A new proposal for evaluating Web page Cleaning Tools

Gaël Lejeune[1] and Lichao Zhu[2]

[1] STIH Sorbonne University, Paris
gael.lejeune@paris-sorbonne.fr
[2] LSHS Paris XIII University
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 the intrinsic evaluation results are not consistent with extrinsic evaluation results. We also show that there are important differences in the results between the studied languages. We conclude that choosing a web page cleaning tool should be made in view of the aimed task rather than on the performances observed through an intrinsic evaluation scheme.

1 Introduction

Many NLP research projects take advantage of the huge amount of textual data available online. These data have shown a great impact on the field by widening the range of accessible tasks and available techniques. With more data, it becomes possible to use data-intensive techniques such as textometry or machine learning. However, as evidenced by Biemann[2], it becomes more and more difficult to verify the validity of the data respecting 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 can not be straightforwardly processed neither. The source code contains non-textual (or non-informative) elements which are not required for NLP tasks. Furthermore, this 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 the real world, web browsers tend to interpret the code in order to correct coding errors or to adapt the code to a particular terminal. 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. It extracts textual segments or discards noise (advertisement, templates, code…). Interestingly, this task has received various names, highlighting the different points of view on this task: *boilerplate removal*, *Web Page Template Detection*, Web Page Cleaning, readability web content extraction. [3] pointed out that it can affect corpus statistics in a way that it requires further inspection.

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”. 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. Most of the variations may be found in the page for each category (finance, sport …) and in the main page. As pointed out by the web-designer Andy Ruledge[3], 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 that automating 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 which aim to improve the reading experience using a browser. This problem had been pointed out a few years ago by researchers like [1].

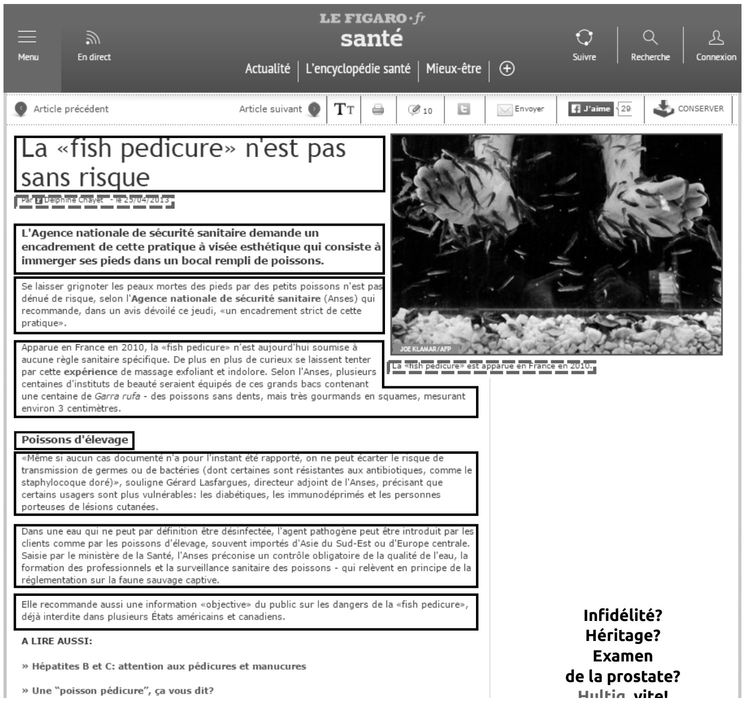
In this article, we focus on press articles for evaluation purposes but we advocate that these issues can be encountered with any type of web harvested data that since most data is not available in RSS related format. This allows us to benefit from available gold standard data in different languages (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[4]:

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…

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

[4] `https://tinyurl.com/lefigaro-fishpedicure`



La «fish pedicure» n'est pas sans risque

Par Delphine Chayet - le 25/04/2013

L'Agence nationale de sécurité sanitaire demande un encadrement de cette pratique à visée esthétique qui consiste à immerger ses pieds dans un bocal rempli de poissons.

