CommonCOW: Massively Huge Web Corpora from CommonCrawl Data and a Method to Distribute them Freely under Restrictive EU Copyright Laws

Roland Schäfer
German Grammar
Freie Universität Berlin
Habelschwerdter Allee 45, 14195 Berlin
roland.schaefer@fu-berlin.de

Abstract

In this paper, I describe a method of creating massively huge web corpora from the CommonCrawl data sets and redistributing the resulting annotations in a stand-off format. Current EU (and especially German) copyright legislation categorically forbids the redistribution of downloaded material without express prior permission by the authors. Therefore, stand-off annotations or other derivates are the only format in which European researchers (like myself) are allowed to re-distribute the respective corpora. In order to make the full corpora available to the public despite such restrictions, the stand-off format presented here allows anybody to locally reconstruct the full corpora with the least possible computational effort. In Section 1., I briefly introduce the technology behind the COW project (Corpora from the Web), which is used to create the CommonCrawl-derived corpora. In Section 2., I provide some details about the resulting CommonCOW (COCO) web corpora. Finally, in Section 3., I introduce a method to circumvent restrictive EU copyright laws by distributing only the corpus annotations (under a CC-BY license) together with a tool that allows users to locally reconstruct the corpus from the annotations and the original CommonCrawl files.

Keywords: web corpora, stand-off annotation, legal issues

1. Web corpora and COW

Over the past decade, web corpora have been created by various research groups (Biemann et al., 2007; Baroni et al., 2009; Pomikalek et al., 2009; Schäfer and Bildhauer, 2012; Schäfer and Bildhauer, 2013), and they have been actively used in Computational Linguistics and Corpus Linguistics. The COCO project introduced in this paper uses the existing technology developed by the COW (Corpora from the Web) project at Freie Universität Berlin (Schäfer and Bildhauer, 2012; Schäfer et al., 2013) in the COW14 version (Schäfer, 2015b).[1]

The backbone of this technology is the *texrex* web page cleaning tool.[2] Using a bundle of computational methods described in previous publications, *texrex* (in its 2015 version *texrex-behindthecow*):

1. filters out perfect and near-duplicate documents (Schäfer and Bildhauer, 2012),
2. strips HTML markup and scripts,
3. converts encodings (to UTF-8) and performs NFC Unicode normalization,
4. detects and annotates the document language,
5. classifies paragraphs as boilerplate or good text (Schäfer, 2015a),
6. classifies documents as containing more or less coherent text (Schäfer et al., 2013),
7. extracts meta information from the crawl headers and the HMTL source,
8. performs IP-based server geolocation.

Linguistic annotation (for Dutch, English, French, German, Spanish, Swedish) is done using available tools. For example, English COW corpora are tokenized using Ucto (van Gompel et al., 2012), part-of-speech tagged and chunked using TreeTagger (Schmid, 1994), and dependency parsed with the Malt Parser (Nivre et al., 2007). German is tokenized, tagged, and chunked using the same tools, named entity recognition is performed using the Stanford NER tool and available German models (Faruqui and Padó, 2010), and morphological analysis is performed using Mate tools (Björkelund et al., 2010). The data are processed on the high performance cluster of Freie Universität Berlin, which is based on SLURM.[3] This is arguably much easier than using Map-Reduce frameworks like Hadoop, and it allows us to create corpora the size of 10 or more billion tokens in weeks' time.[4] More details about the annotations can be found in (Schäfer, 2015b).

2. CommonCOW

So far, the COW project has been based entirely on self-crawled data (Schäfer and Bildhauer, 2012; Schäfer et al., 2014) using the Heritrix web crawler (Mohr et al., 2004). Other projects have either used the same or similar available software (Baroni et al., 2009) or created their own software in order to optimize the efficiency of the crawling process (Suchomel and Pomikálek, 2012). Crawling is a time-consuming and costly process (Olston and Najork, 2010), and a lot of the crawled data (up to roughly 95%) are usually not used in the final corpus because it does not pass one of the usual quality assessment algorithms or is duplicate material (Schäfer and Bildhauer, 2012; Suchomel and Pomikálek, 2012; Biemann et al., 2013; Schäfer and Bildhauer, 2013; Schäfer et al., 2014). It is thus desirable from a web corpus creation perspective to make crawling as efficient as possible or re-use available crawl data.

