Proceedings of the 12th Web as Corpus Workshop, pages 5–13
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Out-of-the-Box and Into the Ditch?
Multilingual Evaluation of Generic Text Extraction Tools

Adrien Barbaresi[1], Gaël Lejeune[2]
(1) Berlin-Brandenburg Academy of Sciences (2) Sorbonne University
Jägerstraße 22-23 10117 Berlin (Germany) / 1 rue Victor Cousin 75005 Paris (France)
barbaresi@bbaw.de / gael.lejeune@sorbonne-universite.fr

Abstract

This article examines extraction methods designed to retain the main text content of web pages and discusses how the extraction could be oriented and evaluated: can and should it be as generic as possible to ensure opportunistic corpus construction? The evaluation grounds on a comparative benchmark of open-source tools used on pages in five different languages (Chinese, English, Greek, Polish and Russian), it features several metrics to obtain more fine-grained differentiations. Our experiments highlight the diversity of web page layouts across languages or publishing countries. These discrepancies are reflected by diverging performances so that the right tool has to be chosen accordingly.

Keywords: Web corpus construction, Web Content Extraction, Boilerplate removal, Evaluation metrics, Cleaneval

1. Introduction

1.1. Web corpus construction

Large “offline” web corpora are now standard throughout disciplines among the research community. Corpus construction notably involves “crawling, downloading, ‘cleaning’ and de-duplicating the data, then linguistically annotating it and loading it into a corpus query tool.” (Kilgarriff, 2007) Although text is ubiquitous on the Web, extracting information from web pages can prove to be difficult. They come in different shapes and sizes mostly because of the wide variety of platforms and content management systems, and not least depending on the context, for instance diverging goals followed during publication. This process involves a significant number of design decisions and turning points in data processing. Depending on the purpose of data collection, a substantial filtering and quality assessment can be crucial.

Recently, approaches using the CommonCrawl[1] have flourished as they allow for faster download and processing by skipping (or more precisely outsourcing) the crawling phase (Habernal et al., 2016; Schäfer, 2016). Barring the fact that finding one’s “own” way through the Web can be preferable, it is clear that such data should not be used without some filtering. Beside the discovery of relevant websites, a major issue consist in selecting appropriate content after download and processing (Schäfer et al., 2013), which may not be straightforward due to unexpected or machine-generated flaws and biases. Some large-scale algorithms can be expected to smooth out irregularities. However, uses requiring a low margin of error and close reading approaches imply constant refinements in the constitution and processing of the dataset, for example in the context of an aggregated lexical information platform (Geyken et al., 2017).

The potential lack of metadata is worsened by a lack of information regarding the content whose adequacy, focus and quality are the object of a *post hoc* evaluation (Baroni et al., 2009). A major challenge lies in the ability to extract and pre-process web data to meet scientific expectations with respect to corpus quality (Barbaresi, 2015). Because of the vastly increasing variety of corpora, text types and use cases, it becomes more and more difficult to assess the usefulness and appropriateness of the gathered web texts for given research objectives. Potential answers can reside in methods such as focused web crawling for corpus construction (Schäfer et al., 2014) and in a degree of focus concerning the selection of sources (Barbaresi, 2016; Barbaresi, 2019).

Regardless of the chosen construction method, an essential operation consists in retaining the desired content while discarding the rest, a polyonymous goal referring to peculiar subtasks or to the whole, most notably web scraping, boilerplate removal, web page segmentation, web page cleaning, or content extraction (Lejeune and Zhu, 2018). The variety of contexts and text genres leads to important design decisions during the collection of texts: could and should the tooling be adapted to particular sources that are targeted (which often amounts to the development of web scraping tools e.g. for news outlets) or should the extraction be as generic as possible to provide opportunistic ways of gathering information? Due to a lack of time resources in academia and elsewhere, the tools are considered as field-tested without a thorough evaluation *in vitro*. This article hopefully makes a step towards the latter.

1.2. State of the art of content extraction

As the use of templates is pervasive on the Web (Bar-Yossef and Rajagopalan, 2002), common approaches to main content detection include heuristic rules, machine learning on labeled training data, and indirectly template-based approaches (for example by identifying duplicated content) (Rae et al., 2018). Although text-based (Kohlschütter and Nejdl, 2008) and visual segmentation algorithms (Cai et al., 2003) have been published on, content extraction mostly draws on Document Object Model (DOM) examination (Gupta et al., 2003). That means considering a given HTML document as a tree structure whose nodes represent parts of the document to be operated on.

[1] https://commoncrawl.org

Text, tag and/or link density have proven to be good heuristics in order to select or discard content nodes, with approaches such as the Content Extraction via Tag Ratios (CETR) (Weninger et al., 2010) or the Content Extraction via Text Density (CETD) algorithms (Sun et al., 2011). Statistical selection of informative nodes through a combination of both methods proved more efficient on comparable datasets (Qureshi and Memon, 2012). Indeed, the large majority of DOM-based approaches try to leverage semantic information conveyed by HTML tags, notably paragraphs (*p*) on which text-to-tag ratios are calculated (Carey and Manic, 2016). An earlier, language-independent approach uses entropy measures applied to feature, links, and content in order to discriminate among parts of a webpage (Kao et al., 2004).

Machine learning approaches have also been used, whose interest generally consists in leveraging advances in classification tasks by treating a HTML document as a series of blocks to be classified. Relevant algorithms notably include conditional random fields (CRF) learning header, text or noisy blocks using markup-based, content-based, and document-related features (Spousta et al., 2008), support vector machines (SVMs) trained on linguistic, structural and visual features (Bauer et al., 2007), or more recently deep learning, for example with convolutional neural networks (CNNs) learning combinations of DOM-based features (Vogels et al., 2018).

