C4Corpus: Multilingual Web-size corpus with free license

Ivan Habernal[†‡], **Omnia Zayed**[‡], **Iryna Gurevych**[†‡]
† Ubiquitous Knowledge Processing Lab
‡ Research Training Group AIPHES
Department of Computer Science, Technische Universität Darmstadt, Hochschulstr. 10, 64289 Darmstadt, Germany
www.ukp.tu-darmstadt.de, www.aiphes.tu-darmstadt.de

Abstract

Large Web corpora containing full documents with permissive licenses are crucial for many NLP tasks. In this article we present the construction of 12 million-pages Web corpus (over 10 billion tokens) licensed under CreativeCommons license family in 50+ languages that has been extracted from CommonCrawl, the largest publicly available general Web crawl to date with about 2 billion crawled URLs. Our highly-scalable Hadoop-based framework is able to process the full CommonCrawl corpus on 2000+ CPU cluster on the Amazon Elastic Map/Reduce infrastructure. The processing pipeline includes license identification, state-of-the-art boilerplate removal, exact duplicate and near-duplicate document removal, and language detection. The construction of the corpus is highly configurable and fully reproducible, and we provide both the framework (*DKPro C4CorpusTools*) and the resulting data (*C4Corpus*) to the research community.

Keywords: CommonCrawl, Creative Commons, Web corpus, Amazon Web Services

1 Introduction

Availability of large-scale corpora is crucial for state-of-the-art Natural Language Processing (NLP). The importance of both annotated and raw large-scale corpora is rapidly increasing due to recent success of neural networks and similar semi- or unsupervised methods in a wide variety of language processing tasks. In recent years, tremendous progress has been made with sentence-level tasks (such as dependency parsing) and genre-specific benchmarks (such as work on the Penn Discourse Treebank). There is also an increasing demand for solutions scaling to heterogeneous document collections on the web. Current trends lean toward multilingual solutions, e.g., universal POS tags (Petrov et al., 2012), which requires heterogeneous corpora in multiple languages. Furthermore, recent document-level research tasks, such as multi-document summarization (Bing et al., 2015) or argumentation analysis (Habernal and Gurevych, 2015), heavily depend on document-level training and evaluation corpora.

One of the big obstacles for the current research is the lack of large-scale freely-licensed heterogeneous corpora in multiple languages, which can be re-distributed in the form of entire documents. Existing corpora are limited along several dimensions. First, they often exhibit monolingual nature, e.g., ClueWeb[1], Annotated English Gigaword[2] (Napoles et al., 2012), and several **WaC* corpora (Ljubešić and Klubička, 2014; Faaß and Eckart, 2013). Second, they are usually available as either n-grams (Brants and Franz, 2006) or shuffled sentences, e.g., COW (Schäfer and Bildhauer, 2012) or Leipzig Corpora (Goldhahn et al., 2012). Third, some corpora cover only a limited range of genres, e.g., discussions (Hládek et al., 2014), newswire (Spoustová and Spousta, 2012), or Wiki-texts (Lyding et al., 2014). Finally, due to the restrictive license of the content, many corpora cannot be re-distributed because of the risk of copyright infringement (Biemann et al., 2013; Schäfer, 2015).

To the best of our knowledge, no current approaches target at filling this gap. In order to scale up to the Internet size, such an approach would require state-of-the-art functional components as well as efficient execution on the corresponding computing infrastructure such as Amazon Elastic MapReduce (EMR). In this paper, we propose a solution to this hard problem. Our approach yields large-scale heterogeneous corpora in multiple languages freely re-distributable at the document level as the major product of our research.

For this purpose, we build upon the CommonCrawl[3] project, the largest multilingual web crawl available to date. We employ state-of-the-art components for Web corpus processing and bring them under the unified framework based on Hadoop platform in order to scale up to 1.8 billion URLs present in the recent CommonCrawl data. Despite many existing works focusing on Web corpus construction (described in the next section), our approach aims at several novel aspects. First, we guarantee full reproducibility of our approach, as both CommonCrawl and our framework are freely accessible. Second, the resulting corpora are also available to the public which, we hope, will fulfill the needs for large textual datasets (in a particular language and with a specific license) and allow various research questions to scale-up without the burden of obtaining the data directly from the Web. Third, our use-case goes beyond sampling unique sentences or n-grams, but rather focuses on entire documents. Our project is entitled *C4Corpus*, an abbreviation of *Creative Commons from Common Crawl Corpus* and is hosted under the *DKPro* umbrella[4] at `https://github.com/dkpro/dkpro-c4corpus` under ASL 2.0 license.

[1] `http://www.lemurproject.org/clueweb12/`

[2] `https://catalog.ldc.upenn.edu/LDC2012T21`

[3] `http://commoncrawl.org/`

[4] DKPro is a community of projects focusing on re-usable NLP software. `http://www.dkpro.org/`

2 Related Work

In the related work section, we will discuss the most relevant research in terms of similar requirements as well as related work for particular components of processing pipeline for creating Web corpora.

Lyding et al. (2014) crawled 388k pages (270k from Wikimedia Foundation) and created a Creative Commons (CC) licensed corpus in Italian containing 250M tokens automatically annotated with lemma, POS and syntactic dependency. The corpus is currently available for download from the author's server.

