Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

Bien choisir son outil d'extraction de contenu à partir du Web

Gaël Lejeune[1] Adrien Barbaresi[2]
(1) Sorbonne Université, 1 rue Victor Cousin, 75005 Paris, France
(2) Académie des Sciences de Berlin-Brandenbourg, Jägerstraße 22-23, 10117 Berlin, Allemagne

RÉSUMÉ

Nous proposons une démonstration sur l'extraction de contenu textuel dans des pages web ainsi que son évaluation. Nous nous concentrons sur les pages web contenant du texte (articles de presse, magazines en ligne et blogs) et montrons que les textes peuvent varier grandement selon différentes dimensions : diachronique, géographique et typologique. Dès lors, les outils et mesures d'évaluation correspondantes sont sujettes à caution : les indicateurs communément utilisés et censés présider au choix de l'outil approprié par les utilisateurs finaux sont à la fois imprécis et difficiles à interpréter.

ABSTRACT

Choosing the appropriate tool for Web Content Extraction

This demonstration focuses on the use and evaluation of Web Content Extraction tools, with a focus on web pages containing text (news articles, magazines and blogs). We show that the texts may differ with respect to diachronic, geographic and typological factors, so that state-of-the-art tools and measures are altogether imprecise and difficult to interpret.

MOTS-CLÉS : Construction de Corpus, Extraction de Contenu, Nettoyage de Pages Web, Cleaneval.

KEYWORDS: Web corpus construction, Web Content Extraction, Boilerplate removal, Cleaneval.

La construction de corpus à partir du Web comprend des opérations allant de la découverte et du téléchargement des sources jusqu'à l'annotation et l'indexation dans des bases de données (Kilgarriff, 2007). Bien que cette construction soit devenue un élément commun des chaînes de traitement de TAL, les détails techniques concernant la mise en œuvre du nettoyage et de la préparation des pages web sont souvent omis. En définitive, chercheurs et utilisateurs s'en trouvent réduits à effectuer des évaluations a posteriori de la qualité et de l'adéquation des données (Baroni *et al.*, 2009). Le processus d'extraction de contenu peut être résumé de la façon suivante : étant donné le code source d'une page web, il s'agit d'extraire le contenu textuel utile et d'identifier les méta-données. Concrètement, cette tâche consiste notamment à écarter ce qui est de l'ordre du squelette de page web et implique une conversion du format HTML à un format texte ou XML. La disponibilité croissante d'outils génériques, par opposition à des approches ad hoc centrées sur l'application de règles d'extraction spécifiques à un site web particulier, a fait progressivement glisser cette tâche au rang d'outil d'ingénierie, alors même qu'elle a un impact direct sur les résultats scientifiques. D'autres approches, exploitant par exemple CommonCrawl [1] s'appuient sur une externalisation de la phase de *crawling* et l'extraction de contenu qui en découle (Habernal *et al.*, 2016). Nous laisserons de côté ici la question du choix des sources proprement dites pour nous concentrer sur les résultats de l'extraction, qui sont en eux-mêmes la base de décisions quant à l'inclusion d'un document donné dans le corpus final (Schäfer *et al.*, 2013). Peut-on réellement laisser de côté la question de ce que l'on a intégré dans des corpus et de l'impact sur les modèles qui vont en être extraits ? La nécessité pour

1. https ://commoncrawl.org

les approches de type apprentissage profond de disposer de grandes quantités de données a forcément conduit à une plus grande légèreté sur la qualité ou la représentativité des données alors même que le besoin de recourir à une analyse fine existe toujours (Geyken *et al.*, 2017).

Nous montrons dans cette démonstration l'impact de différentes méthodes et outils d'extraction de contenus à partir du Web. En raison des performances affichées par les outils, a priori satisfaisantes et en progrès constants, l'impact du choix de l'outil sur la qualité peut être mésestimé par les utilisateurs finaux. Le fait qu'il y ait autant d'outils disponibles est en réalité un indicateur de la disparité dans la qualité des résultats obtenus. En ce sens, les métriques d'évaluation « état de l'art » proposent des scores agrégés qui laissent volontiers de côté des dimensions cruciales au profit d'une approche très générique et anglo-centrée (Barbaresi & Lejeune, 2020) et masquent trois dimensions :

1. Linguistique : les résultats sont extrêmement variables selon les pays d'origine et les langues
2. Typologique, quant à la nature et la forme des sites Web : les manières de construire une page web étant très variées, aucune garantie n'existe sur la robustesse des outils sur ce point
3. Diachronique : le langage du web évolue, comme en témoignent les standards et recommandations, si bien que les outils d'hier ne sont pas toujours adaptés aux pages d'aujourd'hui (Weninger *et al.*, 2016)