The *CommonCOW* (COCO) project uses CommonCrawl crawl data that are available via https and the Amazon S3

[1] http://corporafromtheweb.org/
[2] http://texrex.sourceforge.net/
[3] http://slurm.schedmd.com/
[4] https://hadoop.apache.org/

Corpus	No. of documents
(Dutch)	450,000
English	112,700,000
French	2,200,000
German	1,900,500
Spanish	4,400,000
(Swedish)	380,000

Table 1: Conservatively estimated sizes of the COCO1507 corpora; bracketed languages have been dropped due to low yield

cloud protocol.[5] The CommonCrawl initiative (with which I am not affiliated in any way) offers free access to very huge archived crawled snapshots of the web, and the data are archived by Amazon under their Public Datasets program.[6] I am are currently optimizing the COW tool chain for the processing of the CommonCrawl data using the July 2015 CommonCrawl snapshot. This snapshot consists of 33,957 WARC crawler archive files of an average 0.88 GB (gzipped), totalling at roughly 30 TB. Processing them on my university's high performance cluster with *texrex* takes about one week, and removing near-duplicates is expected to take another five to ten days. The corpus sizes for the COCO1507 corpora derived from this single CommonCrawl snapshot (estimated based on a test run with 100 WARC files from the same snapshot) are given in Table 2. Dutch and Swedish data will not be used to create corpora because they are represented too sparsely in the original data sets. However, based on the previous experience with similar crawl data, the English COCO1507 corpus is expected to be 110 billion tokens large.[7] The linguistic annotation should take about a month based on previous experience (Schäfer, 2015b).

3. Redistribution

3.1. Problem and solution

The redistribution of corpora—and especially web corpora—usually involves legal issues, especially in countries without a Fair Use doctrine that would allow non-commercial use of copyright-protected digital material for research purposes (Samuelson, 1995). For example, German copyright law ("Urheberrecht") requires that corpus designers explicitly ask the author of any text (that reaches a certain threshold of creativity) for their permission before inclusion of the text in the corpus.[8] This is clearly infeasible for large web corpora as described in Section 2. As a workaround, corpora are sometimes distributed as sentence shuffles under the assumption that single sentences never reach the required threshold of creativity, for example COW (Schäfer and Bildhauer, 2012; Schäfer, 2015b) or the Leipzig Corpora Collection (Biemann et al., 2007). This approach is outright unsatisfactory for users who need larger linguistic contexts or whole documents for their research. Even worse, it is not even clear whether there is a reliable copyright exemption for single sentences.[9]

For the redistribution of COCO, a unique solution for the distribution of corpora consisting of *intact* documents was developed that results in very high legal safety for the corpus creators. The method can be summarized as follows:

1. All annotations (including meta data and token-level annotations) from the COCO corpora are distributed freely in the form of stand-off annotation files under a maximally permissive Creative Commons CC-BY license as COCO Annotation (COCOA) files.[10]
2. In these COCOA files, the original tokens (at least most of them, cf. below) are replaced by an offset into a normalized version of original document such that they can later be retrieved from that original document. Since the COCOA format alone does not allow anyone to reconstruct the original work, I see little to no legal risk for the corpus creators.
3. A tool is deployed which automates the process of downloading both the COCOA files and the original CommonCrawl data and recreating a full corpus from both sources on the end user side. Thus, no copyrighted data are actually redistributed by the COW/ COCO corpus creators.

