceedings of the 18th international conference on {World Wide Web}*, pages 1109–1110, 2009.

[15] Eda Baykan, Monika Henzinger, and Ingmar Weber. Web Page Language Identification Based on URLs. In *Proceedings of the VLDB Endowment*, pages 176–187, 2008.

[16] Michael Beiß wenger, Maria Ermakova, Alexander Geyken, Lothar Lemnitzer, and Angelika Storrer. DeRiK: A German Reference Corpus of Computer-Mediated Communication. In *Proceedings of Digital Humanities 2012*, 2012.

[17] Morton Benson. The structure of the collocational dictionary. *International Journal of Lexicography*, 2:1–14, 1989.

[18] Morton Benson, Evelyn Benson, and Robert Ilson. *The BBI Combinatory Dictionary of English: A Guide to Word Combinations*. John Benjamins, Amsterdam, New York, 1986.

[19] Douglas Biber. Representativeness in Corpus Design. *Literary and Linguistic Computing*, 8(4):243–257, 1993.

[20] Chris Biemann, Uwe Quasthoff, Gerhard Heyer, and Florian Holz. ASV Toolbox – A Modular Collection of Language Exploration Tools. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC) 2008*, 2008.

[21] Chris Biemann and Martin Riedl. Text: Now in 2D! A Framework for Lexical Expansion with Contextual Similarity. *Journal of Language Modelling*, 1(1), 2013.

[22] Burton H Bloom. Space/Time Trade-offs in Hash Coding with Allowable Errors. *Commun. ACM*, 13(7):422–426, July 1970.

[23] Jürgen Bortz. *Statistik für Human- und Sozialwissenschaftler*. Springer, Berlin etc., 7 edition, 2010.

[24] Thorsten Brants and Alex Franz. *Web 1T 5-gram Version 1*. Linguistic Data Consortium, Philadelphia, PA, 2006. `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13`.

[25] Andrei Broder, Steven C Glassman, Mark S Manasse, and Geoffrey Zweig. Syntactic Clustering of the Web. *Comput. Netw. ISDN Syst.*, 29(8-13):1157–1166, September 1997.

[26] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Raymie Stata, Andrew Tomkins, and Janet L Wiener. 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, 2000.

[27] Cecil H Brown, Eric W Holman, Søren Wichmann, and Viveka Velupillai. Automated classification of the worlds languages: A description of the method and preliminary results. *STUF-Language Typology and Universals*, 61(4):285–308, 2008.

[28] Soumen Chakrabarti, Byron Dom, and Piotr Indyk. Enhanced Hypertext Categorization Using Hyperlinks. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, SIGMOD '98, pages 307–318. ACM, 1998.

[29] Soumen Chakrabarti, Martin van den Berg, and Byron Dom. Focused Crawling: A New Approach to Topic-specific Web Resource Discovery. *Computer Networks*, 31:1623–1640, 1999.

[30] Junghoo Cho, Hector García-Molina, and Lawrence Page. Efficient Crawling through URL ordering. In *Proceedings of the 8th International World Wide Web Conference*, 1998.

[31] Massimiliano Ciamarita and Marco Baroni. Measuring Web-Corpus Randomness: A Progress Report. pages 127–158.

[32] Michael Cysouw. Quantitative methods in typology. In G. Altmann, R. Köhler, and R. Piotrowski, editors, *Quantitative linguistics: an international handbook*, pages 554–578. Mouton de Gruyter, 2005.

[33] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In *OSDI'04: Proceedings of the 6th Symposium on Operating Systems Design and Implementation*. USENIX Association, 2004.

[34] Thomas Eckart, Uwe Quasthoff, and Dirk Goldhahn. The influence of corpus quality on statistical measurements on language resources. In *LREC*, pages 2318–2321, 2012.

[35] Thomas Eckart, Uwe Quasthoff, and Dirk Goldhahn. Language Statistics-Based Quality Assurance for Large Corpora. In *Proceedings of Asia Pacific Corpus Linguistics Conference 2012, Auckland, New Zealand*, 2012.

