A New Proposal for Evaluating Web Page Cleaning Tools

Gaël Lejeune[1], Lichao Zhu[2]

[1] STIH Sorbonne University, Paris, France

[2] LSHS Paris XIII University, Villetaneuse, France

gael.lejeune@paris-sorbonne.fr, lichao.zhu@gmail.com

Abstract. In this article, we tackle the problem of evaluation of Web Content Extraction tools. This task is seldom studied in the literature although it has important consequences on the linguistic processing of web-based corpora. Here, we compare two types of evaluation. Firstly, an intrinsic (content-based) evaluation which is carried out in a multilingual setting (five languages). Secondly, an extrinsic (task-based) evaluation on the same corpus by studying the effects of the cleaning step on the performances of an NLP pipeline. We show that in the intrinsic evaluation, the results are not consistent with extrinsic evaluation results. We also show that the results differ greatly in the studied languages. We conclude that the choice of a web page cleaning tool should be made with respect to the task that is tackled rather than the performances observed through the intrinsic evaluation scheme.

Keywords. Corpus, multilingual corpora, Web content extraction, Web page cleaning, evaluation, classification.

1 Introduction

Many NLP research projects take advantage of the huge amount of available online textual data. These data have shown a great impact in the field by widening the range of accessible tasks and available techniques. With more data, we could use data-intensive techniques such as textometry or machine learning. However, as evidenced in [2], it becomes more and more difficult to verify if data quality is sufficient for the research objectives. For instance, caution should be taken when using web obtained texts to train a POS tagger, if there is too much noise in the data. Raw documents are difficult to use in a NLP pipeline, because pre-processing steps are needed in order to get a clean text. This problem is often taken to light for PDF documents where the structure, as well as the sentences and words, cannot be extracted properly from each and every document [7]. Documents in raw HTML cannot be straightforwardly processed neither. The source code contains non-textual (or non-informative) elements which are not required for NLP tasks. Furthermore, the noise may even reduce downstream the efficiency of NLP modules.

There is no bi-univocity with HTML : the same rendering can be obtained via various source codes. As W3C standards are seldom respected in practice, web browsers tend to interpret the code in order to correct coding errors or to adapt the code to a particular terminal. In some aspects, HTML show some properties of a natural language : in order to facilitate communication between web coders and users, browsers became very tolerant, accepting syntax errors and allowing ambiguities. To some extent, pre-processing web pages for NLP tasks can be viewed as a binary classification task: the positive class is the text and the negative class is the rest. In a nutshell, the objective is extracting textual segments and/or discarding noise (advertisement, templates, code...). [3] pointed out that this is a very important task since errors can affect corpus statistics in a way that it requires further inspection. Interestingly, this task has received various names, highlighting the different points of view on this task: *boilerplate removal* or *boilerplate detection* [11], *Web Page*

Computación y Sistemas, Vol. 22, No. 4, 2018, pp. 1249–1258
ISSN 1405-5546
doi: 10.13053/CyS-22-4-3062

Template Detection [16], *Web Page Cleaning* [15], or *Web Content Extraction* [10].

In this article, we will focus on techniques that extract the textual content of web pages in order to preserve the integrity of a corpus. We will refer to this task as "Web Content Extraction" (WCE). Our objective is to compare the characteristics of tools developed for this task and to examine different ways to evaluate them. In Section 2 we will expose in details the problems behind Web Content Extraction. In Section 3 we will describe the characteristics of various tools. We will present in section 4 evaluation metrics and data for evaluation. The tools will be evaluated in section 5. We will discuss the results and the evaluation metrics in section 6.

2 State of the Art for Web Content Extraction

From the reader's point of view, discriminating the real textual content seems an easy task. Though website ergonomy may vary a lot, it is easy for the reader to parse the web page at first glance: the title and corresponding article are in the center, surrounded by boilerplateand advertisements. The same template, with tiny variations, is visible in most of the sites. The majority of variations can be found in the page for categories (finance, sport ...) and in the main page. As pointed out by the web-designer Andy Ruledge[1], these differences are motivated by design and advertisement issues rather than ergonomy. It appears that readers use complex strategies to adapt their behaviour to different websites so to automate this process is not trivial. Reading the newspaper on a smartphone without using a dedicated application can be really difficult because the browser is not always able to display correctly the main (textual) content of the page. This issue led to projects like READABILITY[2] which aim to improve the reading experience in using a browser. This problem had been pointed out a few years ago by researchers like [1].

[1] http://andyrutledge.com/news-redux.php

[2] https://github.com/keepcosmos/readability

Here, we choose to focus on press articles for evaluation purposes but we advocate that these issues can be encountered with any type of web harvested data since many data are not available in RSS related format. This allows us to benefit from available gold standard data in different languages (data described in Section 4).

The task of web content extraction (WCE) can be described as a classification problem. Given segments (organized as a list or as a tree), the problem is to classify them as informative or non-informative. Figure 1 shows a proposition of zonal classification for a press article from the web[3]:

informative (solid), segments that belong to the informative content: headline, titles and paragraphs;

borderline (dotted), segments potentially informative: author, date, caption;

non-informative segments giving very few information: boilerplate, advertisement...