Barbaresi and Würzner (2014) crawled 160k blogs from the German version of `wordpress.com`. They filtered pages under CC by looking for a presence of links to Creative Commons websites and reported 0.65 accuracy on 2.5k automatically classified blogs. The corpus is available upon request.

Spoustová and Spousta (2012) manually selected 40 Czech webs and hand-crafted scraping scripts for extracting the textual content resulting in a corpus with 2.6B tokens in three categories (articles, discussions, blogs). Language detection was based on manually crafted word lists, duplicates were removed on the paragraph level. Neither the corpus nor the tools are available anymore.

Versley and Panchenko (2012) crawled the Web with the focus on the German sites in two categories: news-style content and general Web content. Their pipeline included heuristic language detection, boilerplate removal, standard near-duplicate detection and several linguistic annotation steps (morphology and parsing). The paper gives no information regarding the availability of the compiled corpus neither about the content copyright.

Biemann et al. (2013) focused on research questions within the sentence level (distributional semantics and similar). Their framework consists of several Hadoop jobs for different parts of preprocessing (e.g., boilerplate using `html2text` tool, de-duplication of content from the same host, linguistic annotation) and is available as open source.

Schäfer (2015) developed an open-source platform `tetex` that covers all steps in Web corpora construction (language detection, boileplate removal, sentence extraction and deduplication). The throughput of this system is 100M pages in 4 days (12 cores) or 4 hours on a HPC cluster. The resulting output is a set of non-duplicate sentences.

Baroni et al. (2009) introduced the WaCky project which offers three large linguistically processed corpora of English, German and Italian. The authors followed the full pipe-line of creating large web corpus starting with web crawling, then post-crawl cleaning and finally basic linguistic annotation. The post-crawl cleaning step includes boilerplate removal and de-duplication. The linguistic annotation includes tokenization, part-of-speech tagging and lemmatization. The tools and the tagged corpora are available on-line for academic purposes.

Ljubešić and Klubička (2014) based their work on existing tools from Suchomel and Pomikálek (2012) (crawler and boilerplate removal) with focus on Bosnian, Croatian, and Serbian. The corpus contains $\approx$ 1B tokens annotated with the lemma, morphology and syntax layers and is available upon request.

Relevant research that exploits CommonCrawl includes mining parallel texts for machine translation by Smith et al. (2013) or extracting n-grams and building language models by Buck et al. (2014). While these works tackle the issue of extracting data from CommonCrawl, they are very task-specific and do not deal with creating general Web corpus (such as boilerplate removal, de-duplication on the document level, license detection, etc.).

3 Corpora

This section introduces the corpora employed to test our processing pipeline and evaluate the performance of individual components. We experimented with two different corpora to assess the performance of the proposed pipeline (Section 4) on new data as well as report some findings on established corpora.

3.1 CommonCrawl subset

The CommonCrawl data set is a huge internet crawl that has been collected over the last 7 years. Recently, CommonCrawl has been fetching its content every three months. This time dimension is a unique feature of CommonCrawl, compared to i.e. ClueWeb12 which is a one-time snapshot of the Web. As of 2015, the web archive contains about 149 TB of uncompressed data from $\approx$ 1.9 billion web-pages[5] and is hosted on Amazon S3. We downloaded a random subset of the corpus (about 460 million web pages) to enable local processing and conduct preliminary experiments. Experimenting on the whole corpus will be discussed in Section 6.

3.2 Own Crawl

We also performed our own Creative Commons (CC) focused crawling. As a seed set, we used a link graph provided by the Web Data Commons initiative[6] and extracted all pages that pointed to the CC license sites. From this seed of particular pages that are likely to be under CC, we extracted a smaller set of *domains* and exhaustively crawled them using the Nutch crawler running on 100 servers for about 2 weeks in 2015. This resulted into around 100 million crawled pages likely to be under CC for further experiments. We excluded Wikipedia from this crawl, as it can be downloaded directly as a database dump.

4 Processing Pipeline – DKPro C4CorpusTools framework

The proposed pipeline consists of four main components to process an existing web crawl. One of the main design goals of this pipeline is to enable large-scale processing and good reliability, thus, for each component an appropriate tool is adapted or developed in Java then extended to Hadoop MapReduce jobs. The proposed pipeline currently provides license detection, boilerplate removal, language identification, and near-duplicate content removal. Each of these components is described in detail in the following sub-sections. Figure 1 illustrates the MapReduce workflow of the whole pipeline.

[5] `http://blog.commoncrawl.org/2015/10/august-2015-crawl-archive-available/`

[6] `http://webdatacommons.org/`

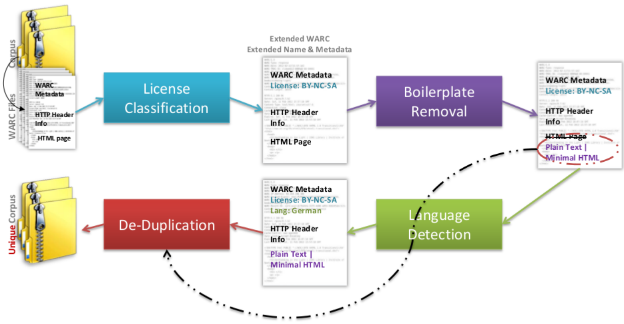

Figure 1: C4Corpus MapReduce workflow

By contrast to other frameworks, our approach builds upon the scalable Map/Reduce paradigm (whereas, for instance, `tetex` (Schäfer, 2015) runs on a HPC cluster) and focuses solely on processing entire documents (as opposed to, e.g., Biemann et al. (2013)).

