WebT-IDC: A Web Tool for Intelligent Dataset Creation A Use Case for Forums and Blogs

Boudabous Maroua, Anna Pappa

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2021) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

WebT-IDC : A Web Tool for Intelligent Dataset Creation A Use Case for Forums and Blogs

Boudabous Maroua[a,*], Pappa Anna[b]

[a] *University Paris 8, 2 rue de la Liberté, Saint-Denis cedex 93526, France*
[b] *University Paris 8, 2 rue de la Liberté, Saint-Denis cedex 93526, France*

Abstract

This paper presents WebT-IDC, an end-to-end *Web Tool for Intelligent Dataset Creation*, able to construct noiseless corpora of feedback on different topics in any language, from web forums and blogs. The method is based on a unique extraction model of pagination element pattern, independent of DOM structure. WebT-IDC is a holistic tool covering all stages, from user's query, crawling and scraping web pages and extracting relevant data, to building text corpora, free from boilerplate and repetitive data, useful in machine learning tasks. The output is a partially labeled dataset, accurate to reflecting a realistic vision of the market, tested in a Transformer based model in order to show the relevance for immediate use in ML task, offering high accuracy in polarity, language and product recognition. The system is evaluated in terms of noise filtering efficiency and computing time as well as precision and recall.

Keywords: Web content mining; Noiseless corpora creation; Web corpora.

1. Introduction

The World Wide Web is considered as a massive depository of all kind of constantly increasing data (text, image, video, sound), widely used as an inexhaustible source for corpora creation by mining it [2], [11].

Before highlight the ever-increasing need for creating and using all kinds of corpora and datasets, it is necessary to provide precision about the use of 'Corpus' and 'Dataset' terms. While both are similar considering their aspect of representative samples of language production, often both providing information of annotations, we may slightly

* Boudabous Maroua. Tel.: +3-362-044-4780.
E-mail address: mboudabous@ai.univ-paris8.fr

differentiate 'corpus' as a collection of gathered text and 'dataset' as a structured collection of textual data with attributes and values.

Concerning linguistic corpus, researchers spent time to analyze and evaluate *"the web as a corpus"* [12], [16], but this notion lately has changed, including flexibility and customization by crawling more controlled parts of the web [6]. There were controversial statements about the theoretical approach of *"what a corpus is or could be"* besides a big collection of texts in a machine-readable format when questions entered the debate about representativeness, size, sampling, balance, design and purpose. The applied requirements to call a text data collection a corpus are the size and the representativeness. We do not consider the balance because of multilingualism. Some languages are less present than others on the web. [12] argued about the practical aspect with the question *"Is corpus x good for task y?"* and encouraged the exploitation of the web potential.

For the WebT-IDC, the term 'dataset' includes the further use for specific research in machine learning tasks, as 'training' and 'test' datasets.

As for the increasing need of corpora and datasets let us remind that they are the main material of many AI projects that rely on data, using Machine Learning (ML) techniques in all aspects of teaching the machines the ability to perform complex natural language operations. Since datasets are at the core of these projects, we constantly need to create large and high-quality text collections composed of quick, flexible specialized and customized multilingual data from the web [9]. Even a search engine was created (https://datasetsearch.research.google.com/) to help unravel all existing datasets in the web, without specifying whether the use of the dataset is free or not. [1].

However, not many languages have corpora made from web resources such as forums and blogs, and even less have corpora made only by reviews and opinions, written in any language, about a specific product or service available anywhere in the world. The potentiality of further treatment for such kind of context accurate enough to reflect realistic vision of the market (about products and services) is limitless.

Despite the existence of some datasets, we still need to create large text datasets for training. Problems related to topic data scarcity could prevent some AI projects from being accomplished. Furthermore, gathering data is not enough; filtering boilerplate and repetitive data often requires a large-scale cleaning, an essential step of managing and analyzing data, as well as classifying and labelling them which is time-consuming. That is why datasets are still manually created and it is a challenge to generate and maintain them.

Since WebT-IDC creates automatically noiseless corpus and datasets, it is essential to mention the difficulties of such an operation. Creating noiseless web corpora implies specific tools for multilingual resources [5]. This task is always relevant as long as multiple steps are involved for preprocessing textual data [12]. Among them, cleaning is crucial for all layers of linguistic and machine learning processing and remains an essential task in data preprocessing by filling in missing values, smoothing noisy data, identifying or removing outliers, resolving inconsistencies [20].