Se laisser grignoter les peaux mortes des pieds par des petits poissons n'est pas dénué de risque, selon l'**Agence nationale de sécurité sanitaire** (Anses) qui recommande, dans un avis dévoilé ce jeudi, «un encadrement strict de cette pratique».

La «fish pedicure» est apparue en France en 2010.

Apparue en France en 2010, la «fish pedicure» n'est aujourd'hui soumise à aucune règle sanitaire spécifique. De plus en plus de curieux se laissent tenter par cette **expérience** de massage exfoliant et indolore. Selon l'Anses, plusieurs centaines d'instituts de beauté seraient équipés de ces grands bacs contenant une centaine de *Garra rufa* - des poissons sans dents, mais très gourmands en squames, mesurant environ 3 centimètres.

Poissons d'élevage

«Même si aucun cas documenté n'a pour l'instant été rapporté, on ne peut écarter le risque de transmission de germes ou de bactéries (dont certaines sont résistantes aux antibiotiques, comme le staphylocoque doré)», souligne Gérard Lasfargues, directeur adjoint de l'Anses, précisant que certains usagers sont plus vulnérables: les diabétiques, les immunodéprimés et les personnes porteuses de lésions cutanées.

Dans une eau qui ne peut par définition être désinfectée, l'agent pathogène peut être introduit par les clients comme par les poissons d'élevage, souvent importés d'Asie du Sud-Est ou d'Europe centrale. Saisie par le ministère de la Santé, l'Anses préconise un contrôle obligatoire de la qualité de l'eau, la formation des professionnels et la surveillance sanitaire des poissons - qui relèvent en principe de la réglementation sur la faune sauvage captive.

Elle recommande aussi une information «objective» du public sur les dangers de la «fish pedicure», déjà interdite dans plusieurs États américains et canadiens.

A LIRE AUSSI:

» Hépatites B et C: attention aux pédicures et manucures

» Une "poisson pédicure", ça vous dit?

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

In state-of-the-art techniques, this borderline category is generally considered as non-informative, mostly because it leaves the task as a binary classification one. Some exceptions exist, for instance the gold-standard corpus built for the BOILERPIPE tool [10] uses 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 of the article; (13%);
2. other segments like article date and captions (3%);
3. readers commentaries (1%);
4. related content, links to related articles (4%);
5. segments belong to the non-informative class (79%).

The authors pinpoint that keeping track of the structure (title levels, lists . . .) is of great interest. We can then 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: 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 un encadrement [. . .]
p	Se laisser grignoter les peaux mortes des pieds par des petits poissons [. . .]
p	Apparue en France en 2010, la ” fish pedicure” n’est aujourd’hui [. . .]
h	Poissons d’élevage
p	Même si aucun cas documenté n’a pour l’instant été rapporté,[. . .]
p	Dans une eau qui ne peut par définition être désinfectée,[. . .]
p	Elle recommande aussi une information ” objective” du public [. . .]

Table 1 exhibits an output 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>`.

3 Text Extraction Tools design

3.1 Features for text Extraction

Perhaps the most intuitive way to perform text extraction is to take advantage of the *Document Object Model* (DOM). There is a strong connection between text extraction and web page segmentation like in [5]. For instance, [15] used DOM level similarities between 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 between informative and non-informative contents. There are more shallow strategies to exploit the HTML structure. [9] used HTML tags density, [8] and [14] relied on n-gram models and [11] proposed to combine these two kinds of features. These tools rely on various features that we can group into four categories according to their analysis level:

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

In this study, we focus on 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 [12]. At first, we wanted to include the well-known tool Readability[5], but no official version is available for free.

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

	Website	Rendering	HTML structure	Textual Content
Boilerpipe	_	★	★	★★★
NCleaner	_	★	_	★★★
Justext	_	_	★★	★★★

3.3 Boilerpipe

Boilerpipe[6] combines criteria designed to model the content of the informative segments. The website dimension is not used 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 pages available. The rendering aspect is slightly used, only an estimation of the optimal width of a line (80 characters) is exploited in order to assess if 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.