4.1 License Detection

Copyright is considered one of the major concerns while building a web corpus. Copyrighted content impedes researchers to use or redistribute the full texts within a large corpus which in turn hinders the progress of many NLP applications such as Text Summarization and Argumentation mining. As mentioned earlier in Section 2, Lyding et al. (2014) and Barbaresi and Würzner (2014) investigated this issue by manually classifying the licensed content within their corpora.

Creative Commons (CC) introduces 7 different types of licenses[7], as described in Table 1, which allow users to grant copyright permissions to their work on-line. One of the goals of this work is to identify the licensed content of an existing web corpus. To achieve this goal, we implemented an algorithm based on regular expressions to scan a single HTML page for the license link pattern, which is then used for classifying the page into one of these 7 CC license categories (or *none* if no CC license is detected).

Acronym	**Rights**
CC0	Public domain
BY	Attribution alone
BY-NC	Attribution + Noncommercial
BY-SA	Attribution + ShareAlike
BY-ND	Attribution + NoDerivatives
BY-NC-SA	Attribution + Noncommercial + ShareAlike
BY-NC-ND	Attribution + Noncommercial + NoDerivatives

Table 1: Creative Commons License Types

In order to evaluate this component, 100 pages in English from our crawl, described previously in sub-section 3.2, were manually annotated. Figure 2 shows the types of licenses present in these 100 pages along with their distribution. For evaluation purposes, we cast this task as binary classification (*CC-family* license versus *none*). Table 2 shows the obtained results in terms of precision, recall and F-score for the *CC-family* class.

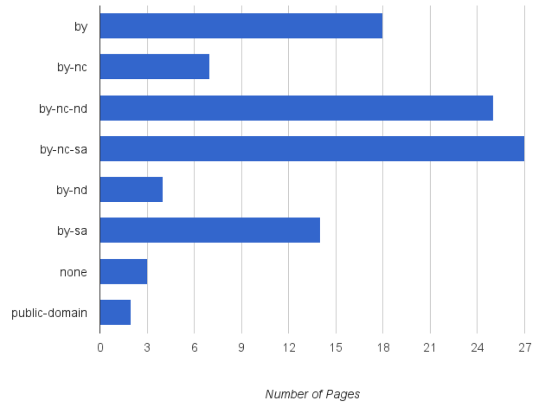

Figure 2: Distribution of license types among the 100 pages of the test set

P	R	F_1
97.97%	100%	98.97%

Table 2: Evaluation of the License Detection component.

Three different sources of false-positives can be identified. One example of a false-positive is that the page is not licensed even though it contains a CC-license link. Another source of false-positives is when multiple CC-license links are found in a single HTML page. Examples of these are: a blog page contains many photos and each photo is licensed under different CC-license type or a blog home page with many articles and each article is licensed under different CC-license type. In the latter case, since the license of the actual blog is not specified explicitly, we introduced a new tag for such cases, namely, ”CC-Undetermined”. Section 5 reports statistics of our crawl and the CommonCrawl subset which include corpus splits according to the license type.

4.2 Boilerplate Removal

Boilerplate removal is an essential step in building web corpora. In this step, our goal is to clean up a web page by removing the uninformative content which has no usage in text understanding such as navigation bars, advertisements, header, footer, etc. We re-implemented the state-of-the-art python algorithm JusText (Pomikálek, 2011) in Java. The algorithm uses heuristics to classify the textual blocks in a given HTML page in one of the four classes, namely

- *bad*, which considers boilerplate blocks
- *good*, which is the main content blocks
- *short*, which is a too short content block, thus a reliable decision cannot be made

[7]`https://creativecommons.org/licenses/`

- *near-good*, which is a content block that lies between a short and a good one. See (Pomikálek, 2011) for a detailed description.

The classification criteria make use of a set of textual features extracted from the HTML page such as the link density, text density, and others (Pomikálek, 2011). After removing the boilerplates, our algorithm can be parametrized to output plain text (by default) or to produce a minimal HTML, where the retaining text parts are printed along with their original HTML tags (such as `<p>`, `<h1>`, etc.). The minimal HTML option allows the user to render the plain text with simple markups and keep some minimal HTML semantics of the output.

Evaluation of this component is performed using the benchmark CleanEval dataset (Baroni et al., 2008) as well as the cleaneval.py script created by Evert (2008) in order to be able to compare our Java implementation to the original JustText Python implementation by Pomikálek (2011). We ran both JusText and our Java re-implementation on the CleanEval Test set which consists of 681 web pages. As shown in Table 3, the obtained results are comparable to (Pomikálek, 2011). The results, after running the CleanEval script, are given in terms of macro-averaged precision, recall and F-score.