We present WebT-IDC, an intelligent web tool able to crawl web pages and extract relevant information about users reviews and feedbacks in order to build a noiseless thematic dataset. The method used is based on a unique language independent extraction model for forums DOM structures containing pagination architecture. The performance of WebT-IDC computation time is due to multithreading parallel processing and direct access to the relevant page's regions.

This paper is organized as follows: Section 2 gives an overview of existing scraping frameworks and tools. Section 3 describes the system in details. Section 4 presents the results of our web tool. At last, Section 5 concludes our work and proposes perspectives for future work.

2. Web Content Mining related works

Over the last years, the world wide web has become a huge source of available data, and many research efforts have been made to take advantage of this wealth in several domains. We distinguish three subcategories for web mining: web content mining [8], web structure mining and web usage mining [23]. Also known as Web Log Mining, Web Usage Mining aims to detect behavioural patterns of web users focusing on web server log files. Web Structure Mining analyses the web network structure by studying hyperlinks and navigational relationships between web pages.

[1] https://www.blog.google/products/search/discovering-millions-datasets-web/

Finally, Web Content Mining (WCM) focuses on the web page's content and helps extract needful data such as text, image or audio files in a ready-to-use format based on users queries.

Web corpora creation belongs to the Web Content Mining (WCM) category. It is commonly based on web scraping technique, which provides specifically coded programs to extract data relevant to users requests and transform them into structured or semi-structured formats [14]. Among the state-of-art frameworks used for web scraping, we cite Apache Nutch, Bootcat, JusText and Scrapy. We briefly present them below.

Apache Nutch [10], is an open-source framework written in JAVA. It has four main components: a web page crawler, a web content fetcher, a web page indexer used to avoid duplicate URL among users requests and a data storing processor. These components implement a multithreading paradigm.

Bootcat [1] is a Perl open-source toolkit helping to build thematic corpora based on the search engine's results and an input list of seed terms describing what user needs. This setting step is not trivial, making it time-consuming since the user should respect a set of predefined rules.

Justext [19] is a python library that works on both text and HTML contents. It is designed to remove boilerplate data by defining a list of stop words and a list of most probable boilerplate tags using a clustering algorithm.

Scrapy [18]: is an open-source Python framework for large scale web scraping. It provides a set of libraries to both crawl and extract data from a web page. This framework is actively updated and helps to extract data from websites. It can also process and store them in the preferred structure and format.

Nevertheless, these frameworks and toolkits do not ensure noiseless extraction nor a completely automatic extraction process.

Moreover, based on web scraping techniques and available libraries, multiple web content mining tools have been created to ease the data gathering process, according to users needs and queries. Among the commonly used scraping tools, we cite :

Screen-scraper is a web content mining tool that interfaces with SQL database server. It offers a graphical interface to ask the user to provide the URL to mine, target data elements and define scraping logic implemented either in Java, .net or PHP. The processing is then running concurrently on multiple websites.[13]

Web Info Extractor is a tool for extracting both structured and unstructured data from web pages data into a local file or a database. It keeps traces of visited web pages adding new content once the page is updated. It deals with all languages and supports multi-task and recursive processing.

Mozenda [17] is another web data extraction tool where users can set up agents to process URLs. It provides a web console to view and organize results then export and publish them. The main drawback is that Mozenda's agent component can run only on the Windows platform.

Web Content Extractor is another data extraction tool for Web scraping. It allows users to mine various websites such as online stores, online auctions, shopping sites, real estate sites, financial sites and business directories. It stores the extracted data in many formats, including Microsoft Excel (CSV), Access, TXT, HTML, XML, SQL script, MySQL script, and any ODBC data source.

Eliminating noise from constructed corpora is considered one of the essential steps when implementing a WCM tool. Authors in [15] proposed a data extraction tool that detects target data section using different string distance measurements such as Levenstein and edit distance. [21] employed a sequence-labelling approach based on Conditional Random Fields (CRF) to classify each page section into either "main content" or "noisy segment" classes. [7] has used the DOM tree tags and attributes to feed data input to a backpropagation artificial neural network (ANN) to cluster web page part into noisy and relevant sections.

These tools, unlike WebT-IDC, do not ensure the entire process chain. Besides, they are based on a semi-automatic extraction process requiring human intervention to identify relevant regions through a sophisticated user interface. Thus, they become time-consuming when applied to large dataset creation.