In state-of-the-art techniques, the borderline category is generally considered as non-informative, probably because it is more convenient to view a problem a binary classification task. SCHAFER-2013 advocated a more subtle way to present the problem. Other authors pointed out the limits of this binary view, for instance the gold-standard corpus built for the BOILERPIPE tool [11] relies on a finer classification scheme. In this article, the authors proposed a typology of segments given by decreasing informativeness with their proportion in the corpus given in brackets:

1. title, subtitle, headline and body; (13%);
2. article date and captions (3%);
3. readers commentaries (1%);
4. related content, links to related articles (4%);
5. non-informative class (79%).

[3] https://tinyurl.com/lefigaro-fishpedicure

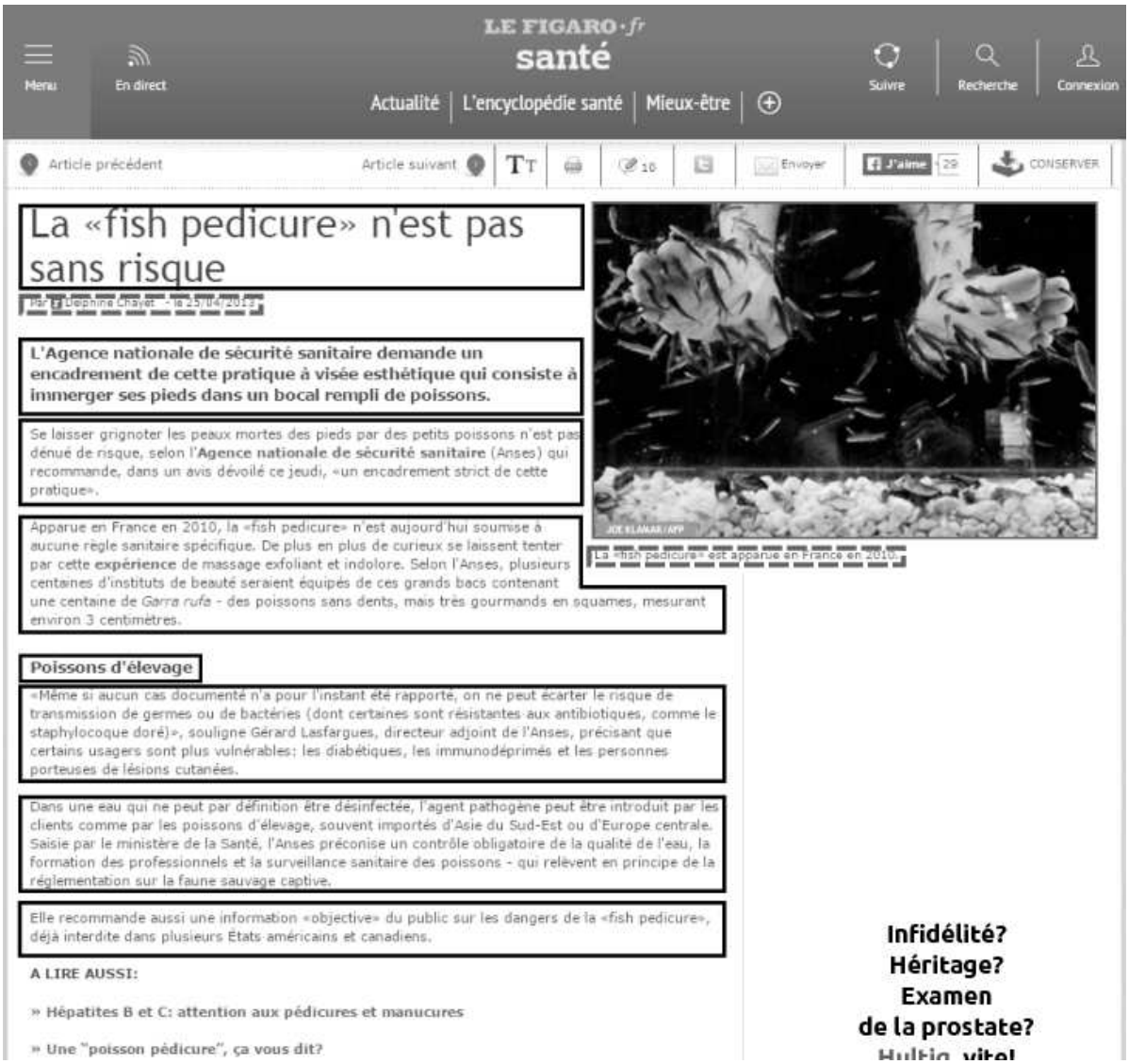

Fig. 1. Example taken from `www.lefigaro.fr`: informative segments are boxed, other elements are non-informative (advertisment, boilerplate).

The authors pinpoint that keeping track of the structure (title levels, lists . . .) is of great interest. We conclude that the text extraction process is a two-step process:

cleaning : removing JAVASCRIPT code, stylesheet information, boilerplate (menu, header, footer);

structuring : tagging each informative segment as a title, paragraph, list item . . .