[36] Halvor Eifring and Rolf Theil. *Linguistics for students of Asian and African languages*. Institutt for østeuropeiske og orientalske studier, 2004.

[37] Stefan Evert. Corpora and collocations. In Anke Lüdeling and Merja Kytö, editors, *Corpus Linguistics. An International Handbook*, chapter 58. Mouton de Gruyter, Berlin, New York, 2008.

[38] Stefan Evert. Google Web 1T5 n-grams made easy (but not for the computer). In *Proceedings of the 6th Web as Corpus Workshop (WAC-6)*, pages 32–40, Los Angeles, CA, 2010.

[39] Stefan Evert and Brigitte Krenn. Methods for the qualitative evaluation of lexical association measures. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 188–195, Toulouse, France, 2001.

[40] Stefan Evert and Brigitte Krenn. Using small random samples for the manual evaluation of statistical association measures. *Computer Speech and Language*, 19(4):450–466, 2005.

[41] Manaal Faruqui and Sebastian Padó. Training and evaluating a German named entity recognizer with semantic generalization. In *Proceedings of KONVENS 2010*, Saarbrücken, Germany, 2010.

[42] Gertraud Fenk-Oczlon and August Fenk. The mean length of propositions is 7 plus minus 2 syllables – but the position of languages within this range is not accidental. In G. D'Ydevalle, editor, *Cognition, Information Processing, and Motivation*, pages 355–359. North Holland: Elsevier Science Publisher, 1985.

[43] Gertraud Fenk-Oczlon and August Fenk. Crosslinguistic correlations between size of syllables, number of cases, and adposition order. In G. Fenk-Oczlon and Ch. Winkler, editors, *Sprache und Natürlichkeit. Gedenkband für Willi Mayerthaler*, pages 75–86. Gunther Narr, Tübingen, 2005.

[44] David Ferrucci and Adam Lally. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Natural Language Engineering*, 10(3-4):327–348, 2004.

[45] J. R. Firth. A synopsis of linguistic theory 1930–55. In *Studies in linguistic analysis*, pages 1–32. The Philological Society, Oxford, 1957.

[46] William Fletcher. Facilitating the compilation and Dissemination of Ad-hoc web corpora. In Guy Aston, Silvia Bernardini, and Dominic Stewart und Silvia Bernadini, editors, *Corpora and language learners*, pages 273–300. Benjamins, Amsterdam, 2004.

[47] Michael Flor. A fast and flexible architecture for very large word n-gram datasets. *Natural Language Engineering*, 19(1):61–93, 2013.

[48] Ian Foster. *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.

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

[50] Yoav Goldberg and Jon Orwant. A dataset of syntactic-ngrams over time from a very large corpus of English books. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*SEM 2013)*, Atlanta, GA, 2013. Data sets available from `http://commondatastorage.googleapis.com/books/syntactic-ngrams/index.html`.

[51] D. Goldhahn. *Quantitative Methoden in der Sprachtypologie: Nutzung korpusbasierter Statistiken (Doctoral dissertation)*. University of Leipzig, Leipzig, Germany, 2013.

[52] Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).

[53] Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. Finding Language Universals: Multivariate Analysis of Language Statistics using the Leipzig Corpora Collection. In *Leuven Statistics Days 2012*, KU Leuven, 2012.

[54] Daniel Gomes and Mário J Silva. Characterizing a national community web. *ACM Trans. Internet Technol.*, 5(3):508–531, 2005.

[55] Joseph H Greenberg. Some universals of grammar with particular reference to the order of meaningful elements. *Universals of language*, 2:73–113, 1963.

[56] Erla Hallsteinsdóttir, Thomas Eckart, Chris Biemann, Uwe Quasthoff, and Matthias Richter. Íslenskur orðasjóður - Building a Large Icelandic Corpus. In *Proceedings of NODALIDA-07*, Tartu, Estonia, 2007.