Regarding the evaluation of extraction methods, the Cleaneval dataset and metrics (Baroni et al., 2008) have been used as a reference by numerous studies. Granularity and metrics used can have a real impact on results. Character and word-level metrics can be considered as a sequence, in a bag of words approach, or as a set and then ranked by F-score (Gottron, 2007).

Web text extraction is not a solved task, user experience in general turns web content extraction into an active field of research, resulting from higher download and rendering speeds overall as well as from a growing tendency to inject content from a wide variety of sources, notably through the development of "reader modes" and "distillers"[2] for web browsers which strive to reduce the amount of "Web bloat" (Ghasemisharif et al., 2019). Furthermore, many existing algorithms have become somewhat obsolete due to the rapid changes in web technologies over the last 15 years (Weninger et al., 2016). Web page structure is also constantly evolving from the perspective of standards. HTML 5 was first released in 2008 to provide support for multimedia and graphical elements. This standard also streamlined syntax while retaining backward-compatibility. It also provided ways to tag the semantic content of documents with a granularity unseen before, with new page structure elements such as *main*, *section*, *article*, *header*, *footer*, *aside*, or *nav*. The standard has been gradually integrated into publishing practices and content management systems, while the recommendations still evolve, the current standard being HTML 5.2.[3] In addition, publication systems combining HTML code with embedded JavaScript are on the rise, which also raises the question of "dry" and rendered page code.

Last, there is a disciplinary gap between computer scientists and corpus linguists, both at the time of and following the "web as corpus" paradigm. As well as other research traditions sharing the Web as a research object without communicating much (Brügger and Laursen, 2019), both communities do not seem to be interconnected, although they could benefit from each other's results. We believe content extraction does not get the amount of attention it deserves in the corpus linguistics community. Additionally, precise metadata extraction is paramount in the humanities and remains a collateral issue of this disciplinary gap.

1.3. Contributions

Distinguishing between whole page and essential parts can help to alleviate many quality problems related to web texts. While this is particularly useful in the case of deduplication and studies relying on frequency-based information, other tasks related to content extraction also benefit from a cleaner text base. In the concrete case of linguistic and lexicographic research, it allows for content checks on the only portion of the document that really counts.

In the following, we describe and evaluate text extraction tools published under open-source licenses and whose installation is straightforward. We perform a comparative benchmark on a multilingual setting consisting of real-world data with a manually annotated gold standard. We discuss the results as well as potentially suitable metrics to obtain more fine-grained differentiation. The insights of this paper are thus threefold in terms of software usability, benchmarking, and metrics.

2. Evaluation method

The evaluation described here focuses on integration and real-world usability of the tested solutions. As in previous evaluation campaigns we target the main content, which is usually the part displayed centrally, without the left or right bars, the header or the footer, but including potential titles and comments. We gathered tools coming from different research and industrial backgrounds, different countries, and developed during different time frames.

2.1. Tested solutions

The current benchmark focuses on the Python programming language which is reportedly the most popular programming language in academia[4] and one of the most popular overall. A few algorithms below are adapted from other languages such as Java and JavaScript, which contributes to giving an exhaustive yet incomplete panorama of available solutions overall.

The following tools keep the structure intact but don't focus on main text extraction, they are kept in the benchmark to see how they perform in terms of recall, that is in order to measure how easy it would be to simply gather all the extractable text:

[2] https://chromium.googlesource.com/chromium/dom-distiller

[3] https://www.w3.org/TR/2017/REC-html52-20171214/

[4] https://spectrum.ieee.org/computing/software/the-top-programming-languages-2019

- Html2text[5] performs text extraction
- Inscriptis[6] converts HTML to text with a particular emphasis on nested tables.

The following tools focus on main text extraction which is the task at hand:

- Boilerpy3[7] is a Python version of the boilerpipe algorithm (Kohlschütter et al., 2010) for boilerplate removal and fulltext extraction;
- Dragnet[8] works as a meta-classifier using different methods weighted by machine learning (Peters and Lecocq, 2013), it requires more dependencies and potentially fine-tuning or re-training to work at its best;
- Goose3[9] can extract information for embedded content but doesn't preserve markup;
- jusText[10] is designed to preserve mainly text containing full sentences along with some markup, it has been explicitly developed to create linguistic resources (Pomikálek, 2011);
- Newspaper[11] is mostly geared towards newspaper texts, provides additional functions but no structured text or comment extraction
- News-please[12] is a news crawler that extracts structured information (Hamborg et al., 2017);
- Python-Readability[13] is a Python port of the Readability library used in Firefox to display distraction-free webpages, it cleans the page and preserves some markup.

The systems are used out-of-the-box or with minimal fine-tuning. Some of them come from an academic and others from an engineering or commercial background. Some are not being actively developed while others are still being updated. There is no reason to believe some would be disadvantaged as the pages they are tested on are anterior to their development. We use different pre-tuned configurations (here after mode) for the tools that offer this possibility: boilerpy3 and jusText. All the code developed for this evaluations is available online.[14]
In the results section we will use the following names for the tools:

- BP3 for Boilerpy3 (default configuration) BP3_Art for the *Article* mode, BP3_KeepE for the *KeepEverything* mode and BP3_Larg for the *Largest* mode;

Data	NBlines	NBtokens	NBchar
Html	1385 (±1303)	4726 (±3921)	75015 (±51924)
Clean	13 (±10)	321 (±323)	2296 (±1982)

Table 1: Corpus statistics on the original Html pages and their manually cleaned versions

- DRAG for Dragnet;
- GOOSE for Goose3;
- JT for jusText (default configuration), JT_en for the *English* mode and JT_langid for the *language dependent* mode;
- NPAPER for Newspaper;
- NPLEASE for News-please;
- READ for *Python-Readability*.