Table 1 shows an output that one can expect from a text extraction process for the example presented in Figure 1. Borderline segments have been tagged as follows: `<author>`, `<date>` et `<caption>`.

Table 1. Expected output for Figure 1

Tags	Content
h	La ” fish pedicure” n’est pas sans risque
author	Par Dephine Chayet
date	25/04/2013
caption	La ” fish pedicure” est apparue en France en 2010.
p	L’Agence nationale de sécurité sanitaire demande [. . .]
p	Se laisser grignoter les peaux mortes des pieds [. . .]
p	Apparue en France en 2010, la ” fish pedicure” n’est [. . .]
h	Poissons d’élevage
p	Même si aucun cas documenté n’a pour l’instant été [. . .]
p	Dans une eau qui ne peut par définition être désinfectée,[. . .]
p	Elle recommande aussi une information ” objective” [. . .]

3 Designing Web Content Extraction Tools

3.1 Features for text Extraction

On of the most intuitive ways to perform text extraction is to take advantage of the *Document Object Model* (DOM), because there is a strong connection between text extraction and web page segmentation like in [5]. For instance, [16] used DOM level similarities among numerous pages coming from the same website. Similar structures are supposed to belong to the non-informative content whereas structural differences will be a clue to detect informative content. A similar approach used tree properties of the DOM, [6] advocates that the relative position of a node in the tree is a strong hint to distinguish informative contents from non-informative contents. There are more shallow strategies to make the most of the HTML structure. [9] used HTML tags density, [8] and [15] relied on n-gram models and [12] proposed to combine these two kinds of features. These tools of various features can be sorted in four categories according to their analysis levels:

Website : common characteristics for different pages;

Rendering : observation of browser(s) rendering;

HTML structure : hierarchy between blocks;

Textual content : sentences, words, n-grams.

3.2 Choice of tools for this Study

We examine in this study three freely available tools exhibiting interesting characteristics in performance and are widely known among the community. Firstly, BOILERPIPE which has been for many years the favorite of the NLP community. Secondly, NCLEANER has the particularity to use character-level language models and has participated in the first CLEANEVAL campaign. Finally, JUSTEXT is a more recent tool which has outperformed BOILERPIPE in various evaluation run by his author [13]. At first, we wanted to include the well-known tool READABILITY[4], but no official version is available for free.

Table 2. Weight of feature types for every tool: irrelevant (-), marginal (★) important (★★) or very important (★★★).

	Website Template	Browser Rendering	HTML Structure	Textual Content
`Boilerpipe`	-	★	★	★★★
`NCleaner`	-	★	-	★★★
`Justext`	-	-	★★	★★★

3.3 `Boilerpipe`

`Boilerpipe`[5] combines criteria designed to model the properties of the content of informative segments as opposed to the content of non-informative segments. The website dimension is not taken into account because, according to the authors, it would make the system more website-dependent and would imply an imbalance between websites with respect to the number of available pages. The rendering aspect is slightly used, only an estimation of the optimal width of a line (80 characters) is exploited in order to assess that a block is made to be read by a human. The most common HTML tags in textual segments are identified: (sub)titles (`<h1>` to `<h6>`), paragraphs (`<p>`) and container (`<div>`). On the contrary, the `<a>` tag allows to identify segments that are unlikely to be informative.

[4] `https://readability.com/`
[5] `http://code.google.com/p/boilerpipe/`

The main feature for `Boilerpipe` is the mean length of tokens in characters. A token is defined as a character string without blanks or punctuation. It is combined with local features and contextual features. Capitalized words, links, pipes ("—") mark non-informative content. On the contrary, points and commas are indicators of informative content. The contextual features are a hypothesis on the relative position of informative and non-informative segments: the blocks of the same class tend to be consecutive (and vice-versa). Therefore, the class of a segment is strongly dependent on the class of the previous and the next segment. For this reason, the rendering is simulated by considering that a segment contains as many lines as it can fill in columns of length 80 (considered as the optimal length for a human reader) while each segment has a minimal length of 1. For each segment, the token density is the number of tokens divided by the number of lines in the segment.

3.4 `NCleaner`

`NCleaner`[6] uses character *n*-grams language models [8]. `NCleaner` computes the probability that a given character belongs to the textual content by analyzing its left context $Pr(c_i|c_1...c_{i-1})$, where c_i is the character at offset i.

The method identifying the *n*-grams (with $1 \leq n \leq 3$) maximises the probability that a given segment belongs to the informative class. The model may be multilingual or computed separately for each language. Three settings can be used:

Default (NC): Language independent *n*-gram model

Non Lexical (NCNL): Turns letters in `a` and digits in `0`

Trained (NCTx): Trained with x pairs ($d_{\texttt{raw}}$, $d_{\texttt{clean}}$)

[6] `https://tinyurl.com/cleaneval`

3.5 `Justext`

`Justext` is a freely available tool which can be used via an API[7], its process has two separated steps [13]. In the first one (*context-free*), three features are computed for each segment: length in *tokens*, number of links and number of function words (according to a predefined list). The system can work with or without these language-dependent lists. For each segment, a first label is obtained with the aforementioned features:

Bad : non-informative

Good : informative

Near good : probably informative

Short : too short to be classified

The second step (*context-sensitive*) adapts the classification of the short and near good segments according to the class of their neighbors. A *Short* segment is classified *Good* if its neighbors are either *Good* or *Near-good*. A *Near-good* one is classified*Good* if at least one of its neighbors is *Good*.