[57] Lushan Han, Abhay L Kashyap, Tim Finin, James Mayfield, and Johnathan Weese. UMBC_EBIQUITY-CORE: Semantic Textual Similarity Systems. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics, 2013.

[58] Monika R Henzinger, Allan Heydon, Michael Mitzenmacher, and Marc Najork. On near-uniform URL sampling. 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, 2000.

[59] Gerhard Heyer and Uwe Quasthoff. Calculating communities by link analysis of URLs. In *Proceedings of the 4th international conference on Innovative Internet Community Systems*, IICS'04, pages 151–156, Berlin, Heidelberg, 2006. Springer-Verlag.

[60] Susan Hunston. Collection Strategies and Design Decisions. In Anke Lüdeling and Merja Kytö, editors, *Corpus Linguistics. {A}n International Handbook*, pages 154–168. Walter de Gruyter, Berlin, 2008.

[61] Frank Keller and Mirella Lapata. Using the Web to Obtain Frequencies for Unseen Bigrams. *Computational Linguistics*, 29(3):459–484, 2003.

[62] Marc Kupietz. Near-duplicate detection in the ids corpora of written german. Technical Report kt-2006-01, Institut für Deutsche Sprache, Mannheim, 2005.

[63] Marc Kupietz, Cyril Belica, Holger Keibel, and Andreas Witt. The German Reference Corpus DeReKo: A Primordial Sample for Linguistic Research. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 1848–1854, Valletta, Malta, 2010. European Language Resources Association (ELRA).

[64] Maciej Kurant, Minas Gjoka, Carter T Butts, and Athina Markopoulou. 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, 2011. ACM.

[65] Maciej Kurant, Athina Markopoulou, and Patrick Thiran. On the bias of BFS (Breadth First Search). In *International Teletraffic Congress (ITC 22)*, 2010.

[66] Yan Chi Lam. Managing the Google Web 1T 5-gram with relational database. *Journal of Education, Informatics, and Cybernetics*, 2(2), 2010.

[67] Thomas K Landauer, Danielle S McNamara, Simon Dennis, and Walter Kintsch, editors. *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates, Mahwah, 2007.

[68] Geoffrey Leech. New resources or just better old ones? the holy grail of representativeness. pages 133–149. 2007.

[69] Anke Lüdeling, Stefan Evert, and Marco Baroni. Using the web for linguistic purposes. pages 7–24.

[70] Jayant Madhavan, Loredana Afanasiev, Lyublena Antova, and Alon Halevy. Harnessing the Deep Web: Present and Future. In *4th Biennial Conference on Innovative Data Systems Research (CIDR)*, 2009.

[71] Arun S Maiya and Tanya Y Berger-Wolf. Benefits of bias: towards better characterization of network sampling. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 105–113, New York, NY, USA, 2011. ACM.

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

[73] Filippo Menczer, Gautam Pant, and Padmini Srinivasan. Topical web crawlers: Evaluating adaptive algorithms. *ACM Trans. Internet Technol.*, 4(4):378–419, 2004.

[74] Gordon Mohr, Michael Stack, Igor Ranitovic, Dan Avery, and Michele Kimpton. Introduction to Heritrix, an Archival Quality Web Crawler. In *Proceedings of the 4th International Web Archiving Workshop (IWAW'04)*, 2004.

[75] Christopher Olston and Marc Najork. *Web Crawling*, volume 4(3) of *Foundations and Trends in Information Retrieval*. now Publishers, Hanover, MA, 2010.

[76] Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. *English Gigaword Fifth Edition*. Linguistic Data Consortium, Philadelphia, USA, 2011.

[77] Jan Pomikálek, Miloš Jakubíček, and Pavel Rychlý. Building a 70 Billion Word Corpus of English from ClueWeb. In *Proceedings of LREC 08*, pages 502–506, Istanbul, 2012.