2.2. Corpus

For our experiments we take advantage of the multilingual, human-annotated corpus DAnIEL, used previously for segmentation and event detection tasks (Lejeune et al., 2012) and extraction (Lejeune and Zhu, 2018). It comprises 1694 documents in five languages: Chinese, English, Greek, Polish and Russian. Each document is present as in its original Html version and as a cleaned version with the text and some markup. To the best of our knowledge it is the largest multilingual corpus for evaluating web content extraction tools.
The documents have been collected in 2011 and 2012 to evaluate a text classification tool. The Html 5 standard was not published as a W3C recommendation before 2014, thus it is to be expected that the documents analyzed here almost exclusively ground on Html 4 which has been a reference since the end of the 1990s.
We wish to compare the results of extrinsic evaluation (e.g. how does the web cleaning tool influence the result of classification) and intrinsic evaluation, e.g. to what extent the extracted content matches the expected outcome. We focus on the latter, not only to find the potentially "best" solution but also to provide more insights on the metrics and results of the evaluation. The dataset is available upon request.
Table 1 shows some statistics on the corpus, the Html original files and the manually curated clean versions. We can see two different families of tools:

- Recall oriented tools such as Html2Text, Inscriptis and BP3_KeepE: they tend to extract much more data than expected
- Precision-oriented tools (all the others) which are really devoted to avoid noise.

Table 2 and Table 3 show statistical descriptions of the output for all the tools, as we are looking for misses or near misses. We define almost empty documents as cases where the size of the output represents less than 10% of the size of the clean document. It shows how many times one can

[5] https://github.com/Alir3z4/html2text/
[6] https://github.com/weblyzard/inscriptis
[7] https://github.com/jmriebold/BoilerPy3
[8] https://github.com/dragnet-org/dragnet
[9] https://github.com/goose3/goose3
[10] https://github.com/miso-belica/jusText
[11] https://github.com/codelucas/newspaper
[12] https://github.com/fhamborg/news-please
[13] https://github.com/buriy/python-readability
[14] https://github.com/rundimeco/waddle

Data	NBlines	NBtokens	NBchar
BP3	22 (±27)	380 (±492)	2656 (±3091)
BP3_Art	16 (±21)	314 (±353)	2287 (±2328)
BP3_KeepE	188 (±133)	1189 (±1009)	8252 (±6363)
BP3_Larg	14 (±22)	285 (±345)	2049 (±2265)
DRAGNET	7 (±10)	252 (±326)	1723 (±2001)
GOOSE	6 (±10)	202 (±297)	1296 (±2091)
HTML2T	335 (±200)	1581 (±1307)	21204 (±13747)
INSCRI	243 (±176)	1409 (±1200)	20649 (±31550)
JT	14 (±17)	381 (±499)	2501 (±3092)
JT_en	6 (±14)	169 (±435)	1008 (±2549)
JT_langid	14 (±17)	376 (±496)	2467 (±3073)
NPAPER	8 (±12)	205 (±314)	1301 (±2015)
NPLEASE	15 (±21)	267 (±361)	1703 (±2277)
READ	35 (±76)	351 (±371)	2932 (±2729)

Table 2: Statistics on the output of the different tools and configurations

Data	el	en	pl	ru	zh
BP3	31.9%	6.9%	2.2%	5.7%	1.0%
BP3_Art	30.8%	6.9%	2.2%	5.3%	0.7%
BP3_KeepE	0.0%	3.6%	0.7%	1.9%	0.0%
BP3_Larg	30.8%	6.9%	2.2%	5.3%	1.0%
DRAGNET	49.1%	1.3%	10.9%	23.2%	3.4%
GOOSE	99.3%	1.5%	11.7%	65.4%	28.0%
HTML2T	0.0%	0.0%	0.0%	0.0%	0.0%
INSCRI	0.0%	0.0%	0.0%	0.0%	0.0%
JT	1.8%	4.2%	0.0%	0.4%	28.7%
JT_en	98.2%	4.2%	99.6%	99.6%	29.2%
JT_langid	1.8%	4.2%	0.0%	0.4%	28.7%
NEWSP	95.2%	1.0%	22.6%	95.4%	29.2%
NEWSP	46.5%	1.3%	5.1%	65.0%	92.9%
READ	0.7%	1.3%	2.2%	0.4%	17.9%

Table 3: Proportion of empty or almost empty ($< 10\%$ of the expected size) files for each language

Tool	Proc. time	Diff. with fastest
INSCRI	**19.7**	x1
DRAG	24.0	x1.2
BP3_KeepE	37.5	x1.9
BP3_Larg	37.7	x1.9
BP3	38.1	x1.9
BP3_Art	39.8	x2.0
JT_english	41.5	x2.1
READ	56.8	x2.9
HTML2T	71.0	x3.6
NPAPER	105.5	x5.5
JT_langid	112.6	x5.7
GOO	191.3	x9.7
JT	322.0	x16.3
NPLEASE	3755.6	x190

Table 4: Mean execution time over 5 iterations (in seconds) to process the test corpus (1694 documents) on a laptop

be sure that the output clearly does not fit the result one can expect from a text extractor. Obviously, the three tools of the recall-oriented family seldom output empty or almost empty files. Most tools seem to be primarily designed for English and not well-adapted to Chinese. We can see the importance of the jusText language models when compared to the English mode (JT_en). But the default configuration performs well, except in Chinese for which we had to adapt the configuration[15]. Because of differences in both data sample and processing it is important to choose appropriate metrics which can highlight disparities in tool efficiency. The metrics are described and discussed in the following section.

3. Results

3.1. Processing time

We present in Table 4 the processing time for each tool. There are noticeable differences between them, partly due to the fact that some tools go far beyond a mere text extraction, most notably News-please. We included this information as it needs to be taken into account for users that need to process data in real time or to clean big datasets but we won't discuss it thoroughly. We can see that Dragnet and Inscriptis seem to be the fastest systems, whereas language settings for jusText affect the results significantly.