The crucial point is that the CommonCrawl initiative, being located in the U. S., are allowed to distribute the data because they can claim Fair Use. This does *not* mean that the data are free of copyright, and from a European perspective the problem arises that any derived data (in my case high quality annotated corpora) cannot be re-distributed if the original copyrighted material is included in the derivative work. In order to get the data to the end user regardless, I therefore separate the annotations from the data and distribute only the annotations along with the tools required to merge them with the original data. Of course, end users still need to make sure that it is legal for them to work with CommonCrawl data.[11] The solution proposed here focusses only on de facto legal safety for corpus creators.

In the remainder of the paper, I provide technical details and estimates of the overhead incurred for the end user based

[5] https://commoncrawl.org/

[6] http://aws.amazon.com/de/public-data-sets/

[7] A reviewer suggested that I should provide more statistics about other languages represented in the CommonCrawl data. While I agree that such statistics would be a great thing to have, my tools do not perform general-purpose language detection, but merely look for those languages for which we have linguistic annotation tool chains in the COW project.

[8] This holds regardless of whether the resulting corpus is distributed commercially or for academic purposes.

[9] A legal assessment recently commissioned by the German Research Council (DFG) is sceptical about the shuffle approach (DFG, 2015, 14). In the famous Infopaq case, snippets of eleven words were considered to reach the level of threshold (https://en.wikipedia.org/wiki/Infopaq_International_A/S_v_Danske_Dagblades_Forening).

[10] http://creativecommons.org/licenses/by/3.0/

[11] As one reviewer pointed out, it might even be illegal in some countries to download and store data with unclear copyright status.