	Our Java re-implementation	Pomikálek (2011) Python implementation
P	94.37%	95.83%
R	81.15%	82.91%
F_1	84.36%	85.70%

Table 3: Evaluation of the Boilerplate Removal component on the CleanEval test set

Although the boilerplate removal phase is always destructive, we allow users to track back to the original HTML by keeping the location and ID of the original file in HDFS/AWS S3. This might be useful if a particular task requires access to the HTML structure or other HTML-specific information even after the boilerplate removal phase.

4.3 Language Identification

The next step in our pipeline is to identify the language of the web pages in the corpus. We rely on an existing Java library[8] which is able to detect over 50 languages by employing character n-grams as features to train a Naïve Bayes classifier. Section 5 shows the most common languages used in our crawl and the CommonCrawl subset.

4.4 Duplicate and Near-Duplicate Content Removal

De-duplication is one of the essential cleaning steps while building a web corpus. We implemented a greedy algorithm by employing the state-of-the-art SimHash algorithm introduced by Charikar (2002) and the bitwise hamming distance technique. We follow a similar approach to the one introduced in (Manku et al., 2007).

[8] `http://code.google.com/p/language-detection/`

Our proposed algorithm is composed of three steps, to remove duplicate and near-duplicate documents from a web-scale corpus, as follows:

1. Cluster possible near-duplicate candidates using the SimHash algorithm.
2. Create pairs of near-duplicate documents by using hamming distance.
3. Delete the shortest document from each pair using a greedy algorithm.

Figure 3 describes an example of the workflow between the three steps. The goal of the first step, which is computed using MapReduce, is to group together possible candidates of near-duplicate documents for further similarity checking. This step starts with representing a web page as a set of character n-grams shingles; then each shingle is hashed into 64-bit hash value. After that, SimHash is utilized to compress these hash values into a single 64-bit binary fingerprint. Each fingerprint is split into bands to build a characteristic matrix for the whole corpus. Documents that have the same bands are grouped together.

The second step is performed locally to calculate the hamming distance between each pair of the near-duplicate candidates. This step is divided into two phases. In the first phase, each set of similar documents, which output from step 1, is converted to tuples. The hamming distance between each tuple is calculated. Based on the hamming distance threshold, near-duplicate pairs are kept for further processing in the second phase. The second phase employs a greedy algorithm in order to select the final set of unique documents. The algorithms make use of two constraints which are: 1) get as many unique documents as possible without redundancies and 2) keep the longest document in order not to lose information.

Near-duplicate removal using hamming distance between documents pairs (two documents with a certain distance are considered equal) and selecting always the longer document from the pair is equivalent to hamming clustering which is a NP-hard optimization problem (Gasieniec et al., 2004). Our greedy algorithm yields a reasonable solution, as the local clusters or duplicate candidates are processed in parallel in MapReduce.

The final step is done using MapReduce to delete redundant documents from the corpus. In this step, the original corpus in addition to the list of documents from step 2 is used to create the unique (non-duplicated) final corpus.

5 Results on small-scale corpora

This section summarizes our experimental results and preliminary findings. For testing our pipeline we used two in-house Web corpora (Section 3) and a private Hadoop cluster with 254 CPUs.

The distribution of pages licensed under Creative Commons is shown in tables 4 and 5. In case of our CC-focused crawl, the number of CC pages is rather high, yielding about 200k pages (90M tokens) for English. The CommonCrawl subset shows similar distribution of languages, but in average only 9% (± 6 pp) are recognized as CC-licensed.