3.2. Evaluation Metrics

Since the Cleaneval campaigns (Baroni et al., 2008), a state-of-the-art evaluation scheme has been set up and accepted by the community. This metric is based on the following assumption: a text is a sequence of tokens with or without Html tags and a good content extraction solution should preserve this sequence. The proposition consists in matching the longest common subsequence between a gold standard version of the text and the result given by an extractor. While there are still unmatched zones, the algorithm recursively finds the next longest common subsequence in these zones. The insertion of a sequence not present in the Gold Standard is a False Positive. Conversely, a sequence that is missing in the result of the extractor is a False Negative. This proved to be convenient since classical metrics like recall, precision and f-score can be computed.

However, this metric has some flaws. First of all, it has a quadratic complexity due to the use of the Ratcliff/Obershelp algorithm (Ratcliff and Metzener, 1988). Even on small datasets it is very slow. Secondly, it does not account properly for recall. For instance, copy-pasting the whole content of the document (e.g. with a very naive *html-to-text* tool) does not achieve 100% recall. As a consequence, we propose to use three additional metrics. Let GT be the Ground Truth and RES be the result of a given extractor and GT_{tok} and RES_{tok} be the sequence of their tokens. Let TP be the number of True Positives, FP the number of False Positives and FN the number of False Negatives.

In order to favor comparisons, the tokenization is produced by the exact same code as in Cleaneval except for Chinese where a segmentation in characters has been

[15] We followed the recommendations from the author: https://github.com/miso-belica/jusText/issues/12.

performed. [16]

The first one, `voc_eval`, simply compares the vocabulary of GT and RE:

- Let GT_{voc} be the set of GT_{tok} and RES_{voc} the set of RES_{tok}
- TP = $|GT_{voc} \cap RES_{voc}|$
- FP = $|RES_{voc} \setminus GT_{voc}|$
- FN = $|GT_{voc} \setminus SET_{voc}|$

The second one, `occ_eval` compares the number of occurrences for each token.

- For each token t in GT_{tok} :
  - $TP = 0, FP = 0, FN = 0$
  - Compute $freq(t_{GT})$ (resp. $freq(t_{RES})$) its frequency in GT (resp. in RES)
  - TP += $min(freq(t_{RES}), freq(t_{GT})$
  - FP += $freq(t_{RES}) - TP$
  - FN += $freq(t_{GT}) - TP$
- For each token u of $RES_{voc} \setminus GT_{voc}$:
  - FP += $freq(t_{RES})$

We also wish to apply other indicators in order to make other types of differences visible among all the tested tools. As such, we opt for two metrics: cosine and euclidean distance. These distances are regularly used for assessing the closeness between two documents (Platt et al., 2010; Buck and Koehn, 2016) , therefore we thought it could yield useful insights in this context.

The last one (`KL_eval`) uses the Kullback-Leibler divergence (a measure of relative entropy between two probability distributions):

- VOC = $GT_{voc} \cup RES_{tok}$ (union of the vocabularies of GT and RES)
- Let P_{gt} (resp. P_{res}) be the probability distribution in GT (resp. RES) of each token of VOC
- for all x in P_{gt} (resp. P_{res}):
  - if $P_{gt}(x) = 0$ (resp.$P_{res}(x) = 0$)
    - $P_{gt}(x) \leftarrow 10^{-5}$ (resp. $P_{res}(x) \leftarrow 10^{-5}$)
- $D_{KL}(P_g \parallel P_{res}) = -\sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P_g(x)}{P_{res}(x)}\right)$

The Kullback-Leibler divergence is not a distance metric since it is not symmetric but it is a way to measure how probability distributions diverge. In our case, we do not need a symmetric measure since we just want to account for the closeness with the GT probability distribution.

The first two metrics allow us to compute recall, precision and f-score whereas `KL_eval` yields a single measure: the smaller the divergence, the greater the similarity of the two documents.

3.3. Evaluation on the multilingual corpus

Table 5 lists the results of each tool on the `clean-eval` evaluation scheme. The precision and recall are means, which is important for the interpretation since document-wise evaluation tends to favor systems that do not yield results much smaller that expected. The f-score is the classical version (with $\beta = 1$) computed on the mean precision and mean recall. We could also have chosen to compute a mean of the different f-scores but decided it would be strange to have a geometric mean of harmonic means.

The first thing we can see is that BP3 is very efficient. READABILITY offers a slightly worse result but with a higher recall whereas JUSTEXT exhibits a drop in recall in comparison. DRAGNET has the highest precision score but with a recall below 60%. The recall-oriented tool family leads to lower scores but we can see that INSCRIPTIS is better than HTML2TEXT in both recall and precision. It seems to be a good tool for task when it is important to get as much content as possible.

Tool	f-score	precision	recall
BP3_Art	**78.84**	82.80 (±26)	75.24 (±34)
BP3_Larg	76.44	84.57 (±26)	69.74 (±35)
READ	75.87	72.18 (±28)	79.96 (±27)
BP3	72.83	75.42 (±25)	70.41 (±33)
JT	71.22	78.93 (±25)	64.88 (±41)
JT_langid	70.71	78.96 (±25)	64.02 (±41)
DRAGNET	69.66	**87.53** (±22)	57.85 (±38)
NPLEASE	58.46	69.00 (±41)	50.72 (±45)
GOO	53.93	83.89 (±22)	39.74 (±43)
NPAPER	50.83	82.20 (±22)	36.78 (±44)
BP3_KeepE	47.19	31.74 (±20)	91.97 (±20)
INSCRI	42.95	27.72 (±17)	**95.28** (±13)
JT_en	37.15	79.81 (±21)	24.21 (±39)
HTML2T	33.98	20.86 (±16)	91.47 (±15)

Table 5: Evaluation with the `clean-eval` metric, sorted by descending f-score (computed on the mean precision and the mean recall)

The `clean-eval` measures for the quality of web page cleaning is widely used but it uses a convoluted algorithm relying on the alignment of sequences of words. Its rationale is quite straightforward: nobody wants to have a discontinuous version of the data or to have words in the wrong order. But it appears that in HTML code, the sequence of text blocks is in the same order as the original text. One can see there is not much difference between this evaluation and `occ_eval` (Table 7). There are some differences in ranking concerning the `voc_eval` metric (Table 6. Therefore, we can say that we can use the `occ_eval` metric which has the advantage of being around ten times faster to compute.