4 Methods and Corpus for Evaluation

The CLEANEVAL framework allows to evaluate the effectiveness of content extraction and the correctness of the structure. The evaluation script given by the organizers has three configurations: *text only* (TO) and *text and markup labelled* (TM) or *unlabelled* (III). In the latter configuration, the name of the tag is not taken into account, so that the sequence `<p><p><l>` is equivalent to `<p><p><p>`. For each document, an automatically cleaned version is compared to the Gold Standard via a transformation in a token sequence. An edit distance between the two sequences is obtained by applying the Ratcliff algorithm [14]. This algorithm matches the longest common subsequences and then applies recursively in the unmatched zones. With the example given in Table 3, it is possible to compute recall and precision.

Although the CLEANEVAL metrics have been widely used in the domain, there are some

[7] `http://nlp.fi.muni.cz/projects/justext/`

Table 3. Putting into practice the Ratcliff algorithm for evaluating the difference between a test sequence $s_1 =$ "$totititoti$" and the Gold Standard $s_2 =$ "$tototiti$".

Operation	Offset	Substring (length)	Evaluation Influence
Insertion	0	"to" (2)	$FalseNegatives + 2$
No change	2	"$totiti$" (6)	$TruePositives + 6$
Deletion	6	"$toti$" (4)	$FalsePositives + 4$

drawbacks that we want to mention. First of all, in the TM configuration, all tokens (word or markup) have the same weight so that a system offering very bad markup may still have good results. Then, the use of graphic words as tokens is not fit for languages like Chinese. Finally, the way the edit distance is transformed into False postives/negatives brings up a paradox on the interpretation of the evaluation: a system returning all the segments as positive will not get a 100% recall, which is counter-intuitive. The CLEANEVAL script has therefore been improved by introducing a character-level evaluation. In this configuration, a *token* is a character. However, the measures given are still hard to interpret. For instance, the second example in Figure 2 where each of the tools selected some noisy segments. According to classic CLEANEVAL measures, there is a clear advantage for BOILERPIPE; as for humans, JUSTEXT made more reliable choices by keeping the caption rather than a reader comment. Interestingly, character-based evaluation seems to be more reliable in that particular case. In the next section, we will describe our corpus and propose measures for extrinsic evaluation in order to verify if there is a correlation between intrinsic evaluation results and "real life" application.

Building a gold standard with a reasonable size is a time-consuming task, particularly considering that we have two objectives in mind: (I) testing on various languages and (II) performing a task-based evaluation. To our knowledge, these constraints were not met by any of the corpora used to evaluate the tools presented above. The DANIEL corpus [4] has been the closest thing we could get as a good multilingual corpus for extrinsic evaluation. It contains documents in five languages (Chinese, English, Greek, Polish and

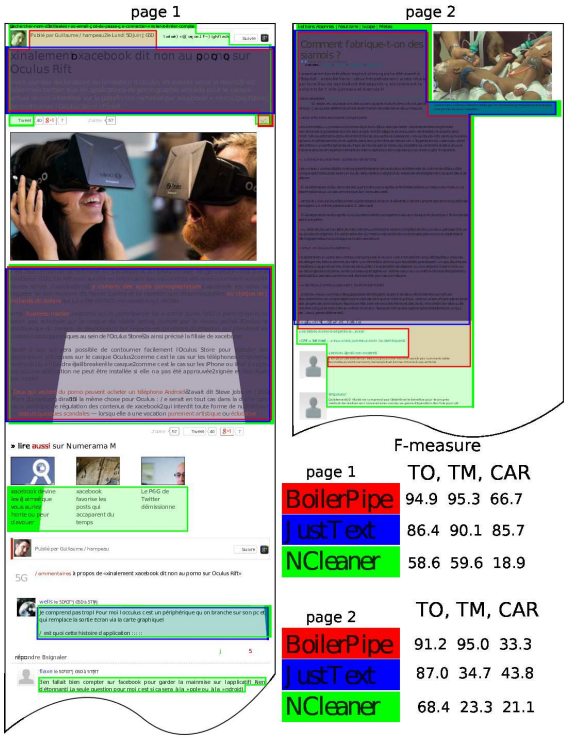

Fig. 2. Example of intrinsic evaluation of the three tools with default settings

Russian) and is available with manually curated content. However, we found one major issue with this corpus: the structure is very poor since each segment is tagged as a paragraph. In consequence, we were not able to perform the *labeled* version of the *text and markup* evaluation. The corpus has been released for evaluating a classification system specialized on epidemic surveillance. The code for the system is available online [8] although we had to ask directly the authors to get the appropriate lexical resources for all the languages of the corpus. Unfortunately, the original HTML files are not provided, so we had to retrieve them. Since the corpus has been constituted in 2012, some of the original files were no longer online. About 80% of the corpus has been retrieved with important variations between languages (Table 4).