The main feature for Boilerpipe is the mean length of tokens, defined as character strings without blanks or punctuation. It is combined with local features and contextual features. Capitalized words, links, pipes ("—") mark non-informative content. To 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 consecutivei (and vice-versa). Therefore, the class of a segment is strongly dependent on the class of the previous and the next segment. For this purpose, the rendering is simulated by considering that a segment contains as many lines as it can fill columns of length 80 (considered as the optimal length for a human reader). Each segment has a minimal length of 1. For each segment, the token density is computed by dividing the number of tokens by the number of lines contained by the segment.

3.4 NCleaner

NCleaner[7] uses character *n*-grams language models [8]. NCleaner computes the probability that a given character belongs to the textual content by analyzing its left

[5] https://readability.com/

[6] http://code.google.com/p/boilerpipe/

[7] https://tinyurl.com/cleaneval

context $Pr(c_i|c_1...c_{i-1})$, where c_i is the character a the offset i. The method identifies the *n*-grams (with $1 \leq n \leq 3$) that 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
NonLexical (NCNL): Turns letters in `a` and digits in `0`
Trained (NCTx): Trained with x pairs ($d_{\texttt{raw}}$, $d_{\texttt{clean}}$)

3.5 `Justext`

`Justext` is a freely available tool which can be used via an API[8], its process has two separate steps [12]. 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 classification is performed with the 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 evaluating the content extraction and the correctness of the structure. An evaluation script is available, it 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 [13]. This algorithm matches the longest common subsequences and then applies recursively in the unmatched regions. 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 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.

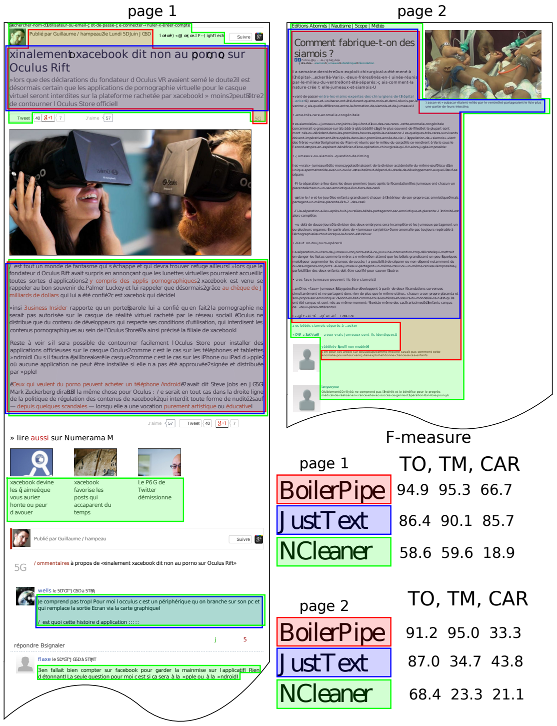
[8] `http://nlp.fi.muni.cz/projects/justext/`

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)	$FalsePositives+ = 4$

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. See 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 human eye, 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 the corpus we choose and propose measures for extrinsic evaluation in order to verify if there is a correlation between intrinsic evaluation results and ”real life” application.

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



page 1	TO	TM	CAR
BoilerPipe	94.9	95.3	66.7
JustText	86.4	90.1	85.7
NCleaner	58.6	59.6	18.9

page 2	TO	TM	CAR
BoilerPipe	91.2	95.0	33.3
JustText	87.0	34.7	43.8
NCleaner	68.4	23.3	21.1

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. The corpus contains documents in five languages (Chinese, English, Greek, Polish and 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. This has not allowed us to perform the *labeled* version of the *text and markup* evaluation. This corpus has been released for evaluating a classification system specialized on epidemic surveillance. The code for the system is available online [9] 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).

Table 4: DANIEL Number of documents in the corpus and proportion of retrieved ones.

	Chinese	English	Greek	Polish	Russian	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.

These figures show that BP outperforms the other tools and the combinations when we evaluate on the complete corpus. At first, we expected that good results will come 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)[10] 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 becomes more relevant.