Table 8 shows the evaluation with cosine distance, euclidean distance and Kullback-Leibler divergence. Interestingly, this metric seems to be able to highlight systems that show a good balance between silence and noise (like READABILITY and JUSTEXT). Moreover, it does not penalize much systems with large recall scores (like INSCRIPTIS or HTML2TEXT).

[16] See https://github.com/rundimeco/waddle

Tool	f-score	precision	recall
JT	**75.68**	81.83 (±22)	70.39 (±36)
JT_langid	75.32	81.93 (±22)	69.69 (±35)
BP3_Art	75.30	78.96 (±26)	71.96 (±34)
BP3Larg	73.75	80.94 (±26)	67.74 (±34)
READ	72.52	70.91 (±29)	74.21 (±31)
BP3	71.78	73.22 (±25)	70.40 (±33)
DRAGNET	68.94	86.23 (±22)	57.43 (±36)
NPLEASE	67.28	**92.51** (±17)	52.86 (±44)
GOOSE	60.08	89.51 (±19)	45.21 (±41)
NPAPER	56.78	88.78 (±18)	41.74 (±42)
BP3_KeepE	45.82	30.73 (±22)	90.01 (±20)
INSCRI	45.69	30.56 (±21)	**90.43** (±18)
JT_en	43.57	88.17 (±18)	28.94 (±38)
HTML2T	36.59	23.10 (±17)	87.96 (±18)

Table 6: Evaluation with the `voc_eval` metric

Tool	f-score	precision	recall
BP3_Art	**76.38**	80.60 (±24)	72.57 (±33)
BP3_Larg	74.54	82.90 (±24)	67.72 (±33)
JT	74.13	81.36 (±23)	68.08 (±37)
JT_langid	73.73	81.50 (±23)	67.31 (±37)
READ	73.25	72.43 (±28)	74.09 (±30)
BP3	72.50	74.27 (±24)	70.81 (±32)
DRAGNET	67.09	86.82 (±21)	54.67 (±37)
NPLEASE	66.64	**92.03** (±17)	52.23 (±44)
GOOSE	57.74	89.42 (±19)	42.64 (±42)
NPAPER	54.78	88.68 (±18)	39.63 (±43)
BP3_KeepE	42.02	27.41 (±21)	89.98 (±18)
JT_en	41.35	88.09 (±18)	27.01 (±39)
INSCRI	37.10	23.22 (±18)	**92.22** (±13)
HTML2T	33.45	20.56 (±17)	89.80 (±14)

Table 7: Evaluation with the `occ_eval` metric

Tool	KL div.	Euclidean	Cosine
JT	1.15 (±1.5)	0.17 (±0.2)	**0.12** (±0.1)
BP3_KeepE	1.16 (±1.0)	0.22 (±0.1)	0.36 (±0.2)
JT_langid	1.17 (±1.5)	0.17 (±0.2)	**0.12** (±0.1)
INSCRI	1.18 (±0.7)	0.25 (±0.1)	0.41 (±0.2)
BP3_Art	1.21 (±1.8)	0.15 (±0.2)	0.13 (±0.2)
BP3	1.29 (±1.8)	0.17 (±0.2)	0.15 (±0.2)
HTML2T	1.31 (±0.8)	0.21 (±0.1)	0.41 (±0.2)
BP3_Larg	1.31 (±1.8)	0.15 (±0.2)	0.13 (±0.2)
READ	1.38 (±2.2)	0.17 (±0.3)	0.17 (±0.3)
DRAGNET	1.87 (±2.1)	0.22 (±0.3)	0.19 (±0.2)
GOOSE	2.66 (±2.5)	0.36 (±0.3)	0.26 (±0.3)
NPAPER	2.98 (±2.6)	0.40 (±0.3)	0.29 (±0.3)
NPLEASE	3.36 (±3.6)	0.60 (±0.7)	0.36 (±0.4)
JT_en	3.76 (±2.5)	0.51 (±0.3)	0.36 (±0.3)

Table 8: Evaluation with the `KL_eval` metric, euclidean and cosine distances

Tool	f-score	precision	recall
NPAPER	**90.36**	90.39 (±17)	90.33 (±15)
GOOSE	89.76	**92.01** (±17)	87.62 (±16)
DRAGNET	88.01	87.80 (±21)	88.23 (±19)
NPLEASE	87.83	86.86 (±16)	88.83 (±15)
READ	86.21	83.50 (±19)	89.11 (±16)
BP3_Art	85.95	86.18 (±18)	85.71 (±28)
JT	83.63	82.04 (±23)	85.29 (±25)
BP3_Larg	82.92	87.26 (±20)	78.98 (±30)
JT_langid	82.68	82.03 (±24)	83.34 (±26)
JT_en	82.68	82.03 (±24)	83.34 (±26)
BP3	81.40	77.32 (±20)	85.94 (±26)
BP3_KeepE	52.38	36.36 (±21)	93.65 (±20)
INSCRI	45.74	29.81 (±17)	**98.24** (±4)
HTML2T	44.17	28.70 (±17)	95.82 (±7)

Table 9: Evaluation with the `clean-eval` metric (documents in English) , sorted by descending f-score

This is not surprising since, even with smoothing, this measure tends to favor close probabilities in the same order of magnitude, in other words $P(x) = 1 * 10^{-4}$ is closer to $Q(x) = 3 * 10^{-4}$ than $R(x) = 1 * 10^{-5}$.