In the next section, we describe the two main components implemented by WebT-IDC : the web crawling and filter component and the web scraping and corpus creation component.

3. System description

To meet the urging need of large multilingual corpora ready-to-use in machine learning tasks, we developed a web tool able to construct a noiseless textual corpus composed of reviews and opinions about products and services posted in forums and blogs websites.

The design of WebT-IDC is user query driven, without any knowledge about the web page architecture or the language used to express the reviews. It is composed by two modules ensuring, respectively, web crawling with filtering and web scraping with corpus creation component. The former searches for web pages over the world wide web that matches best the user query via search engines and filters boilerplate and repetitive data. The latter extracts users' reviews text and complementary information such as date of the post, location, ranking rate and saves them into semi-structured output files ready to feed machine learning tasks. Figure 2 details the WebT-IDC tool's algorithm.

3.1. The crawling and filter component

As shown in figure 1, the crawler and filter component takes as inputs the user query q and a topic t. It starts by browsing over the net to collect seed web pages relevant to the user's need via the method *crawl_for_query()* that uses GoogleScraper [22] tool, an open-source tool extracting all found links. Then, it validates the collected list of URLs using a filter component removing duplicate HTTP links (*validate_remove_duplicates()*), and consequently avoiding extracting the same data multiple times within the final output dataset. At the end of this step, we obtain a clean list of seed URLs ready to be processed by the scraping and corpus creation component described in the next section.

Algorithm 1 WebT-IDC : *web crawling and filter component*

Require: user query q, topic t
Ensure: A list of matching URLs
  candidate_urls = crawl_for_query(q, t)
  list_urls = validate_remove_duplicates(*candidate_urls*)
  for all *url* in *list_urls* **do**
    scrap_url(*url* , t)
  end for

Fig. 1. **WebT-IDC crawling Algorithm**

3.2. The scraping and corpus creation component

In order to provide robust and adaptable scraping processing (for different type of web pages), we first performed an analysis of the forums DOM structures. We got in-depth information about how data is organized. The common element that allows robustness is the presence of the pagination element. Then, we implemented the data extraction component, starting by targeting relevant data regions within the web page until the dataset generation. The following section details this process.

3.2.1. Analysis of web page architecture

This inspection step aims to find out how web pages with reviews and comments are presented and therefore to obtain general information in order to create an adaptable web scraping component. Subsequently, we can structurally separate a given web site content into relevant and noisy data sections. We classify as "noisy content" advertisements and HTML formatting tags. However, we only classify as relevant section the parts of the page that match the user's expectations and needs. The relevant section in forums and blogs is the one that gathers users' reviews and answers. Figure 3 delimits the relevant page section we want to export to a website example for beauty products.

The inspection step is mainly based on the DOM (Document Object Model) tree describing any web page. In the World Wide Web, each URL is considered a document described by a hierarchical tree-shaped object model representing how the information of the internal page is organized into several elementary components with their

Algorithm 2 WebT-IDC : *web scraping component*

Require: An url page u, topic t , file f
Ensure: A thematic csv file dataset
  xq = construct_xpath_pagination_bar()
  pag_urls = direct_access_pagination_urls()
  for all url in pag_urls **do**
    $relevant\_part$ = target_relevant_content(xq)
    $review\_block\_pattern$ = extra_info_tags_recognition($relevant\_part$)
    xq_inf = prepare_xpath($review\_block\_pattern$)
    extract_data_into_file(xq_inf, f)
  end for