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

[9] `https://github.com/rundimeco/daniel`

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

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		
Mesures	P	R	F_1	P	R	F_1	P	R	F_1
BP	81.80	**88.89**	**85.20**	76.93	**81.12**	**78.97**	64.47	**85.42**	**73.48**
BP–JTA	85.01	80.20	82.54	76.94	63.86	69.79	73.30	58.74	65.22
BP–JTS	83.23	82.87	83.05	75.12	66.03	70.28	69.22	62.07	65.45
JTA	68.75	83.41	75.37	63.79	67.03	65.37	61.94	63.23	62.58
JTA–BP	72.54	85.86	78.64	69.43	73.28	71.31	66.76	69.34	68.02
JTS	62.68	86.30	72.62	56.93	68.63	62.23	54.24	66.57	59.78
JTS–BP	66.31	88.74	75.90	62.95	75.76	68.77	59.42	72.70	65.39
NC	**98.53**	39.38	56.27	**96.65**	23.15	37.36	**89.01**	30.82	45.78
NCT5	60.43	23.83	34.18	53.81	16.03	24.70	48.41	19.89	28.19
NCT25	56.14	25.70	35.26	53.25	18.73	27.72	45.11	21.77	29.36

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	**61.32**	**52.90**	**56.80**	**77.12**	**63.55**	**69.68**	84.48	**67.99**	**75.34**
BP–JTA	39.60	08.60	14.13	76.38	25.98	38.77	44.40	9.79	16.05
BP–JTS	39.60	08.60	14.13	76.38	25.98	38.77	44.40	9.79	16.05
JTA	23.25	11.72	15.58	71.31	32.05	44.23	49.27	16.78	25.03
JTA–BP	49.64	31.25	38.35	68.96	30.67	42.46	**89.91**	32.71	47.97
JTS	23.25	11.72	15.58	71.31	32.05	44.23	49.27	16.78	25.03
JTS–BP	49.64	31.25	38.35	68.96	30.67	42.46	**89.91**	32.71	47.97

(b) English

	TO			CAR			TM		
	P	R	F_1	P	R	F_1	P	R	F_1
BP	85.97	**92.02**	**88.89**	84.92	**91.03**	**87.87**	69.29	**93.59**	79.63
BP–JTA	**86.98**	82.60	84.73	**87.60**	79.80	83.51	**81.09**	76.02	78.48
BP–JTS	86.36	85.30	85.83	87.25	82.84	84.99	79.44	80.15	**79.79**
JTA	68.41	85.38	75.96	69.98	82.79	75.85	67.17	79.69	72.90
JTA–BP	75.70	88.04	81.41	75.67	87.00	80.94	71.14	82.90	76.57
JTS	66.68	88.20	75.94	68.16	85.97	76.03	63.95	83.94	72.60
JTS–BP	73.78	90.94	81.47	73.83	90.46	81.30	68.30	87.28	76.63

(c) Polish

	TO			CAR			TM		
	P	R	F_1	P	R	F_1	P	R	F_1
BP	83.27	85.28	**84.26**	80.76	82.08	**81.42**	63.27	**85.57**	**72.75**
BP–JTA	**85.24**	78.34	81.64	**82.95**	73.32	77.84	**76.27**	67.74	71.75
BP–JTS	83.77	81.83	82.79	82.01	77.14	79.50	71.71	73.57	72.63
JTA	67.78	82.63	74.47	67.64	78.53	72.68	62.74	73.05	67.51
JTA–BP	68.63	84.02	75.55	66.33	79.42	72.29	61.13	74.61	67.20
JTS	63.23	86.12	72.92	62.56	81.29	70.71	54.10	77.59	63.75
JTS–BP	64.28	**87.54**	74.13	61.84	**82.50**	70.69	53.37	78.92	63.68

(d) Russian

	TO			CAR			TM		
	P	R	F_1	P	R	F_1	P	R	F_1
BP	58.79	79.33	67.53	51.67	70.16	59.51	37.92	85.50	52.54
BP–JTA	**67.77**	72.20	**69.92**	**53.65**	56.30	54.94	**48.63**	62.85	54.83
BP–JTS	61.91	75.22	67.92	48.46	58.53	53.02	40.29	67.11	50.35
JTA	52.72	81.78	64.11	41.28	63.35	49.98	42.94	75.04	54.62
JTA–BP	53.77	83.49	65.41	48.91	76.14	**59.56**	45.63	82.41	**58.74**
JTS	45.48	85.17	59.30	34.54	64.33	44.95	32.21	80.10	45.95
JTS–BP	46.57	**86.81**	60.62	42.56	**79.96**	55.55	34.40	**86.68**	49.25