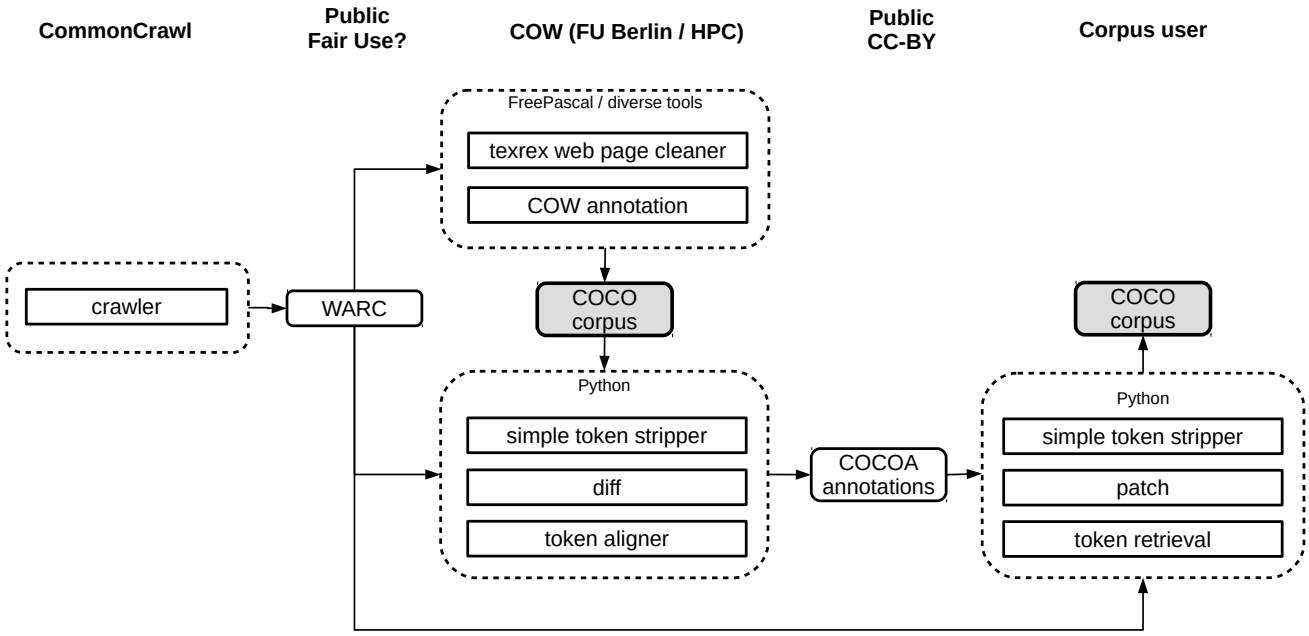

Figure 1: Workflow of COCO/COCOA corpus construction

on the current versions of the tools. Technically, two major problems have to be solved. One has to make sure that the alignment works perfectly (Section 3.2.), and one has to make sure sure that the process of reconstructing the corpus is feasible for potential users in terms of bandwidth and runtime (Section 3.3.).

3.2. Workflow and data integrity

Figure 3.1. summarizes the entire workflow. The CommonCrawl project runs a crawler to collect the primary HTML data, and they make them available for download in a standardized web archive (WARC) format. The COW project downloads the WARC files and creates clean, normalized, and annotated corpora from them using *texrex* and the COW annotation pipeline. The resulting CommonCOW (COCO) corpora are then aligned (cf. below) with the WARC data and transformed into stand-off CommonCOW Annotation (COCOA) files in such a way that end users can re-create the full corpora from those COCOA files. The end user then downloads both the COCOA and the WARC files in order to reconstruct the COCO corpus with a Python tool provided by me.

To illustrate the format in which the annotations are deployed, I provide Figure 2. The files are in CWB-compatible format (Evert and Hardie, 2011) with minimal XML and inlined one-token-per-line data where the annotation levels are separated with tabs. For each document (enclosed in a `<doc>` tag), the byte offset in the corresponding WARC file and its byte-length are specified. Instead of tokens, the indices i of the token in the original document are specified as `@i@`. For performance reasons, and because they are legally irrelevant, all tokens which do not consist entirely of letters are inserted literally (here `&apos;s` and `.`). To further obscure the original sentence (especially for languages without rich inflectional morphology like English), lemma annotations are replaced by `@id@` if they are

```
<doc ...  offset="92575231" length="12331">
<s>
<vc>
@8702@   VB    check    1   0  null
</vc>
<nc>
@8703@   DT    @id@     2   6  det
@8704@   NP    @id@     3   6  poss
&apos;s  POS   &apos;s   4   3  possessive
</nc>
<nc>
@8705@   NP    @id@     5   6  nn
@8706@   NN    @id@     6   1  dobj
</nc>
<advc>
@8707@   RB    @id@     7   1  advmod
</advc>
<pc>
@8708@   IN    @id@     8   1  prep
<nc>
@8709@   NNS   update   9   8  pobj
</nc>
</pc>
.        SENT  .        10  1  punct
</s>
```

Figure 2: Simplified extract of a COCOA annotation file; the sentence shown is *Check the EU's Customs website periodically for updates.*

identical (up to capitalization) to the token. Clearly, distributing data as shown in Figure 2 does not violate the original authors' copyright.

There were some technical difficulties which needed to be solved. The WARC files contain the raw HTML sources, and it is not a trivial task to locate the tokens as they appear in the final COCO corpus in the (usually messy) markup. My tools (*texrex* and the linguistic annotation tool chain) perform a large number of cleanups and normalizations, and consequently many of the tokens in the final corpus do not even occur literally in the HTML source code at all. After extensive experimentation, I implemented the following

procedure that effectively solves this problem.[12]

After the COCO corpus has been created, a Python tool goes through the WARC files again and applies fast and robust HTML stripping to the original data (document by document) and converts everything to UTF-8 (using the *BeautifulSoup* package).[13] From this extracted text, only the alphabetic characters and the blanks are retained, resulting in the *source token buffer*. Then, the tool extracts those tokens from the COCO corpus that consist exclusively of alphabetic characters (without annotations), creating the *target token buffer*. Ideally, the two text buffers should be identical, but because of the aforementioned normalizations, this is usually not the case. Using Google's *diff_patch_match* library, a diff-based patch is created which, if applied to the source token buffer, yields the target token buffer.[14] The patch object is pickled (i. e., serialized), compressed with zlib, Base64-encoded and included as non-human-readable textual data in the COCOA output within a custom XML tag.[15] The COCOA token information is then created from COCO by replacing all purely alphabetic tokens with their offsets in the target token buffer.