Fig. 2. **WebT-IDC Scraping Algorithm**

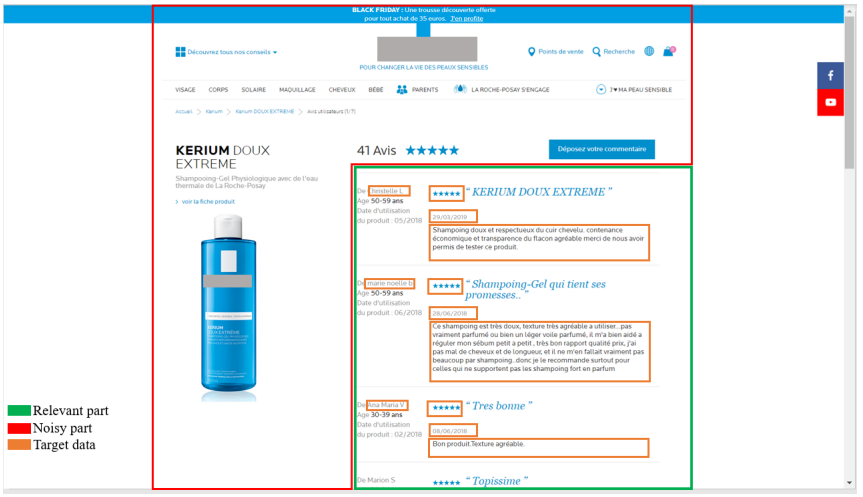


Fig. 3. Relevant section vs Noisy section

positioning constraints. Each component is associated to a style sheet class (CSS) applying visual and rendering features.

This analysis step revealed that the pagination HTML element is the central component used in website reviews to make users' experience smoother and more user-friendly. The pagination element allows the organization of the main content, i.e. opinions and responses, into several subgroups of n elements distributed over x pagination links. The *pagination* element is the main component in our scraping algorithm since it ensures direct access to the most likely relevant nodes in the DOM tree without any further time-consuming sequential analysis algorithms.

Let denote $\boldsymbol{D}$ the depth of the *pagination bar* node that means the distance between the node and the root element of the DOM tree. The relevant data depth is most likely to match either the same depth level as the *pagination element* node, or the n previous depth where $0 < n < 2$.

Direct access to the relevant target section considerably reduces the scraping execution time since it avoids browsing all the elements of the DOM tree.

The scraping process extracts from the relevant page section the review text and the information related to the reviewer's ID, evaluation score, publishing date, and location when they are provided. These additional features are useful for training dataset annotation in machine learning tasks, such as predicting temporal and geographic trends, positive and negative scores. In our algorithm, the presence of theses features is detected by using structural pattern matching elements, detailed in section 3.2.2.

3.2.2. *Defining the scraping logic*

The scraping processing is mainly based on both the DOM tree elements (tags) and style sheet classes. It is written in Python programming language using partly the previously introduced Scrapy framework [18] and XPath querying language [3] to selecting nodes in semi-structured files. XPath simplifies the direct access to the target nodes by introducing three concepts: the navigation axis, the node's test and the predicates. It is worth noting that Scrapy represents only a part of the developing framework while the logic and the extraction rules are implemented by our algorithm.

3.2.3. *Direct Access to the most relevant part of a given input web page*

In the rest of this paper, the pagination element refers to an HTML element that can be defined by either an or <nav/> or <a/> or <button/> node having at least a style sheet class that entirely or partially matches the lemma of the words 'page' and 'more'. This assumption comes as a result of our preliminary analysis. We note that web pages with no pagination element are not managed by our tool.

In fact, this function is complex and plays a crucial role in the extraction process since it helps considering only the part of the page relevant to each use case. It employs the created navigational query to browse the page's DOM architecture. It directly points out the DOM subtree referred most likely to a list of review blocks containing all kinds of needed information.

3.2.4. *Complementary data recognition and extraction*

We implement a pattern matching function to make complementary information extraction more adaptable and generic. The function *extra_info_tags_recognition()* runs as follows. First, we consider as input the previously targeted HTML part. Then we convert it into a string format by concatenating its composing tags along with their corresponding style sheet class, omitting the root tag. Next, we seek recurring path patterns to locate unitary review block (the review text, date, evaluation, and location). Each element of the extracted complementary information is distinguished by a predefined list of frequently used style sheet classes, deduced from the prior inspection step. Since the commercial web sites, forums and blogs, present reviews via multiple hypertext links, WebT-IDC automatically manages the pagination URLs and applies the above-defined algorithm on each of them. Figure 4 illustrates the pattern matching process applied to a page web having the HTML architecture as presented previously in figure 3.

Fig. 4. Complementary information pattern recognition

3.2.5. Thematic multilingual corpus/dataset creation

To make the collected reviews valuable information for further machine learning (or other NLP) tasks, we store them into separate semi-structured CSV files grouped by theme and language. Therefore, each file represents a topic about either a product or a service expressed in a different language. Each line presents the user's feedback described by the comment's text, the user name, the publication date and location when provided, and some extra annotation such as the corresponding theme and the language in which the comment is expressed.