[8] https://github.com/rundimeco/daniel

Table 4. DANIEL Number of documents in the corpus and proportion of retrieved ones (zh: Chinese, en: English, el: Greek, pl: Polish, ru: Russian)

	zh	en	el	pl	ru	Total
Files	446	475	390	352	426	2089
Retrieved	91%	100%	70%	78%	63%	81%
Pos. class	16	31	26	30	41	144
Retrieved	100%	100%	**65%**	90%	71%	83%

5 Intrinsic and Extrinsic Evaluation

The tools experimented in this section are the following: BOILERPIPE (BP) JUSTEXT with stoplist (JTA) and without stoplist (JTS) (Section 3.5), NCLEANER in its standard configuration (NC), and its learning configuration with 5 (NT5) and 25 (NT25) text pairs. We also tried to combine the Text extraction Tools in a pipeline fashion. For instance, $BP-JTS$ means that the document was first cleaned with BP and then was given as input to JTS.

Table 5. Intrinsic evaluation on all languages: Precision (P), Recall (R) and F_1-measure (F_1): *Text Only* (TO), CHAracter (CHA) and *Text and Markup* (TM).

	TO			CHA			TM		
	P	R	F_1	P	R	F_1	P	R	F_1
BP	81.8	**88.9**	**85.2**	76.9	**81.1**	**79.0**	64.5	**85.4**	**73.5**
BP–JTA	85.0	80.2	82.5	76.9	63.8	69.8	73.3	58.7	65.2
BP–JTS	83.2	82.9	83.1	75.1	66.0	70.3	69.2	62.1	65.5
JTA	68.8	83.4	75.4	63.8	67.0	65.4	61.9	63.2	62.6
JTA–BP	72.5	85.9	78.6	69.4	73.3	71.3	66.8	69.3	68.0
JTS	62.7	86.3	72.6	56.9	68.6	62.2	54.2	66.6	59.8
JTS–BP	66.3	88.7	75.9	63.0	75.8	68.8	59.4	72.7	65.4
NC	**98.5**	39.4	56.3	**96.7**	23.1	37.34	**89.0**	30.8	45.8
NCT5	60.4	23.8	34.2	53.8	16.0	24.7	48.4	19.9	28.2
NCT25	56.1	25.7	35.3	53.3	18.7	27.7	45.1	21.8	29.4

These figures show that BP outperforms the other tools and the combinations when we evaluate the entire corpus. At first, we expected to achieve good results by combining the good recall of BP and the high precision of NC with a $BP-NC$ pipeline, but all combinations gave poor results.

A language-by-language analysis (Table 6)[9] shows that BP makes the difference with rather isolating languages like Chinese and English. When we consider morphologically rich languages (particularly Russian), the results are more balanced and combining the tools gives better results.

[9]For this table, we excluded NCLEANER because its results, except precision, were really bad.

Table 6. Results by language for intrinsic evaluation

(a) Chinese

	TO			CAR			TM		
	P	R	F_1	P	R	F_1	P	R	F_1
BP	**61.3**	**52.9**	**56.8**	**77.1**	**63.5**	**69.7**	84.5	**68.0**	**75.3**
BP–JTA	39.6	8.6	14.1	76.4	26.0	38.8	44.4	9.79	16.1
BP–JTS	39.6	8.6	14.1	76.4	26.0	38.8	44.4	9.79	16.1
JTA	23.2	11.7	15.6	71.3	32.0	44.2	49.3	16.8	25.0
JTA–BP	49.6	31.2	38.4	69.0	30.7	42.5	**89.9**	32.7	48.0
JTS	23.2	11.7	15.6	71.3	32.0	44.2	49.3	16.8	25.0
JTS–BP	49.6	31.2	38.4	69.0	30.7	42.5	**89.9**	32.7	48.0

(b) English

	TO			CAR			TM		
	P	R	F_1	P	R	F_1	P	R	F_1
BP	86.0	**92.0**	**88.9**	84.9	**91.0**	**87.9**	69.3	**93.6**	79.6
BP–JTA	**87.0**	82.6	84.7	**87.6**	79.8	83.5	**81.1**	76.0	78.5
BP–JTS	86.4	85.3	85.8	87.2	82.8	85.0	79.4	80.2	**79.8**
JTA	68.4	85.4	76.0	70.0	82.8	75.8	67.2	79.7	72.9
JTA–BP	75.7	88.0	81.4	75.7	87.0	80.9	71.1	82.9	76.6
JTS	66.7	88.2	75.9	68.2	86.0	76.0	64.0	83.9	72.6
JTS–BP	73.8	90.9	81.5	73.8	90.5	81.3	68.3	87.3	76.6