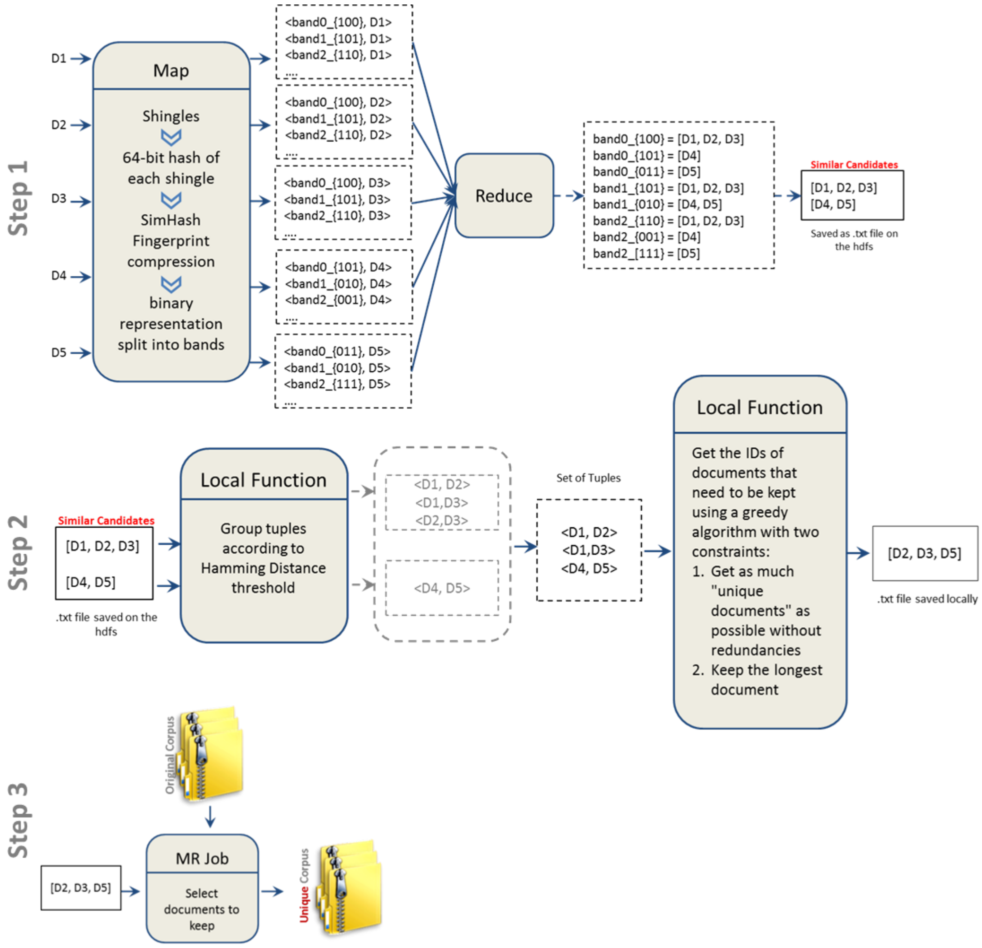

Figure 3: De-duplication workflow example

However, given that we analyzed only a fraction of the entire CommonCrawl corpus, the final corpus size will be presented in section 6.

Regarding the duplicate detection, we found that 32 million pages in our crawl (out of 100 million) were exact duplicates. This is not surprising, because the crawler went "deep" in the seed domains only. The number of exact matches in CommonCrawl subset was much lower (about 1%).

5.1 Properties of 'clean' corpora

To get some insights into the content of the resulting corpora, we compare our Web corpora to the Brown corpus (as a linguistically "clean" and balanced corpus) and to the entire English Wikipedia corpus (as the largest CC-licensed corpus). We report Spearman's rank correlation of top N words (100, 1000, and 10,000) following the methodology from (Schäfer and Bildhauer, 2013). Table 6 shows that the rank of top N words from the Brown corpus correlates with other four corpora with a little difference between Wikipedia and our corpora. On the other hand, the correlation of top ranked N words from our corpora with the Brown corpus and Wikipedia is rather low. We examined the results more in detail and found that our web corpora still contain many frequent non-linguistic tokens (html and http-related tokens). This might be partly due to insufficient preprocessing and we plan to investigate this issue deeper. There is room for further testing the resulting corpora in terms of their suitability for various tasks and their "comparability", which is out of scope of the current paper. These can follow methodologies proposed by Lijffijt et al. (2014) or Sharoff (2013).

6 Scaling up to full CommonCrawl

We ran the entire pipeline on the full CommonCrawl (November 2015 crawl), which consists of 35,700 `warc.gz` files (total size 32.59 TB) and contains 2,052,525,490 crawled records. The crawled content is a mixture of various types determined by the HTTP content header, but the majority (>99%) are actual HTML pages (either `text/html` or `application/xhtml+xml`). Table 7 reports several statistics of the final corpus after boilerplate removal and de-duplication. The final CC-licensed corpus is publicly available in our Amazon S3 bucket `s3://ukp-research-data/c4corpus/cc-final-2015-11/`.[9] Note that we do not guarantee that the license is correctly detected; it should be always checked with the original HTML file. The corpus has about

[9]See the user's guide at `https://github.com/dkpro/dkpro-c4corpus/` for a detailed explanation how to access the data.

	BY	BY-NC	BY-NC-ND	BY-NC-SA	BY-ND	BY-SA	CC-unsp.	CC-0	Total CC	None	Tokens CC
en	19 195	4 036	13 911	18 243	1 550	101 203	1 469	658	160 265	13 895 925	79 493 716
es	679	249	857	838	58	2 522	256	6	5 465	77 421	6 002 529
unk	192	39	189	123	15	4 337	41	5	4 941	609 480	1 426 947
bn	154	36	144	166	20	3 161	7	2	3 690	101 702	2 832 013
fr	155	88	172	142	10	1 873	18	1	2 459	61 817	2 582 482
it	119	27	276	199	16	1 385	26	1	2 049	17 740	2 069 067
pt	248	259	156	95	27	766	7	0	1 558	22 224	2 210 233
nl	19	3	27	215	2	1 240	4	0	1 510	11 019	807 119
id	744	8	11	15	2	603	1	4	1 388	17 138	1 262 530
de	150	22	316	143	23	367	21	0	1 042	53 650	832 240

Table 4: Number of documents under CC-licenses for top 10 languages identified in our CommonCrawl subset and the number of pages without free license (the *none* column); the last column shows the total number of tokens in the Creative-Commons licensed pages.

	BY	BY-NC	BY-NC-ND	BY-NC-SA	BY-ND	BY-SA	CC-unsp.	CC-0	Total CC	None	Tokens CC
en	32 084	1 805	30 010	13 544	96 948	32 558	1 623	978	209 550	3 735	91 210 644
bn	2 291	2 312	113 115	5 674	295	938	10 618	0	135 243	156	46 769 924
es	1 624	1 622	8 436	9 021	41	653	22	0	21 419	80	10 524 783
fr	1 454	785	7 999	4 299	251	365	52	0	15 205	1 140	10 703 237
pt	52	2	14 043	173	1	7	0	0	14 278	0	7 675 435
it	44	34	7 790	439	11	167	0	0	8 485	4	1 824 652
de	1 456	1 678	668	1 462	46	893	70	0	6 273	28	2 481 760
cs	0	0	765	915	0	251	0	0	1 931	0	945 958
unk	290	54	828	343	30	97	41	3	1 686	73	448 556
nl	181	1	4	8	62	155	1	0	412	3 918	185 660

Table 5: Number of documents under CC-licenses for top 10 languages identified in our own crawl and the number of pages without free license (the *none* column); the last column shows the total number of tokens in the Creative-Commons licensed pages.