(e) Greek

	TO			CAR			TM		
	P	R	F_1	P	R	F_1	P	R	F_1
BP	91.84	**96.48**	**94.10**	**87.58**	**91.59**	**89.54**	66.74	**91.67**	77.24
BP–JTA	**93.62**	90.93	92.25	79.98	76.75	78.33	**86.18**	78.85	82.35
BP–JTS	93.33	92.76	93.05	80.04	78.60	79.31	84.01	81.82	**82.90**
JTA	88.10	90.07	89.08	73.39	73.95	73.67	76.65	74.68	75.65
JTA–BP	88.92	91.51	90.20	85.15	87.46	86.29	75.73	76.80	76.26
JTS	70.83	92.57	80.25	59.41	74.87	66.25	62.16	78.40	69.34
JTS–BP	72.53	93.82	81.81	71.15	89.67	79.34	66.92	81.15	73.35

results obtained by DANIEL on the reference corpus are compared to those obtained with the automatically cleaned documents.

In the reference line are mentioned the results obtained with the gold standard texts (manually cleaned). Please 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-

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 same result as the reference is achieved.

	English			Chinese			Greek			Polish			Russian			All Docs		
Mesures	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	60.00	25.71	36.00	78.95	**93.75**	**85.71**	**85.71**	35.94	50.00	76.47	48.15	59.09	76.19	55.17	64.00	**74.68**	47.58	58.13
BP–JTA	61.54	45.71	52.46	71.43	31.25	43.48	66.67	70.59	**68.57**	65.63	77.78	71.19	76.67	**79.31**	77.97	68.14	**62.10**	**64.98**
BP–JTS	**65.22**	42.86	51.72	71.43	31.25	43.48	63.16	70.59	66.67	64.71	**81.48**	**72.13**	74.19	**79.31**	76.67	67.54	**62.10**	64.71
JTA	55.17	45.71	50.00	66.67	37.50	48.00	59.09	**76.47**	66.67	59.26	59.26	59.26	82.14	**79.31**	**80.70**	64.35	59.68	61.92
JTA–BP	59.09	37.14	45.61	66.67	37.50	48.00	62.50	58.82	60.61	65.38	62.96	64.15	76.00	65.52	70.37	66.33	52.42	58.56
JTS	55.56	42.86	48.39	66.67	37.50	48.00	66.67	58.82	62.50	56.67	62.96	59.65	**82.61**	65.52	73.08	64.42	54.03	58.77
JTS–BP	60.87	40.00	48.28	66.67	37.50	48.00	66.67	47.06	55.17	62.96	62.96	62.96	78.26	62.07	69.23	67.02	50.81	57.80
NC	58.33	**60.00**	**59.15**	N/A	0.00	N/A	N/A	0.00	N/A	80.00	14.81	25.00	N/A	0.00	N/A	60.98	20.16	30.30
NCT5	52.94	25.71	34.62	**83.33**	31.25	45.45	N/A	0.00	N/A	**82.35**	51.85	63.64	60.00	20.69	30.77	62.96	27.42	38.20
NCT25	50.00	25.71	33.96	**83.33**	31.25	45.45	20.00	5.88	9.09	**82.35**	51.85	63.64	61.54	27.59	38.09	62.71	29.84	40.44
Reference	**68.89**	**88.57**	**77.50**	**80.00**	**100**	**88.89**	**68.42**	**76.47**	**72.22**	**61.76**	**77.78**	**68.85**	**72.73**	**82.76**	**77.42**	**69.54**	**84.68**	**76.36**

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 one can see in Table 6a. NCLEANER performs globally worse but obtains some good results on English and Polish. In fact, the performances vary a lot between languages. See for instance Tables 6b to 6d[11].