(c) Polish

	TO			CAR			TM		
	P	R	F_1	P	R	F_1	P	R	F_1
BP	83.3	85.3	**84.3**	80.8	82.1	**81.4**	63.3	**85.6**	**72.8**
BP–JTA	**85.2**	78.3	81.6	**83.0**	73.3	77.8	**76.3**	67.7	71.8
BP–JTS	83.8	81.8	82.8	82.0	77.1	79.5	71.7	73.6	72.6
JTA	67.8	82.6	74.5	67.6	78.5	72.7	62.7	73.0	67.5
JTA–BP	68.6	84.0	75.5	66.3	79.4	72.3	61.1	74.6	67.2
JTS	63.2	86.1	72.9	62.6	81.3	70.7	54.1	77.6	63.8
JTS–BP	64.3	**87.5**	74.1	61.8	**82.5**	70.7	53.4	78.9	63.7

(d) Russian

	TO			CAR			TM		
	P	R	F_1	P	R	F_1	P	R	F_1
BP	58.8	79.3	67.5	51.7	70.2	59.5	37.9	85.5	52.5
BP–JTA	**67.8**	72.2	**69.9**	**53.6**	56.3	54.9	**48.6**	62.9	54.8
BP–JTS	61.9	75.2	67.9	48.5	58.5	53.0	40.3	67.1	50.4
JTA	52.7	81.8	64.1	41.3	63.4	50.0	42.9	75.0	54.6
JTA–BP	53.8	83.5	65.4	48.9	76.1	**59.6**	45.6	82.4	**58.7**
JTS	45.5	85.2	59.3	34.5	64.3	45.0	32.2	80.1	46.0
JTS–BP	46.6	**86.8**	60.6	42.6	**80.0**	55.5	34.4	**86.7**	49.2

(e) Greek

	TO			CAR			TM		
	P	R	F_1	P	R	F_1	P	R	F_1
BP	91.8	**96.5**	**94.1**	**87.6**	**91.6**	**89.5**	66.7	**91.7**	77.2
BP–JTA	**93.6**	90.9	92.2	80.0	76.8	78.3	**86.2**	78.8	82.3
BP–JTS	93.3	92.8	93.0	80.0	78.6	79.3	84.0	81.8	**82.9**
JTA	88.1	90.1	89.1	73.4	74.0	73.7	76.7	74.7	75.7
JTA–BP	88.9	91.5	90.2	85.2	87.5	86.3	75.7	76.8	76.3
JTS	70.8	92.6	80.2	59.4	74.9	66.2	62.2	78.4	69.3
JTS–BP	72.5	93.8	81.8	71.2	89.7	79.3	66.9	81.2	73.3

In our opinion, these results show that there is a strong interest in digging deeper. Table 7 shows the results for the task-based (extrinsic) evaluation. The classification results obtained by DANIEL on the reference corpus are compared to those in the automatically cleaned documents.

Table 7. Results for extrinsic evaluation, N/A represents non computable values (no True Positives). Red figures show cases where the results after WCE are better, green figures show cases where the WCE did not affect the result.

	English			Chinese			Greek			Polish			Russian			All Docs		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
BP	60.0	25.7	36.0	79.0	**93.8**	**85.7**	**85.7**	35.9	50.0	76.5	48.1	59.1	76.2	55.2	64.0	**74.7**	47.6	58.1
BP–JTA	61.5	45.7	52.5	71.4	31.2	43.5	66.7	70.6	**68.6**	65.6	77.8	71.2	76.7	**79.3**	78.0	68.1	**62.1**	**65.0**
BP–JTS	**65.2**	42.9	51.7	71.4	31.2	43.5	63.2	70.6	66.7	64.7	**81.5**	**72.1**	74.2	**79.3**	76.7	67.5	**62.1**	64.7
JTA	55.2	45.7	50.0	66.7	37.5	48.0	59.1	**76.5**	66.7	59.3	59.3	59.3	82.1	**79.3**	**80.7**	64.3	59.7	61.9
JTA–BP	59.1	37.1	45.6	66.7	37.5	48.0	62.5	58.8	60.6	65.4	63.0	64.2	76.0	65.5	70.4	66.3	52.4	58.6
JTS	55.6	42.9	48.4	66.7	37.5	48.0	66.7	58.8	62.5	56.7	63.0	59.6	**82.6**	65.5	73.1	64.4	54.0	58.8
JTS–BP	60.9	40.0	48.3	66.7	37.5	48.0	66.7	47.1	55.2	63.0	63.0	63.0	78.3	62.1	69.2	67.0	50.8	57.8
NC	58.3	**60.0**	**59.1**	N/A	0.00	N/A	N/A	0.00	N/A	80.0	14.8	25.0	N/A	0.00	N/A	61.0	20.2	30.3
NCT5	52.9	25.7	34.6	**83.3**	31.2	45.5	N/A	0.00	N/A	**82.3**	51.9	63.6	60.0	20.7	30.8	63.0	27.4	38.2
NCT25	50.0	25.7	34.0	**83.3**	31.2	45.5	20.0	5.88	9.09	**82.3**	51.9	63.6	61.5	27.6	38.1	62.7	29.8	40.4
Reference	**68.9**	**88.6**	**77.5**	**80.0**	**100**	**88.9**	**68.4**	**76.5**	**72.2**	**61.8**	**77.8**	**68.8**	**72.7**	**82.8**	**77.4**	**69.5**	**84.7**	**76.4**

In the reference line are mentioned the results obtained with the gold standard texts (manually cleaned). We note that they are slightly different from the original article since we only take into account documents where the original HTML version has been found. One can see that the results for JTA and JTS are strictly identical. This is due to the fact that no stop-list is used for this particular language. Interestingly, both JTA-BP and JTS-BP combinations have the same results. The reason is that for Chinese there is little difference in the content extracted by the two tools as we can see in Table 6a. NCLEANER performs globally worse but obtains some good results in English and in Polish. In fact, the performances vary a lot in different languages. See for instance Tables 6b to 6d[10].