”Gold” corpus	Top N	Brown	Wiki	Our crawl CC	Common-Crawl CC	Common-Crawl no-CC
Brown	10^2		0.76	0.81	0.82	0.83
	10^3		0.58	0.71	0.69	0.68
	10^4		0.70	0.72	0.72	0.71
Wikipedia	10^2	0.84		0.79	0.76	0.77
	10^3	0.53		0.61	0.55	0.58
	10^4	0.61		0.71	0.68	0.69
Our crawl CC	10^2	0.78	0.61		0.83	0.76
	10^3	0.47	0.46		0.73	0.67
	10^4	0.54	0.57		0.75	0.71
Common Crawl CC	10^2	0.44	0.44	0.53		0.84
	10^3	0.23	0.28	0.48		0.77
	10^4	0.52	0.57	0.78		0.78
Common Crawl no-CC	10^2	0.26	0.29	0.36	0.69	
	10^3	0.25	0.30	0.47	0.78	
	10^4	0.49	0.57	0.73	0.78	

Table 6: Spearman’s rank correlations between top N words from a pair of corpora (*CommonCrawl* denotes the subset of CommonCrawl as introduced in Section 3). The top N words were drawn from the corpus in the ”Gold” corpus column. All values are statistically significant ($p < 0.01$).

29 GB (gzipped) and contains more than 12 million pages (10.8 billion tokens) in 53 languages.

Table 8 lists top 20 domains from the English sub-corpus. According to these top-domain names, the corpus contains a mixture of Q/A sites, blogs, discussion forums, database-like sites, and wikis. A deeper investigation of the corpus properties with respect to explicit Web genres is planned as future work.

Rank	Top domain	Pages
1	`stackexchange.com`	902 115
2	`blogspot.com`	502 962
3	`stackoverflow.com`	387 553
4	`bookrags.com`	224 968
5	`travelpod.com`	173 477
6	`marinespecies.org`	130 309
7	`wikia.com`	128 129
8	`wordpress.com`	121 261
9	`familysearch.org`	118 593
10	`superuser.com`	100 944
11	`serverfault.com`	97 454
12	`wikitravel.org`	88 162
13	`uniprot.org`	85 606
14	`askubuntu.com`	81 716
15	`hindawi.com`	81 647
16	`wikipedia.org`	72 522
17	`destructoid.com`	71 387
18	`owasp.org`	66 637
19	`msdn.com`	59 429
20	`androidcentral.com`	58 054

Table 8: Top 20 top domains in the English sub-part of the CC-licensed C4Corpus.

6.1 Discussion

6.1.1 Technical aspects

We used the Amazon Elastic Map/Reduce (EMR) infrastructure for processing the full CommonCrawl. Since our

	BY	BY-NC	BY-NC-ND	BY-NC-SA	BY-ND	BY-SA	CC-unsp.	CC-0	Total	Tokens
en	1 606 052	314 139	1 163 214	1 171 304	197 768	3 078 922	112 385	30 643	7 674 427	7 733 601 646
es	106 047	40 343	133 336	125 092	11 735	413 058	36 802	1 248	867 661	815 155 576
fr	27 279	6 318	25 455	22 204	1 626	353 438	1 624	235	438 179	366 308 592
it	20 125	5 715	43 677	34 108	2 581	293 308	4 483	213	404 210	303 947 215
pt	45 028	46 953	30 825	18 604	4 791	174 996	1 597	37	322 831	355 029 035
id	144 200	2 124	3 365	3 200	370	120 029	2 559	177	276 024	200 776 031
nl	3 175	1 110	4 074	11 011	657	217 604	590	11	238 232	99 831 013
unk	18 459	3 429	15 444	8 385	965	182 364	4 938	303	234 287	91 002 113
sv	4 859	313	891	3 683	337	158 969	5 542	18	174 612	65 590 449

Table 7: Number of documents under CC-licenses for top 10 languages identified in the full CommonCrawl.

framework is developed on top of Hadoop Map/Reduce (version 2.6), it can be directly deployed at EMR without any modifications. However, scaling up to 32 TB (34k mappers) on 2000+ core cluster brings several unexpected technical challenges.

For most of the steps, we launched a cluster made of 2-3 master and 16-64 spot instances of `c4.8xlarge` nodes (32 CPUs each). The advantage of spot instances is their lower price as compared to the reserved ones, but it comes with the risk of losing them when the bid price is overbidden by other EMR customers. It turned out that loosing nodes during the boilerplate removal phase (phase 1) had detrimental effects and the job usually could not recover. As the prices of spot instances vary with respect to AWS region and day of the week (companies use spot instances for their weekly batches), configuring and launching the cluster with spot instances to successfully complete the job is rather tricky.[10] Furthermore, tuning the performance of Map/Reduce jobs requires experimenting with several dozens of parameters, as discussed in (White, 2015, p. 201).