3.4. Results by language

The results on the five languages of the corpus describe major discrepancies between the tools. First of all, Table 9 shows the results obtained on English documents with the `clean-eval` metric and Table 10 the results for the `occ_eval` metric. Again, we can see that `occ_eval` yields comparable results. Since it is a simpler measure we will focus on this one for the remainder of the article.

One can see that the scores are much higher than the scores showed in Tables 5 and 7, which highlights that English is a very specific case. Our results demonstrate that most tools are primarily designed to process English documents. Furthermore, the tools that perform very well in this subcorpus are not as efficient on the multilingual corpus. So, one cannot rely on results evaluated solely on English to draw conclusions on the efficiency of a tool in real-world multilingual settings.

Except the three recall-oriented tools, all yield an `occ_eval` f-score of 80% and higher. NEWSPAPER outperforms the other tools with an f-score above 90%. GOOSE is slightly below and close to NEWSPLEASE but it is much faster (around 35 times according to Table 4). The three tools designed for readability (READABILITY itself but also NEWSPAPER and NEWS-PLEASE) all perform very well.

Table 11 introduces the results on the Greek subcorpus. The three best tools perform comparably to the three top tools for English. It is interesting to see that the language-dependent JUSTEXT configuration yields results comparable to the default configuration. NEWSPAPER, GOOSE and obviously JT_EN perform poorly on this subcorpus. It is obvious for the latter but it is astonishing that the other two do not perform well.

Table 12 shows the results obtained on the Polish subcorpus. We can see that the results are much lower than in English and Greek, both in terms of precision and recall. The best performers on the English subcorpus do not offer comparable results except for NEWSPLEASE andJUSTEXT.

Tool	f-score	precision	recall
NPAPER	**91.32**	91.34 (±16)	91.31 (±14)
GOOSE	90.69	**92.94** (±16)	88.54 (±15)
NPLEASE	88.91	87.89 (±15)	89.96 (±14)
DRAGNET	88.78	88.52 (±19)	89.04 (±18)
READ	87.16	84.31 (±18)	90.21 (±15)
BP3_Art	87.00	87.50 (±17)	86.51 (±28)
JT	84.86	83.16 (±22)	86.62 (±24)
BP3_Larg	84.72	89.14 (±18)	80.73 (±28)
JT_langid	84.08	83.35 (±22)	84.83 (±25)
JT_en	84.08	83.35 (±22)	84.83 (±25)
BP3	82.56	78.46 (±20)	87.11 (±26)
BP3_KeepE	52.66	36.56 (±21)	94.08 (±19)
INSCRI	45.84	29.88 (±17)	**98.46** (±4)
HTML2T	44.61	28.98 (±17)	96.84 (±6)

Table 10: Evaluation with the `occ_eval` metric (documents in English)

Tool	f-score	precision	recall
JT_langid	**88.95**	90.41 (±21)	87.54 (±21)
JT	88.80	89.97 (±21)	87.66 (±21)
READ	86.62	83.03 (±19)	90.54 (±11)
BP3_Art	74.63	88.17 (±19)	64.70 (±44)
BP3_Larg	74.58	89.56 (±18)	63.90 (±43)
BP3	74.17	87.60 (±17)	64.31 (±44)
NPLEASE	65.07	**96.00** (±12)	49.21 (±47)
BP3_KeepE	51.20	34.79 (±16)	96.92 (±5)
INSCRI	50.66	34.21 (±15)	**97.56** (±5)
DRAGNET	43.82	93.94 (±15)	28.57 (±33)
HTML2T	41.03	26.06 (±14)	96.39 (±5)
NPAPER	5.58	92.98 (±18)	2.88 (±12)
GOOSE	2.98	95.11 (±12)	1.51 (±6)
JT_en	2.33	94.10 (±16)	1.18 (±1)

Table 11: Evaluation with the `occ_eval` metric (documents in Greek)

Tool	f-score	precision	recall
BP3_Art	**84.20**	85.11 (±22)	83.32 (±26)
NPLEASE	83.13	86.02 (±21)	80.44 (±29)
JT	82.47	77.71 (±25)	87.85 (±17)
JT_langid	82.15	77.89 (±25)	86.90 (±18)
BP3_Larg	81.40	86.24 (±23)	77.07 (±28)
DRAGNET	79.79	85.84 (±21)	74.54 (±33)
READ	79.23	77.50 (±23)	81.03 (±24)
BP3	78.11	73.03 (±24)	83.96 (±23)
GOOSE	74.84	86.32 (±25)	66.05 (±35)
NPAPER	73.86	85.04 (±21)	65.28 (±41)
BP3_KeepE	48.42	32.69 (±18)	93.35 (±14)
INSCRI	43.28	28.00 (±16)	**95.28** (±11)
HTML2T	36.06	22.45 (±15)	91.57 (±11)
JT_en	1.96	**91.06** (±16)	0.99 (±1)

Table 12: Evaluation with the `occ_eval` metric (documents in Polish)

Tool	f-score	precision	recall
JT	**76.29**	71.64 (±29)	81.59 (±22)
JT_langid	75.99	71.57 (±29)	80.99 (±23)
READ	74.27	72.29 (±27)	76.36 (±26)
BP3_Larg	72.58	77.30 (±31)	68.40 (±34)
BP3_Art	69.31	70.11 (±35)	68.53 (±34)
BP3	66.50	60.82 (±28)	73.34 (±28)
DRAGNET	50.94	85.13 (±27)	36.34 (±31)
NPLEASE	42.64	93.16 (±20)	27.64 (±41)
GOOSE	40.24	90.96 (±21)	25.83 (±38)
BP3_KeepE	36.93	23.30 (±20)	88.89 (±19)
INSCRI	32.53	19.77 (±16)	**91.75** (±17)
HTML2T	29.55	17.63 (±14)	91.35 (±14)
NPAPER	5.14	92.34 (±19)	2.64 (±9)
JT_en	3.55	**95.37** (±12)	1.81 (±6)

