Out-of-the-Box and Into the Ditch? Multilingual Evaluation of Generic Text Extraction Tools
Adrien Barbaresi, Gaël Lejeune

To cite this version:
Adrien Barbaresi, Gaël Lejeune. Out-of-the-Box and Into the Ditch? Multilingual Evaluation of Generic Text Extraction Tools. Language Resources and Evaluation Conference (LREC 2020), 2020, pp.5-13. hal-02732851

HAL Id: hal-02732851
https://hal.science/hal-02732851v1
Submitted on 2 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Out-of-the-Box and Into the Ditch?
Multilingual Evaluation of Generic Text Extraction Tools

Adrien Barbaresi¹, Gaël Lejeune²
(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 CommonCrawl¹ 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.

¹https://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 demand 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” for web browsers which strive to reduce the amount of “Web bloat” (Ghasemsharif 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). Page web 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. 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 de-duplication 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 academia4 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:

---

4https://spectrum.ieee.org/computing/software/the-top-programming-languages-2019
The following tools focus on main text extraction which is the task at hand:

- **BOILERPY** is a Python version of the boilerpipe algorithm (Kohlschütter et al., 2010) for boilerplate removal and fulltext extraction;
- **DRAGNET** 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;
- **GOOSE** can extract information for embedded content but doesn’t preserve markup;
- **JUSTEXT** 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** is mostly geared towards newspaper texts, provides additional functions but no structured text or comment extraction
- **NEWS-PLEASEx** is a news crawler that extracts structured information (Hamborg et al., 2017);
- **PYTHON-READABILITY** 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: **BOILERPY** and **JUSTEXT**. All the code developed for this evaluations is available online.

In the results section we will use the following names for the tools:

- **BP3 for BOILERPY** (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 |
|--------|---------|----------|--------|
| Htm    | 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** for the **English** mode and **JT_langid** for the **language dependent** mode;
- **NPAPER** for **NEWSPAPER**;
- **NPLEASEx** 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 2 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/hamborg/news-please
13 https://github.com/bury/python-readability
14 https://github.com/rundimeco/waddle
Table 2: Statistics on the output of the different tools and configurations

| 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|
| JINSCR    | 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  |
| NPAAPER   | 8±12    | 205±314  | 1301±2015  |
| NPLEASE   | 15±21   | 267±361  | 1703±2277  |
| READ      | 35±76   | 351±371  | 2932±2729  |

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

| Data      | cl | 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| 0.0| 11.7| 65.4| 28.0|
| HTML2T    | 0.0| 0.0| 0.0| 0.0| 0.0|
| JINSCR    | 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|

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 INSCRIPITIS 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
performed.\footnote{See https://github.com/rundimeco/waddle}

The first one, eval$_{voc}$, 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$_voc$ ∩ RES$_voc$|
- FP = |RES$_voc$ \ GT$_voc$|
- FN = |GT$_voc$ \ 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$_{voc}$ \ 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\cite{Platt2010, Buck2016}, 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 $P_g$ (resp. $P_{res}$) be the probability distribution in GT (resp. RES) of each token of VOC
- for all $x$ in $P_g$ (resp. $P_{res}$):
  - if $P_g(x) = 0$ (resp. $P_{res}(x) = 0$)
    - $P_g(x) \leftarrow 10^{-5}$ (resp. $P_{res}(x) \leftarrow 10^{-5}$)
  - $D_{KL}(P_g || P_{res}) = - \sum_{x \in X} P_g(x) \log \left( \frac{P_g(x)}{P_{res}(x)} \right)$

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 JUS TEXT 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     | 75.24   |
| BP3_Larg| 76.44   | 84.57     | 69.74   |
| BP3    | 75.87   | 72.18     | 79.96   |
| READ   | 72.83   | 75.42     | 70.41   |
| JT     | 71.22   | 78.93     | 64.88   |
| JT_lanid| 70.71   | 78.96     | 64.02   |
| DRAGNET| 69.66   | 87.53     | 57.85   |
| NPLEASE| 58.46   | 69.00     | 50.72   |
| GOO    | 53.93   | 83.89     | 39.47   |
| NPAPER | 50.83   | 82.20     | 36.78   |
| BP3_KeepE| 47.19   | 31.74     | 91.97   |
| INSCRI | 42.95   | 27.72     | 95.28   |
| JT_en  | 37.15   | 79.81     | 24.21   |
| HTML2T | 33.98   | 20.86     | 91.47   |

### 3.4 Results on the multisegmented resources

Table\[6\] 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 JUS TEXT). Moreover, it does not penalize much systems with large recall scores (like INSCRIPTIS or HTML2TEXT).
The results on the five languages of the corpus describe major discrepancies between the tools. First of all, tools that perform very well in this subcorpus are primarily designed to process English documents. An obvious example is the N-gram based tools. This is not surprising since, even with smoothing, this measure tends to favor close probabilities in the same order of magnitude, in other words \( P(x) = 1 \times 10^{-4} \) is closer to \( Q(x) = 3 \times 10^{-4} \) than \( R(x) = 1 \times 10^{-5} \).  

### 3.4. Results by language

The results on the five languages of the corpus describe major discrepancies between the tools. First of all, Table 8 shows the results obtained on English documents with the clean-eval metric and Table 9 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.

The three tools designed for readability (READABILITY, GOOSE, and COSE) and cosine distances perform very well. The three tools designed for readability (READABILITY, GOOSE, and COSE) outperform the other tools with an f-score above 90%. COSE is slightly below and close to NEWSPLEASE but it is much faster (around 35 times according to Table 9). The three tools designed for readability (READABILITY, GOOSE, and COSE) all perform very well.

Table 9 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 justtext 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 10 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 and justtext.
Table 10: Evaluation with the \textit{occ\_eval} metric (documents in English)

| Tool         | f-score | precision | recall |
|--------------|---------|-----------|--------|
| NPAPER       | 91.32   | 91.34 (±16) | 91.31 (±14) |
| GOOSE        | 90.69   | 92.94 (±16) | 88.54 (±15) |
| NPLEASE      | 89.81   | 92.89 (±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 (±44) |
| HTML2T       | 44.61   | 28.98 (±17) | 96.84 (±64) |

Table 11: Evaluation with the \textit{occ\_eval} metric (documents in Greek)

| Tool         | f-score | precision | recall |
|--------------|---------|-----------|--------|
| BP3\_Art     | 84.20   | 85.11 (±22) | 83.26 (±20) |
| 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 (±44) |
| 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 \textit{occ\_eval} metric (documents in Polish)

Table 13: Evaluation with the \textit{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        | 40.60   | 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 \textit{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, \textsc{JustText} is among the best performers. Contrary to the Polish corpus, it is BP3\_Larg that is the best BP3 configuration. We can see again that \textsc{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 \textsc{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 \textsc{Inscriptis} is clearly the most efficient solution. In general BP3 and \textsc{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 Bafinski, 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 WaCy 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., Martin, R., Schmidt, M., Scholler, S., Steiger, 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.

Ghasemifar, 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). Gooleology 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). Foo-