[78] Michael O. Rabin. Fingerprinting by random polynomials. Technical Report TR-CSE-03-01, Center for Research in Computing Technology, Harvard University, Harvard, 1981.

[79] Carlos Ramisch, Paulo Schreiner, Marco Idiart, and Aline Villavicencio. An evaluation of methods for the extraction of multiword expressions. In *Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 50–53, Marrakech, Morocco, 2008.

[80] Patrick Riehmann, Henning Gruendl, Bernd Froehlich, Martin Potthast, Martin Trenkmann, and Benno Stein. The Netspeak WordGraph: Visualizing keywords in context. In *Proceedings of the 4th IEEE Pacific Visualization Symposium (PacificVis '11)*, pages 123–130. IEEE, 2011.

[81] Paat Rusmevichientong, David M Pennock, Steve Lawrence, and C Lee Giles. Methods for sampling pages uniformly from the World Wide Web. In *In AAAI Fall Symposium on Using Uncertainty Within Computation*, pages 121–128, 2001.

[82] Mejdl S. Safran, Abdullah Althagafi, and Dunren Che. Improving Relevance Prediction for Focused Web Crawlers. In *IEEE/ACIS 11th International Conference on Computer and Information Science (ICIS), 2012*, pages 161–166, 2012.

[83] Andreas Salomonsson, Svetoslav Marinov, and Pierre Nugues. Identification of entities in swedish. In *Proceedings ot The Fourth Swedish Language Technology Conference*, pages 62–63, Lund, 2012.

[84] Roland Schäfer, Adrien Barbaresi, and Felix Bildhauer. The Good, the Bad, and the Hazy: Design Decisions in Web Corpus Construction. In *Proceedings of WAC8*, 2013.

[85] Roland Schäfer and Felix Bildhauer. Building Large Corpora from the Web Using a New Efficient Tool Chain. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 486–493, Istanbul, 2012. ELRA.

[86] Roland Schäfer and Felix Bildhauer. *Web Corpus Construction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool, San Francisco etc., 2013.

[87] M Ángeles Serrano, Ana Maguitman, Marián Boguñá, Santo Fortunato, and Alessandro Vespignani. Decoding the structure of the WWW: A comparative analysis of Web crawls. *ACM Trans. Web*, 1(2), 2007.

[88] Serge Sharoff. Creating general-purpose corpora using automated search engine queries. In *WaCky! Working papers on the Web as Corpus. Gedit*, 2006.

[89] Anil Kumar Singh and Harshit Surana. Can corpus based measures be used for comparative study of languages? In *Proceedings of Ninth Meeting of the ACL Special Interest Group in Computational Morphology and Phonology*, pages 40–47. Association for Computational Linguistics, 2007.

[90] Jae Jung Song. *Linguistic Typology: Morphology and Syntax*. Harlow, Longman, 2001.

[91] Miroslav Spousta, Michal Marek, and Pavel Pecina. Victor: The web-page cleaning tool. pages 12–17.

[92] Padmini Srinivasan, Filippo Menczer, and Gautam Pant. A General Evaluation Framework for Topical Crawlers. *Inf. Retr.*, 8(3):417–447, 2005.

[93] Vít Suchomel and Jan Pomikálek. Effcient Web Crawling for Large Text Corpora. In Adam Kilgarriff and Serge Sharoff, editors, *Proceedings of the seventh Web as Corpus Workshop*, pages 40–44, 2012.

[94] Morris Swadesh. Lexico-statistic dating of prehistoric ethnic contacts: with special reference to north american indians and eskimos. *Proceedings of the American philosophical society*, 96(4):452–463, 1952.

[95] Morris Swadesh. Towards greater accuracy in lexicostatistic dating. *International journal of American linguistics*, 21(2):121–137, 1955.

[96] George Kingsley Zipf. *Human Behavior and the Principle of Least-Effort*. Addison-Wesley, Cambridge, MA, 1949.