6.1.2 CommonCrawl data

One critical question when creating a CC-licensed Web corpus is: Should we rather make our own CC-focused crawl instead of relying on CommonCrawl?

CommonCrawl has several potential drawbacks. First, the crawl has been performed on a fixed set of URLs.[11] One implication is that no new sites are discovered, on the other hand one can explore the evolution of the present sites in time; this depends on the application requirements. Second, the majority of crawled pages are in English. On one hand, this reflects the language distribution on the Web in general; on the other hand the size of non-English sites can be a limiting factor for some applications.

Performing own focused-crawl on CC-sites is feasible (see results in section 5) but has also several disadvantages. Despite obvious technical challenges of Web-size crawling (see for example (Boldi et al., 2016) for a state-of-the-art crawler description), the reproducibility of the downstream results is not ensured. Usually, institutions perform their private crawl and publish only the results (i.e., sentences and vocabularies, pre-trained word embeddings, annotated documents). The raw crawls are never made public, usually because of legal issues or simply because of technical difficulties due to their extreme size. By contrast to CommonCrawl, applications that leverage the private crawls thus depend on a proprietary crawl snapshot and cannot be reproduced by other researchers.[12] We believe that focused-crawl of CC sites can be a preferable solution as long as the raw crawls are available to the public like in the case of CommonCrawl.

7 Conclusions

In this paper, we proposed a solution to the problem of re-distributing and re-using the full text of large web corpora due to the copyright restriction. A framework is introduced to process a large-scale multilingual Web-based corpus which incorporates state-of-the-art components for license detection, language identification, boilerplate removal and documents de-duplication. The framework is designed to support efficient execution and scalability by employing distributed resources such as Hadoop and Amazon Elastic MapReduce (EMR). Experiments are done to analyze the efficiency of the framework. Our results indicate that it is possible to create a large corpus with free content for multiple languages, which is the ultimate goal to boost full reproducibility and enable unrestricted data sharing within the NLP community. We provide the *DKPro C4CorpusTools* framework under ASL license at `github.com/dkpro/dkpro-c4corpus`. The resulting corpora are publicly available at Amazon S3 in the `s3://ukp-research-data/c4corpus/` bucket.

Acknowledgements

This work was supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant № I/82806, by the German Institute for Educational Research (DIPF), by the German Research Foundation (DFG) via the German-Israeli Project Cooperation (DIP, grant DA 1600/1-1), the GRK 1994 AIPHES (DFG), and by Amazon Web Services in Education Grant award.

[10] But it still pays off— the reserved price for a single `c4.8xlarge` instance is $1.68 per hour while the average price for a spot instance is about $0.71, which makes about $750 difference for 24 hours of computing on a 32-nodes cluster.

[11] `https://goo.gl/ol150K`

[12] Even if a list of crawled URLs is provided, one cannot fully restore the original corpus; this is for example the case of the recent Leeds Web genre corpus (Asheghi et al., 2016) where we could only retrieve about 3/4 of the URLs as the rest has since disappeared from the internet.

8 References

Asheghi, N. R., Sharoff, S., and Markert, K. (2016). Crowdsourcing for web genre annotation. *Language Resources and Evaluation*, in press, jan.

Barbaresi, A. and Würzner, K.-M. (2014). For a fistful of blogs: Discovery and comparative benchmarking of republishable German content. In Faaß, G. and Ruppenhofer, J., editors, *Workshop Proceedings of the 12th Edition of the KONVENS Conference*, pages 2–10, Hildesheim, Germany. Universität Hildesheim.

Baroni, M., Chantree, F., Kilgarriff, A., and Sharoff, S. (2008). Cleaneval: a Competition for Cleaning Web Pages. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., and Tapias, D., editors, *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may. European Language Resources Association (ELRA).

Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Biemann, C., Bildhauer, F., Evert, S., Goldhahn, D., Quasthoff, U., Schäfer, R., Simon, J., Swiezinski, L., and Zesch, T. (2013). Scalable construction of high-quality web corpora. *Journal for Language Technology and Computational Linguistics*, 28(2):23–59.

Bing, L., Li, P., Liao, Y., Lam, W., Guo, W., and Passonneau, R. (2015). Abstractive multi-document summarization via phrase selection and merging. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1587–1597, Beijing, China, July. Association for Computational Linguistics.

Boldi, P., Marino, A., Santini, M., and Vigna, S. (2016). BUbiNG: Massive Crawling for the Masses. *arXiv preprint*. http://arxiv.org/abs/1601.06919.

Brants, T. and Franz, A. (2006). Web 1T 5-gram Version 1. Linguistic Data Consortium.

Buck, C., Heafield, K., and van Ooyen, B. (2014). N-gram Counts and Language Models from the Common Crawl. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3579–3584, Reykjavik, Iceland. European Language Resources Association (ELRA).

Charikar, M. S. (2002). Similarity estimation techniques from rounding algorithms. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 380–388, New York, NY, USA. ACM.