La dimension de la langue a souvent été escamotée alors que les (rares) expériences sur le sujet montrent une variabilité importante des résultats (Lejeune & Zhu, 2018). La dimension typologique a un impact en termes d'évaluation des besoins de l'utilisateur final : cherche-t-on un outil « tout terrain », efficace sur des sources variées, ou un outil « hi-fi », fiable sur les sources les plus fréquemment rencontrées ? Enfin, peut-on se fier aux résultats d'un outil sur des données de l'année X pour prédire son efficacité sur des données de l'année X+1 ? Cette dimension ne nous semble pas avoir obtenu l'attention qu'elle mérite dans la littérature sur les corpus web. Les « bonnes » propriétés qui assuraient de bons résultats à un temps T sont-elles conservées au fil de l'adaptation d'un outil aux données nouvelles ? La rétro-comparabilité des données [2] est-elle assurée ? Nous présentons ici des outils d'extraction de contenu parmi les plus utilisés avec un éclairage sur leurs performances, avec pour référence l'environnement Python, très présent si ce n'est majoritaire dans le monde de la recherche [3]. Certains outils étant adaptés d'autres langages, notre comparatif permet d'offrir un large tour d'horizon. Dans le tableau 1 nous reprenons la catégorisation de (Barbaresi & Lejeune, 2020).

Cat.	Outil	Version	Adresse Github	Référence
I	HTML2TEXT	2020.1.16	Alir3z4/html2text/	
I	INSCRIPTIS	1.0	weblyzard/inscriptis	
II	NEWSPAPER3K	0.2.8	codelucas/newspaper	
II	NEWS-PLEASE	1.4.25	fhamborg/news-please	(Hamborg *et al.*, 2017)
II	READABILITY	0.7.1	buriy/python-readability	
III	BOILERPY3	1.0.2	jmriebold/BoilerPy3	(Kohlschütter *et al.*, 2010)
III	DRAGNET	2.0.4	dragnet-org/dragnet	(Peters & Lecocq, 2013)
III	GOOSE3	3.1.6	goose3/goose3	
III	JUSTEXT	2.2.0	miso-belica/jusText	(Pomikálek, 2011)
III	TRAFILATURA	0.4.1	adbar/trafilatura	(Barbaresi, 2019)

TABLE 1: Outils orientés rappel (I), orientés lisibilité(II), spécifiquement dédiés à la tâche (III)

2. Par analogie avec la rétro-compatibilité.
3. https ://spectrum.ieee.org/computing/software/the-top-programming-languages-2019

	multi	el	en	pl	ru	zh	Tps (sec.)	Diff/réf
TRAFILATURA_FB	**75,69**	81,29	84,30	75,13	68,51	69,22	109,9	x5,6
READABILITY	74,62	84,81	85,24	76,08	**71,79**	55,2	56,8	x2,9
BOILERPY3_ART	72,73	63,06	82,24	**80,41**	63,02	**74,91**	39,8	x2,0
JUSTEXT	63,7	86,55	80,33	78,97	70,47	2,18	322,0	x16,3
JUSTEXT_LANGID	63,31	**86,74**	79,26	78,56	69,83	2,18	112,6	x5,7
DRAGNET	58,21	34,0	86,04	72,81	43,31	54,89	24,0	x1,2
NEWSPLEASE	48,83	47,93	86,71	78,05	27,26	4,22	3755,6	x190
INSCRIPTIS	40,10	48,48	43,37	40,1	29,6	38,94	**19,7**	x1
GOOSE	37,87	2,45	88,6	68,26	27,0	3,06	191,3	x9,7
NEWSPAPER	32,37	3,74	**89,3**	63,34	3,37	2,11	105,5	x5,5
HTML2TEXT	31,2	38,43	41,64	32,27	26,37	17,31	71,0	x3,6
JUSTEXT_EN	17,63	2,11	79,26	1,71	2,96	2,11	41,5	x2,1

TABLE 2: Résultats sur le corpus DANIEL, Macro-moyenne des F-mesures sur la classe langue, et f-mesure par langue (mesure CLEANEVAL), les temps de calcul sont une moyenne sur 5 tests et sont exprimés en secondes, le ratio de vitesse est calculé par rapport à l'outil le plus rapide (INSCRIPTIS)

Les comparaisons détaillées de ces outils sont accessibles dans (Barbaresi & Lejeune, 2020) et sont reprises sur un dépôt GITHUB dédié [4] qui doit permettre non seulement la reproductibilité de l'évaluation mais également son suivi, certains outils étant encore activement en développement. Nous reprenons dans le tableau 2 une partie des résultats afin de montrer à quel point les outils peuvent avoir des comportements très variables en termes d'efficacité générale, en termes d'efficacité par langue et enfin en termes de temps de traitement, ce qui n'est pas négligeable. L'évaluation présentée ici utilise la F-mesure obtenue avec les mesures de CLEANEVAL (Baroni *et al.*, 2008), elle a été menée sur le corpus DANIEL (Lejeune *et al.*, 2012; Lejeune & Zhu, 2018) qui comporte un peu plus de 1600 documents en 5 langues avec leur version HTML et leur version nettoyée manuellement.