Obviously, all the tools seem to be firstly trained for English corpora. This is probably the main reason of the performance gap between JT and BP, the largest difference at the advantage of the first one is observed in the results of Table 6b. However, this is not correlated with the in vivo performances presented in Table 7. With the Greek corpus, BP performs even better in intrinsic evaluations but again there is no correlation with the in vivo performances. JT is more efficient in the Russian corpus which is correlated with good performances in the Text and Markup (TM) intrinsic evaluation. Interestingly, the BP–JTA and BP–JTS combinations offer even better results.

On the opposite, the best BP performances are obtained on the Chinese subset whereas the intrinsic evaluation on this dataset has given one of its worst results (Tables 6a). The only corpus where we have a real correlation between the two evaluations is the Polish one (Table 6c). In some cases, the results for Extrinsic Evaluation are even better than those of reference. When precision is better (see red figures in the Precision column of Table 8), it means that there were so many missing parts after WCE that these documents could not be False Positives. Therefore, the DANIEL classifier had no difficulty to classify them in the negative class, some of them were empty. This is a strong bias, the results are improved for bad reasons.

This bias happened only once for Polish. In this particular case, it is both biased from the WCE and from the DANIEL tool: a poor structure extraction for a False Negative made it possible for the system to correctly select the document.

Table 8 gives the best tool for each measure and each evaluation type. In this table "JT", "BP-JT" and "NCT" show that some tools shared the best result. BP is by far the best stand-alone tool but many combinations (BP-JTA for instance) give even better results. BP has the best results for Chinese, Greek and Polish.

As soon as the markup is taken into account in the evaluation process, the gap between BP and JT diminishes (except for the Chinese data). NCLEANER results vary a lot, with respect both to language and evaluation type. The TM evaluation metric is the most consistent with the intrinsic evaluation. This result is not surprising since DANIEL relies on both the content and the text structure.

6 Discussion

In this article, we showed how difficult, but important, the evaluation of Web Content Extraction (WCE) tools is. We compared an intrinsic evaluation scheme, the state-of-the-art CLEANEVAL metrics, and an extrinsic evaluation scheme which measured the influence of the WCE tools on downstream modules. We showed that the intrinsic evaluation gives incorrect insights on the quality of the WCE tools. This is due to the algorithms themselves as well as to more general assumption that the best way to evaluate NLP

[10] Again, we removed NCLEANER results since they were poor in this multilingual setting

Table 8. Best tool for each measure in each configuration.

	Text Only (TO)		
	P	R	F_1
Chinese	BP (61.32)	BP (52.90)	BP (56.80)
English	BP-JTA (86.98)	BP (92.02)	BP (88.89)
Greek	BP-JTA (93.62)	BP (96.48)	BP (94.10)
Polish	BP-JTA (85.24)	JTS-BP (87.54)	BP (84.26)
Russian	BP-JTA (67.77)	JTS-BP (86.81)	BP-JTA (69.92)
All	BP-JTA (85.01)	BP (88.89)	BP (85.20)

	Text and Markup (TM)		
	P	R	F_1
Chinese	JT-BP (89.91)	BP (67.99)	BP (75.34)
English	BP-JTA (81.09)	BP (93.59)	BP-JTS (79.79)
Greek	BP-JTA (86.18)	BP (91.67)	BP-JTS (82.90)
Polish	BP-JTA (76.27)	BP (85.57)	BP (72.75)
Russian	BP-JTA (48.63)	JTS-BP (86.68)	JTA-BP (58.74)
All	BP-JTA (73.30)	BP (85.42)	BP (73.48)

	Character Based (CHA)		
	P	R	F_1
Chinese	BP (77.12)	BP (63.55)	BP (69.68)
English	BP-JTA (87.60)	BP (91.03)	BP (87.87)
Greek	BP (87.58)	BP (91.59)	BP (89.54)
Polish	BP-JTA (82.95)	JTS-BP (82.50)	BP (81.42)
Russian	BP-JTA (53.65)	JTS-BP (79.96)	JTA-BP (59.56)
All	BP-JTA (76.94)	BP (81.12)	BP (78.97)

	Extrinsic Evaluation (EE)		
	P	R	F_1
Chinese	NCT (83.33)	BP (93.75)	BP (85.71)
English	BP-JTS (65.22)	NC (60.00)	NC (59.15)
Greek	BP (85.71)	JTA (76.47)	BP-JTA (68.57)
Polish	NCT (82.35)	BP-JTS (81.48)	BP-JTS (72.13)
Russian	NCNL (100)	BP-JT, JTA (79.31)	JTA (80.70)
All	BP (74.68)	BP-JT (62.10)	BP-JTA (64.98)

modules would be to evaluate independently of a task or a pipeline. We showed that WCE tools obtaining outstanding accuracy through intrinsic evaluation can be much less satisfactory in an extrinsic evaluation scheme. Furthermore, results are not consistent in different languages.