Evert, S. (2008). A lightweight and efficient tool for cleaning web pages. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., and Tapias, D., editors, *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Faaß, G. and Eckart, K. (2013). Sdewac – a corpus of parsable sentences from the web. In Gurevych, I., Biemann, C., and Zesch, T., editors, *Proceedings of 25th GSCL International Conference*, pages 61–68, Darmstadt, GE. Springer Berlin / Heidelberg.

Gasieniec, L., Jansson, J., and Lingas, A. (2004). Approximation algorithms for hamming clustering problems. *Journal of Discrete Algorithms*, 2(2):289 – 301. Combinatiorial Pattern Matching.

Goldhahn, D., Eckart, T., and Quasthoff, U. (2012). Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In Calzolari, N., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 759–765, Istanbul, Turkey, May. European Language Resources Association (ELRA).

Habernal, I. and Gurevych, I. (2015). Exploiting Debate Portals for Semi-Supervised Argumentation Mining in User-Generated Web Discourse. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2127–2137, Lisbon, Portugal. Association for Computational Linguistics.

Hládek, D., Staš, J., and Juhár, J. (2014). Slovak Web Discussion Corpus. In Przepiórkowski, A. and Ogrodniczuk, M., editors, *9th International Conference on NLP, PolTAL 2014*, pages 463–469, Warsaw, PL. Springer International Publishing.

Lijffijt, J., Nevalainen, T., Säily, T., Papapetrou, P., Puolamäki, K., and Mannila, H. (2014). Significance testing of word frequencies in corpora. *Digital Scholarship in the Humanities*, pages 1–24.

Ljubešić, N. and Klubička, F. (2014). {bs,hr,sr}WaC – Web corpora of Bosnian, Croatian and Serbian. In Bildhauer, F. and Schäfer, R., editors, *Proceedings of the 9th Web as Corpus Workshop (WAC9)@EACL 2014*, pages 29–35, Gothenburg, Sweden. Association for Computational Linguistics.

Lyding, V., Stemle, E., Borghetti, C., Brunello, M., Castagnoli, S., Dell'Orletta, F., Dittmann, H., Lenci, A., and Pirrelli, V. (2014). The PAISÀ Corpus of Italian Web Texts. In Bildhauer, F. and Schäfer, R., editors, *Proceedings of the 9th Web as Corpus Workshop (WAC9) @ EACL 2014*, pages 36–43, Gothenburg, Sweden. Association for Computational Linguistics.

Manku, G. S., Jain, A., and Das Sarma, A. (2007). Detecting near-duplicates for web crawling. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 141–150, New York, NY, USA. ACM.

Napoles, C., Gormley, M., and Van Durme, B. (2012). Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX '12)*, pages 95–100, Montreal, Canada. Association for Computational Linguistics.

Petrov, S., Das, D., and McDonald, R. (2012). A universal part-of-speech tagset. In Calzolari, N., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2089–2096, Istanbul, Turkey, May. European Language Resources Association (ELRA).

Pomikálek, J. (2011). *Removing boilerplate and duplicate content from web corpora*. Ph.D. thesis, Masaryk university, Faculty of informatics, Brno, Czech Republic.

Schäfer, R. and Bildhauer, F. (2012). Building large corpora from the web using a new efficient tool chain. In Calzolari, N., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 486–493, Istanbul, Turkey, May. European Language Resources Association (ELRA).

Schäfer, R. and Bildhauer, F. (2013). *Web Corpus Construction*. Morgan & Claypool Publishers.

Schäfer, R. (2015). Processing and querying large web corpora with the COW14 architecture. In Bański, P., Biber, H., Breiteneder, E., Kupietz, M., Lüngen, H., and Witt, A., editors, *Proceedings of the 3 rd Workshop on Challenges in the Management of Large Corpora (CMLC-3)*, pages 28–34, Lancaster, UK, July.

Sharoff, S. (2013). Measuring the Distance Between Comparable Corpora Between Languages. In Sharoff, S., Rapp, R., Zweigenbaum, P., and Fung, P., editors, *Building and Using Comparable Corpora*, pages 113–130. Springer Berlin Heidelberg, Berlin, Heidelberg.

Smith, J. R., Saint-Amand, H., Plamada, M., Koehn, P., Callison-Burch, C., and Lopez, A. (2013). Dirt Cheap Web-Scale Parallel Text from the Common Crawl. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1374–1383, Sofia, Bulgaria. Association for Computational Linguistics.

Spoustová, J. and Spousta, M. (2012). A High-Quality Web Corpus of Czech. In Calzolari, N., Choukri, K., Declerck, T., Dogan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 311–315, Istanbul, Turkey. European Language Resources Association (ELRA).

Suchomel, V. and Pomikálek, J. (2012). Efficient web crawling for large text corpora. In Sharoff, S. and Kilgarriff, A., editors, *Proceedings of the seventh Web as Corpus Workshop (WAC7)*, pages 39–43, Lyon.

Versley, Y. and Panchenko, Y. (2012). Not Just Bigger: Towards Better-Quality Web Corpora. In *Proceedings of the seventh Web as Corpus Workshop (WAC7) @WWW 2012*, pages 44–52, Lyon. Association for Computational Linguistics.

White, T. (2015). *Hadoop: The Definitive Guide*. O'Reilly Media, Inc., Sebastopol, CA, 4th edition.