Table 13: Evaluation with the `occ_eval` metric (documents in Russian)

Tool	f-score	precision	recall
BP3_Art	**63.30**	71.28 (±24)	56.93 (±22)
BP3_Larg	57.95	72.53 (±24)	48.26 (±22)
BP3	55.20	70.08 (±25)	45.53 (±19)
DRAGNET	44.53	81.81 (±23)	30.59 (±18)
READ	42.36	48.00 (±32)	37.91 (±28)
GOOSE	20.60	82.54 (±17)	11.77 (±9)
JT_langid	19.19	82.32 (±17)	10.86 (±5)
JT	19.19	82.32 (±17)	10.86 (±5)
JT_en	19.18	82.80 (±17)	10.84 (±5)
NPAPER	19.17	82.72 (±17)	10.84 (±5)
BP3_KeepE	19.08	10.85 (±15)	78.94 (±18)
HTML2T	13.83	7.62 (±11)	74.87 (±15)
NPLEASE	13.31	**97.52** (±12)	7.14 (±13)
INSCRI	12.97	7.06 (±10)	**79.52** (±14)

Table 14: Evaluation with the `occ_eval` metric (documents in Chinese), evaluation by character n-grams

It seems harder to extract text from Russian pages since no system is able to achieve above 80% f-score (Table 13). Again, JusText is among the best performers. Contrary to the Polish subcorpus, it is BP3_Larg that is the best BP3 configuration. We can see again that Readability performs very well on other languages than English.

Finally, the worst results are related to the Chinese subcorpus (Table 14). BP3 outperforms the rest of the field by far. One can see that the choice of a tool is much more important for Chinese than for English since many tools result in f-scores below 20%. We can note that it is the only language for which Inscriptis does not achieve 90% recall.

3.5. Is there a winner?

The results we presented yield differentiated insights so that it is difficult to give a definitive and universal answer. First of all, if one targets recall and/or speed Inscriptis is clearly the most efficient solution. In general BP3 and Readability are the most stable systems and the only ones that perform reasonably well for Chinese.

If we do not consider Chinese, JUSTEXT in its language-independent setting seems to be the most efficient solution for multilingual corpora. That being said, this setting is much slower and it is not strictly comparable as it uses additional information but most of all it does not appear to perform better. For texts in English GOOSE and NEWSPAPER outperform the other systems. For Polish, BP3_ART shows a comparable f-score than JUSTEXT but with a better precision. For Russian BP3_LARG is a good solution if one needs precision but JUSTEXT achieves a satisfying trade-off between precision and recall. According to our study, there appears to be no benefit from more intricate machine-learning approaches, DRAGNET does not stand out and does not perform poorly either. However, the amount of additional training data needed to potentially improve its results is a penalty in terms of usability compared to the other solutions for which parameter tuning could lead to improvements much faster. JUSTEXT is such an example where changing settings can be done easily.

4. Conclusions and outlook

The article focused on a comparative benchmark of open-source tools used on web documents from 2011 and 2012 written in five different languages, along with a discussion of suitable metrics. Content processing is affected by both diatopic and diachronic factors, whereas vocabulary analysis and distance metrics can yield more fine-grained information which complements the CLEANEVAL evaluation standard. Rule-based approaches appear to be more efficient in the long run, all the more since they are both easier to use and to parametrize.

Most tools are developed with particular page styles in mind, mostly from the English-speaking world. Our data shows that linguistic factors are most probably reflected in HTML structures, which deeply affects extraction processes. The experiments above highlight the diversity of layouts and web coding practices depending on language and most probably on the country from which a document is published. These discrepancies are reflected by diverging performances so that the right tool has to be chosen accordingly.

In addition, different eras of web development result in diverging “HTMLects”. Our corpus provides a snapshot of a past version of the Web which proves to be challenging for some tools. As such, it is useful to assess how data from Web archives can be processed. These findings prompt for further studies on the evaluation of tool robustness with respect to the ever-changing Web. We have reasons to believe that the success of standardized publishing platforms and the consecutive advent of HTML 5 changes the way text is published on the Web, all of which could pave the way for further examinations.

5. References

Bar-Yossef, Z. and Rajagopalan, S. (2002). Template Detection via Data Mining and its Applications. In *Proceedings of the 11th International Conference on World Wide Web*, pages 580–591.

Barbaresi, A. (2015). *Ad hoc and general-purpose corpus construction from web sources*. Ph.D. thesis, École Normale Supérieure de Lyon.

Barbaresi, A. (2016). Efficient construction of metadata-enhanced web corpora. In Paul Cook, et al., editors, *Proceedings of the 10th Web as Corpus Workshop*, pages 7–16. Association for Computational Linguistics.

Barbaresi, A. (2019). The Vast and the Focused: On the need for thematic web and blog corpora. In Piotr Bański, et al., editors, *Proceedings of the CMLC-7 workshop*, pages 29–32.

Baroni, M., Chantree, F., Kilgarriff, A., and Sharoff, S. (2008). Cleaneval: a Competition for Cleaning Web Pages. In *Proceedings of LREC*, pages 638–643. 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.

Bauer, D., Degen, J., Deng, X., Herger, P., Gasthaus, J., Giesbrecht, E., Jansen, L., Kalina, C., Kräger, T., Märtin, R., Schmidt, M., Scholler, S., Steger, J., Stemle, E., and Evert, S. (2007). FIASCO: Filtering the internet by automatic subtree classification. In *Building and Exploring Web Corpora: Proceedings of the 3rd Web as Corpus Workshop (WAC-3)*, pages 111–121.