Obviously, all the tools seem to be firstly trained for English corpora. This is probably the main reason for the performance gap between JT and BP, the largest difference at the advantage of the former 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), it means that there were so many missing parts after WCE that these documents could not be False Positives. This is a 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

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

(with the exception of 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.

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)

6 Discussion

In this article, we showed how difficult, but important, is the evaluation of Web Content Extraction (WCE) tools. 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 used as well as to the more general assumption that the best way to evaluate NLP 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 atisfactory 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 is it 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 under-estimated. 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: noisy input data, if applicable, and not ideal, perfectly cleaned, corpora which is unlikely to encounter in real-world applications. WCE is not an engineering task but a real NLP task, it should incite the community to conceive systems resilient to noisy input and which are not designed to work only in laboratory conditions.

References

1. Baluja, S.: Browsing on small screens: recasting web-page segmentation into an efficient machine learning framework. In: Proceedings of the 15th international conference on World Wide Web. pp. 33–42. WWW '06, ACM, New York, NY, USA (2006)
2. Barbaresi, A.: Ad hoc and general-purpose corpus construction from web sources. Ph.D. thesis, École normale supérieure de Lyon, France (2015)
3. Biemann, C., Bildhauer, F., Evert, S., Goldhahn, D., Quasthoff, U., Schäfer, R., Simon, J., Swiezinski, L., Zesch, T.: Scalable construction of high-quality web corpora. JLCL 28(2), 23–59 (2013)
4. Brixtel, R., Lejeune, G., Doucet, A., Lucas, N.: Any Language Early Detection of Epidemic Diseases from Web News Streams. In: International Conference on Healthcare Informatics (ICHI) (2013)
5. Chakrabarti, D., Kumar, R., Punera, K.: A graph-theoretic approach to webpage segmentation. In: Proceedings of the 17th international conference on World Wide Web. pp. 377–386. WWW '08, ACM, New York, NY, USA (2008)
6. Das, S.N., Vijayaraghavan, P.K., Mathew, M.: Article: Eliminating noisy information in web pages using featured dom tree. International Journal of Applied Information Systems 2(2), 27–34 (May 2012), published by Foundation of Computer Science, New York, USA
7. Doucet, A., Kazai, G., Meunier, J.L.: ICDAR 2011 Book Structure Extraction Competition. In: Proceedings of the Eleventh International Conference on Document Analysis and Recognition (ICDAR'2011). pp. 1501–1505. Beijing, China (September 2011)
8. Evert, S.: A lightweight and efficient tool for cleaning web pages. In: Actes du 4ème Workshop Web as Corpus, LREC 2008 (2008)
9. Ferraresi, A., Zanchetta, E., Baroni, M., Bernardini, S.: Introducing and evaluating ukwac, a very large web-derived corpus of english. In: Actes du 4ème Workshop Web as Corpus, LREC 2008 (2008)
10. Kohlschütter, C., Fankhauser, P., Nejdl, W.: Boilerplate detection using shallow text features. In: Proceedings of the third ACM international conference on Web search and data mining. pp. 441–450. WSDM '10, ACM, New York, NY, USA (2010)
11. Pasternack, J., Roth, D.: Extracting article text from the web with maximum subsequence segmentation. In: WWW. pp. 971–980 (2009)
12. Pomikálek, J.: Removing boilerplate and duplicate content from web corpora. Disertacnı práce, Masarykova univerzita, Fakulta informatiky (2011)
13. Ratcliff, J.W., Metzener, D.E.: Pattern matching: The gestalt approach. Dr. Dobbs Journal 13(7), 46, 47, 59–51, 68–72 (Jul 1988)
14. Spousta, M., Marek, M., Pecina, P.: Victor: the Web-Page Cleaning Tool. In: Actes du 4ème Workshop Web as Corpus, LREC 2008 (2008)
15. Vieira, K., da Silva, A.S., Pinto, N., de Moura, E.S., Cavalcanti, J.a.M.B., Freire, J.: A fast and robust method for web page template detection and removal. In: ACM international conference on Information and knowledge management. pp. 258–267. CIKM '06, ACM, New York, NY, USA (2006)