Nous pouvons voir que les différences de résultats entre les outils sont significatives et que la qualité des résultats obtenus sur l'anglais ne préjuge absolument pas de la consistance des résultats concernant d'autres langues. Pour illustrer ce point nous avons fait figurer dans le tableau le résultat du modèle JUSTEXT sur l'anglais. À la vue de ce comparatif, il ne semble pas exister de solution « tout terrain » même si l'on observe un certain parallélisme entre l'anglais et le polonais. En outre, la vitesse n'est pas une dimension à négliger puisque si l'on compare au système le plus rapide (INSCRIPTIS) on observe des changements d'ordre de grandeur crucial pour l'abord de données massives : avec GOOSE, le résultat est presque 10 fois plus lent, plus de 15 fois plus lent pour le modèle indépendant de la langue de JUSTEXT. Ce dernier cas est intéressant puisque ce modèle indépendant est légèrement plus efficace que le modèle `langid` mais au prix d'un temps de traitement 3 fois plus grand. Enfin, NEWSPLEASE est un cas particulier puisque c'est un outil qui fait bien d'autres choses que le nettoyage.

Bien choisir son outil d'extraction de contenu passe donc non seulement par le choix et la configuration des outils en fonction des sources, mais aussi par des comparatifs et des observations, même à échelle réduite, qui permettent d'exploiter les différences tant sur l'efficacité des extracteurs que sur les temps de traitement.

4. https://github.com/rundimeco/waddle

Références

BARBARESI A. (2019). Generic Web Content Extraction with Open-Source Software. In *Proceedings of KONVENS 2019, Kaleidoscope Abstracts*, p. 267–268 : GSCL.

BARBARESI A. & LEJEUNE G. (2020). Out-of-the-Box and Into the Ditch ? Multilingual Evaluation of Generic Text Extraction Tools. In *Proceedings of the 12th Web as Corpus workshop (WAC-XII)* : ELRA. à paraître.

BARONI M., BERNARDINI S., FERRARESI A. & ZANCHETTA E. (2009). The WaCky Wide Web : a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, **43**(3), 209–226.

BARONI M., CHANTREE F., KILGARRIFF A. & SHAROFF S. (2008). Cleaneval : a Competition for Cleaning Web Pages. In *Proceedings of LREC*, p. 638–643 : ELRA.

GEYKEN A., BARBARESI A., DIDAKOWSKI J., JURISH B., WIEGAND F. & LEMNITZER L. (2017). Die Korpusplattform des "Digitalen Wörterbuchs der deutschen Sprache" (DWDS). *Zeitschrift für germanistische Linguistik*, **45**(2), 327–344.

HABERNAL I., ZAYED O. & 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)*, p. 914–922.

HAMBORG F., MEUSCHKE N., BREITINGER C. & GIPP B. (2017). news-please : A generic news crawler and extractor. In M. GAEDE, V. TRKULJA & V. PETRA, Éds., *Proceedings of the 15th International Symposium of Information Science*, p. 218–223.

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

KOHLSCHÜTTER C., FANKHAUSER P. & 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*, p. 441–450.

LEJEUNE G., BRIXTEL R., DOUCET A. & LUCAS N. (2012). Daniel : Language independent character-based news surveillance. In *International Conference on NLP*, p. 64–75 : Springer.

LEJEUNE G. & ZHU L. (2018). A New Proposal for Evaluating Web Page Cleaning Tools. *Computación y Sistemas*, **22**(4).

PETERS M. E. & LECOCQ D. (2013). Content extraction using diverse feature sets. In *Proceedings of the 22nd International Conference on World Wide Web*, p. 89–90.

POMIKÁLEK J. (2011). *Removing boilerplate and duplicate content from web corpora*. Thèse de doctorat, Masaryk University.

SCHÄFER R., BARBARESI A. & 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*, p. 7–15.

WENINGER T., PALACIOS R., CRESCENZI V., GOTTRON T. & MERIALDO P. (2016). Web Content Extraction – a Meta-Analysis of its Past and Thoughts on its Future. *ACM SIGKDD Explorations Newsletter*, **17**(2), 17–23.