Brügger, N. and Laursen, D. (2019). Introduction: Digital humanities, the web, and national web domains. In *The Historical Web and Digital Humanities*, pages 1–9. Routledge.

Buck, C. and Koehn, P. (2016). Quick and reliable document alignment via TF/IDF-weighted cosine distance. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 672–678, Berlin, Germany, August. Association for Computational Linguistics.

Cai, D., Yu, S., Wen, J.-R., and Ma, W.-Y. (2003). VIPS: a Vision-based Page Segmentation Algorithm. Technical report, Microsoft Research.

Carey, H. J. and Manic, M. (2016). HTML web content extraction using paragraph tags. In *25th International Symposium on Industrial Electronics (ISIE)*, pages 1099–1105. IEEE.

Geyken, A., Barbaresi, A., Didakowski, J., Jurish, B., Wiegand, F., and Lemnitzer, L. (2017). Die Korpusplattform des ”Digitalen Wörterbuchs der deutschen Sprache” (DWDS). *Zeitschrift für germanistische Linguistik*, 45(2):327–344.

Ghasemisharif, M., Snyder, P., Aucinas, A., and Livshits, B. (2019). SpeedReader: Reader Mode Made Fast and Private. In *Proceedings of the World Wide Web Conference*, pages 526–537.

Gottron, T. (2007). Evaluating content extraction on HTML documents. In *Proceedings of the 2nd International Conference on Internet Technologies and Applications*, pages 123–132.

Gupta, S., Kaiser, G., Neistadt, D., and Grimm, P. (2003). DOM-based content extraction of HTML documents. In *Proceedings of the 12th international conference on World Wide Web*, pages 207–214.

Habernal, I., Zayed, O., and Gurevych, I. (2016).

C4Corpus: Multilingual Web-size corpus with free license. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 914–922.

Hamborg, F., Meuschke, N., Breitinger, C., and Gipp, B. (2017). news-please: A generic news crawler and extractor. In Maria Gaede, et al., editors, *Proceedings of the 15th International Symposium of Information Science*, pages 218–223.

Kao, H.-Y., Lin, S.-H., Ho, J.-M., and Chen, M.-S. (2004). Mining web informative structures and contents based on entropy analysis. *IEEE Transactions on Knowledge and Data Engineering*, 16(1):41–55.

Kilgarriff, A. (2007). Googleology is bad science. *Computational Linguistics*, 33(1):147–151.

Kohlschütter, C. and Nejdl, W. (2008). A Densitometric Approach to Web Page Segmentation. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 1173–1182.

Kohlschütter, C., Fankhauser, P., and Nejdl, W. (2010). Boilerplate detection using shallow text features. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM '10*, pages 441–450.

Lejeune, G. and Zhu, L. (2018). A New Proposal for Evaluating Web Page Cleaning Tools. *Computación y Sistemas*, 22(4).

Lejeune, G., Brixtel, R., Doucet, A., and Lucas, N. (2012). Daniel: Language independent character-based news surveillance. In *International Conference on NLP*, pages 64–75. Springer.

Peters, M. E. and Lecocq, D. (2013). Content extraction using diverse feature sets. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 89–90.

Platt, J., Toutanova, K., and Yih, W.-t. (2010). Translingual document representations from discriminative projections. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 251–261, Cambridge, MA, October. Association for Computational Linguistics.

Pomikálek, J. (2011). *Removing boilerplate and duplicate content from web corpora*. Ph.D. thesis, Masaryk University.

Qureshi, P. A. R. and Memon, N. (2012). Hybrid model of content extraction. *Journal of Computer and System Sciences*, 78(4):1248–1257.

Rae, A. R., Kim, J., Le, D., and Thoma, G. R. (2018). Main Content Detection in HTML Journal Articles. In *Proceedings of the ACM Symposium on Document Engineering 2018*, pages 1–4, New York, NY, USA. ACM.

Ratcliff, J. W. and Metzener, D. E. (1988). Pattern Matching: The Gestalt Approach. *Dr. Dobb's Journal*, 13(7):46.

Schäfer, R., Barbaresi, A., and Bildhauer, F. (2013). The Good, the Bad, and the Hazy: Design Decisions in Web Corpus Construction. In *Proceedings of the 8th Web as Corpus Workshop*, pages 7–15.

Schäfer, R., Barbaresi, A., and Bildhauer, F. (2014). Focused Web Corpus Crawling. In *Proceedings of the 9th Web as Corpus workshop (WAC-9)*, pages 9–15.

Schäfer, R. (2016). CommonCOW: Massively Huge Web Corpora from CommonCrawl Dataand a Method to Distribute them Freely under Restrictive EU Copyright Laws. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16)*, pages 4500–4504.

Spousta, M., Marek, M., and Pecina, P. (2008). Victor: the Web-Page Cleaning Tool. In *4th Web as Corpus Workshop (WAC-4)*, pages 12–17.

Sun, F., Song, D., and Liao, L. (2011). DOM-based content extraction via text density. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 245–254.

Vogels, T., Ganea, O.-E., and Eickhoff, C. (2018). Web2text: Deep structured boilerplate removal. In *European Conference on Information Retrieval*, pages 167–179. Springer.

Weninger, T., Hsu, W. H., and Han, J. (2010). CETR: content extraction via tag ratios. In *Proceedings of the 19th international conference on World Wide Web*, pages 971–980.

Weninger, T., Palacios, R., Crescenzi, V., Gottron, T., and Merialdo, P. (2016). Web Content Extraction – a Meta-Analysis of its Past and Thoughts on its Future. *ACM SIGKDD Explorations Newsletter*, 17(2):17–23.