In a more general aspect, we wanted to highlight a scarcely studied drawback of NLP pipelines: how a component of an NLP pipeline may have a bad influence on downstream processing. In other words, how likely it is to provoke cascading errors. Choosing NLP components by relying on an evaluation in laboratory conditions (e.g. with a somewhat ideal input) may lead to unexpected outcomes. It appears to be particularly true for WCE although the importance of this task seems to be underestimated. We can cite here the CLEANEVAL organizers who stated that "Cleaning webpages is a low-level, unglamorous task and yet it is increasingly crucial". With that aspect in mind, NLP pipeline should be evaluated in real conditions: with noisy input data, if applicable. Perfectly cleaned corpora which are very unlikkely to encounter in practice. In that aspect, WCE is not an engineering task but a real NLP task, it should incite the community to conceive systems resilient to noisy input and designed to work not only in laboratory conditions.

References

1. **Baluja, S. (2006).** Browsing on small screens: recasting web-page segmentation into an efficient machine learning framework. *Proceedings of the 15th international conference on World Wide Web*, WWW '06, ACM, New York, NY, USA, pp. 33–42.
2. **Barbaresi, A. (2015).** *Ad hoc and general-purpose corpus construction from web sources*. Ph.D. thesis, École normale supérieure de Lyon, France.
3. **Biemann, C., Bildhauer, F., Evert, S., Goldhahn, D., Quasthoff, U., Schäfer, R., Simon, J., Swiezinski, L., & Zesch, T. (2013).** Scalable construction of high-quality web corpora. *JLCL*, Vol. 28, No. 2, pp. 23–59.
4. **Brixtel, R., Lejeune, G., Doucet, A., & Lucas, N. (2013).** Any Language Early Detection of Epidemic Diseases from Web News Streams. *International Conference on Healthcare Informatics (ICHI)*, pp. 159–168.
5. **Chakrabarti, D., Kumar, R., & Punera, K. (2008).** A graph-theoretic approach to webpage segmentation. *Proceedings of the 17th international conference on World Wide Web*, WWW '08, ACM, New York, NY, USA, pp. 377–386.
6. **Das, S. N., Vijayaraghavan, P. K., & Mathew, M. (2012).** Article: Eliminating noisy information in web pages using featured dom tree. *International Journal of Applied Information Systems*, Vol. 2, No. 2, pp. 27–34. Published by Foundation of Computer Science, New York, USA.
7. **Doucet, A., Kazai, G., & Meunier, J.-L. (2011).** ICDAR 2011 Book Structure Extraction Competition. *Proceedings of the Eleventh International Conference on Document Analysis and Recognition (ICDAR'2011)*, Beijing, China, pp. 1501–1505.
8. **Evert, S. (2008).** A lightweight and efficient tool for cleaning web pages. *Proceedings of LREC 2008*, pp. 3489–3493.

9. **Ferraresi, A., Zanchetta, E., Baroni, M., & Bernardini, S. (2008).** Introducing and evaluating ukwac, a very large web-derived corpus of english. *Proceedings of the 4th Web as Corpus Workshop, LREC 2008*, pp. 47–54.
10. **Fu, L., Meng, Y., Xia, Y., & Yu, H. (2010).** Web content extraction based on webpage layout analysis. *2010 Second International Conference on Information Technology and Computer Science*, pp. 40–43.
11. **Kohlschütter, C., Fankhauser, P., & Nejdl, W. (2010).** Boilerplate detection using shallow text features. *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, ACM, New York, NY, USA, pp. 441–450.
12. **Pasternack, J. & Roth, D. (2009).** Extracting article text from the web with maximum subsequence segmentation. *WWW*, pp. 971–980.
13. **Pomikálek, J. (2011).** Removing boilerplate and duplicate content from web corpora. *Disertacnı práce, Masarykova univerzita, Fakulta informatiky*.
14. **Ratcliff, J. W. & Metzener, D. E. (1988).** Pattern matching: The gestalt approach. *Dr. Dobbs Journal*, Vol. 13, No. 7, pp. 46, 47, 59–51, 68–72.
15. **Spousta, M., Marek, M., & Pecina, P. (2008).** Victor: the Web-Page Cleaning Tool. *Proceedings of the 4th Web as Corpus Workshop, LREC 2008*, pp. 12–17.
16. **Vieira, K., da Silva, A. S., Pinto, N., de Moura, E. S., Cavalcanti, J. a. M. B., & Freire, J. (2006).** A fast and robust method for web page template detection and removal. *ACM international conference on Information and knowledge management*, CIKM '06, ACM, New York, NY, USA, pp. 258–267.

Article received on 09/12/2017; accepted on 15/02/2018.
Corresponding author is Gaël Lejeune.