When the COCO corpus is re-created from WARC and COCOA files by the end user, the same type of HTML stripping and UTF-8 conversion is run on the document from the WARC file before the patch is applied. The clean corpus tokens can then be retrieved from the patched text buffer by looking up the indices stored in the COCOA file.

One further measure is taken to reduce the legal risk for the corpus creators. The URL of the original web page and its IP address are included as meta data in COCO corpora. In COCOA, they are encrypted in a way that they cannot be decrypted without also downloading the WARC files. In other words, the sources of the documents are obfuscated. This is achieved by creating a digest from the raw HTML document as found in the WARC file and using this digest as a key to encrypt the URL and IP address (by simply applying a byte-wise XOR). The user-side tool decrypts the meta data using the same method.

The Python reference implementations are released under a permissive BSD open-source license. By the time of this writing, they have been implemented and tested on small data sets but not on the actual large data sets. However, both the COCOA data files and the Python tools will be made available still in 2016.

[12] I previously had hopes that the plain-text versions offered directly by CommonCrawl, the so called WET files, could be used at least for the process that is run by the end user. The quality of the text extraction in the WET files is too low to be usable, however.

[13] `http://www.crummy.com/software/BeautifulSoup/`

[14] `https://code.google.com/archive/p/google-diff-match-patch/`

[15] The serialized patch inflates the size of the COCOA files by roughly 10%. Since the procedure described here is both highly reliable and makes processing on the end user side very effective, this overhead is a small price to pay.

Task	s per file
download WARC and COCOA	60s
COCOA/WARC merger	20s
total	80s

Table 2: Estimated overhead for end users involved in the corpus reconstruction process on a single Intel Core i7 CPU with a 100 Mbit/s downstream.

3.3. Feasibility for the end user

On the corpus creators' side, running the simple token stripper, creating the diff/patch, and aligning the tokens takes 300ms per document on average on a single Intel i7 or Xeon core. On my university's high performance cluster, processing the whole July 2015 CommonCrawl snapshot can be achieved in well under one day.[16] Based on the current development version, the performance on a standard consumer computer is estimated in Table 3.3.

Since there are approximately 34,000 files in the CommonCrawl July 2015 snapshot, in order to reconstruct the English 112 million document (110 billion token) COCOC1507 corpus (Table 2.), the tool will run for $80s \times 34,000 = 31.5d$.[17] Most better equipped computers will be able to run several instances of the tool simultaneously, thus reducing the total time needed. For a 112 million document/110 billion token corpus, this is clearly acceptable.

4. Outlook

The process of reconstructing a COCO corpus requires some effort, and the *end users* referred to in this paper are most likely predominantly institutions and research groups with well equipped machines and a high download bandwidth rather than individuals. Therefore, I hope that providers in countries with more permissive legislation use the tools described here to reconstruct COCO corpora and make them accessible (ideally also in convenient interfaces) while respecting the CC-BY license for the annotations and giving the COCO/COW project the required credit by referencing this paper. In the long run, I hope that initiatives like COCO will help to undermine the restrictive and anachronistic EU copyright laws and pave the way for a Fair Use doctrine in the EU.

In my own work, I will continue to process CommonCrawl snapshots (ideally all of them), mostly in order to increase corpus sizes for languages other than English. I will also include COCO data in current evaluations of the quality and composition of web corpora (partly joint work with Felix Bildhauer).

5. Acknowledgements

Research presented in this paper was funded by the German Research Council (DFG), grant SCHA1916/1-1. I would like to thank the Zedat data center of Freie Universität Berlin for computing time on their high-performance

[16] This is just the extra time needed for creating the COCOA files.

[17] Notice that the time required to actually create such a corpus from scratch is in the region of several CPU years (Schäfer, 2015b).

cluster (HPC). I am also grateful to Felix Bildhauer for his ongoing collaboration on the COW and COCO projects.

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., Heyer, G., Quasthoff, U., and Richter, M. (2007). The leipzig corpora collection - monolingual corpora of standard size. In *Proceedings of Corpus Linguistic 2007*, Birmingham, UK.

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):23–60.

Björkelund, A., Bohnet, B., Hafdell, L., and Nugues, P. (2010). A high-performance syntactic and semantic dependency parser. In *Coling 2010: Demonstration Volume*, pages 33–36, Beijing.

DFG. (2015). Handreichung: Informationen zu rechtlichen Aspekten bei der Handhabung von Sprachkorpora. Technical report, Deutsche Forschungsgemeinschaft.

Evert, S. and Hardie, A. (2011). Twenty-first century corpus workbench: Updating a query architecture for the new millennium. In *Proceedings of the Corpus Linguistics 2011 conference*, Birmingham. University of Birmingham.

Faruqui, M. and Padó, S. (2010). Training and evaluating a german named entity recognizer with semantic generalization. In *Proceedings of KONVENS 2010*, Saarbrücken, Germany.

Mohr, G., Stack, M., Ranitovic, I., Avery, D., and Kimpton, M. (2004). Introduction to Heritrix, an archival quality web crawler. In *Proceedings of the 4th International Web Archiving Workshop (IWAW'04)*.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., and Marsi, E. (2007). Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Olston, C. and Najork, M. (2010). *Web Crawling*, volume 4(3) of *Foundations and Trends in Information Retrieval*. now Publishers, Hanover, MA.

Pomikalek, J., Rychly, P., and Kilgarriff, A. (2009). Scaling to billion-plus word corpora. *Research in Computing Science*, 41. Special issue: Advances in Computational Linguistics.

Samuelson, P. (1995). Copyright's fair use doctrine and digital data. *Publishing Research Quarterly*, 11(1):27–39.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, September.

Schäfer, R. and Bildhauer, F. (2012). Building large corpora from the web using a new efficient tool chain. In Nicoletta Calzolari, et al., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 486–493, Istanbul. ELRA.

Schäfer, R. and Bildhauer, F. (2013). *Web Corpus Construction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool, San Francisco.

Schäfer, R., Barbaresi, A., and Bildhauer, F. (2013). The good, the bad, and the hazy: Design decisions in web corpus construction. In Stefan Evert, et al., editors, *Proceedings of the 8th Web as Corpus Workshop (WAC-8)*, pages 7–15, Lancaster. SIGWAC.

Schäfer, R., Barbaresi, A., and Bildhauer, F. (2014). Focused web corpus crawling. In Felix Bildhauer et al., editors, *Proceedings of the 9th Web as Corpus workshop (WAC-9)*, pages 9–15, Stroudsburg. Association for Computational Linguistics.

Schäfer, R. (2015a). Accurate and efficient general-purpose boilerplate detection. in prep.

Schäfer, R. (2015b). Processing and querying large web corpora with the COW14 architecture. In Piotr Bański, et al., editors, *Proceedings of Challenges in the Management of Large Corpora 3 (CMLC-3)*, Lancaster. UCREL.

Suchomel, V. and Pomikálek, J. (2012). Effcient Web crawling for large text corpora. In Adam Kilgarriff et al., editors, *Proceedings of the seventh Web as Corpus Workshop*, pages 40–44.

van Gompel, M., van der Sloot, K., and van den Bosch, A. (2012). Ucto: Unicode tokeniser. version 0.5.3. reference guide. ILK Technical Report ILK 12-05, Induction of Linguistic Knowledge Research Group, Tilburg Centre for Cognition and Communication, Tilburg University, Tilburg, November.