In order to keep our scraper efficient in term of execution time, we applied the multi-threading paradigm on two levels:

- *Seed URLs mining*: The scraping component takes as input the flat file, containing a list of relevant URLs after the user's query, which is the output from the previous crawling and filtering step. These web pages are parallel parsed using multiple python threads respecting a predefined degree of parallelism n. By default, the parameter n is set to the number of input URLs. However, it can be freely configured to fit the hardware settings.
- *Pagination URLs*: For the forums and blogs web pages using HTML pagination mechanism, we defined a degree of parallelism p in order to deal with several pagination URLs simultaneously.

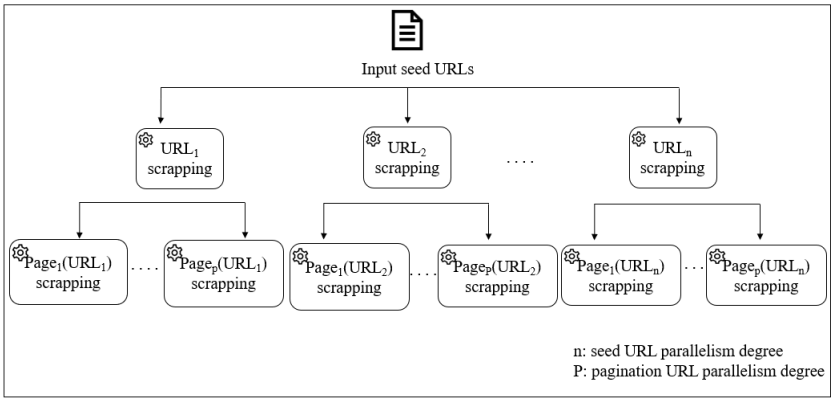


Fig. 5. WebT-IDC parallel processing

Figure 5 shows the parallel processing applied to optimize the execution time of the scraping process. The two degrees of parallelism depend on the performance of the electronic devices, such as the hard drive and memory. To ease parallelism degrees tuning, we decided to fix p as reference to the number of pagination for a given URL.

4. Experiments and results

4.1. Experiments Setting

To evaluate the algorithm performance in terms of adaptability, execution time and extraction noiselessness, we run multiple rounds of experiments, each time with a different specific theme. Hereafter, we denote by theme either a product (smartphone, beauty cream) or a service (airlines services, delivery services). As input data, we only have to provide the user's search query, including the topic information. For the tests, we set up the crawling component (this feature is adaptable) to collect the first ten (different) URLs returned by the search engine, considering that they include the most relevant web pages corresponding to the given query.

The crawled web URLs were structured into 25 different HTML architectures and expressed in different languages (French, English, Spanish, Italian, German, Russian and Greek).

All the experiments were run on a dual-core laptop with 8Go RAM. We tuned the seed URL parallelism degree to be equal to the crawled output list size for every experiment run. However, given a web page, the parallelism degree, denoted by p, is dynamic and fits the number of pagination URLs.

As mentioned earlier, our tool is designed to be multilingual and adaptable to whatever web page architecture, regardless of the language and structure dimension. It is automatic in all processing steps, so there is no need to set up a specific parameter to determine the language used in a given input URL page. It is automatically deducted from the URL's parameters and structure.

4.2. Experiments and Results Evaluation

We evaluated the performance of WebT-IDC using the precision and recall metrics, as well as the execution time.

Table 1 represents the comparison between *WebT-IDC* and three scraping tools that do not provide the crawling feature. Whereas table 2 compares the presented tool to existing tools build upon frameworks that allow both crawling and scraping.

All the tested tools are implemented with a semi-automated extraction approach that requires human expertise to identify the relevant data nodes.

This selection process must be adjusted each time following the inherent architecture of the web page changes.

As these tools do not provide the crawling mechanism, we used our model to fix the input seed URLs to scrap. The results show that WebT-IDC can achieve 1OO% precision with a fully automatic extraction process. Execution time performance is also competitive.

Table 1. WebT-IDC performance and feature evaluation vs existing scrapers.

Scrapping Tool	Features Crawling	Scraping	Adaptability	Performance Precision (%)	Excecution time (ms)
WebExtractor	No	semi-automatic	No	100	135
Mozenda	No	semi-automatic	No	100	72
Webscraper.io	No	semi-automatic	No	100	**62**
WebT-IDC	**Yes**	**automatic**	**Yes**	**100**	**67**

In table 2, we evaluated the quality of the scraped data comparing to the experiment presented in [15] since the tool is not available for new tests. So, we repeated the same experiment by testing twenty web pages from the domain "ubuntu-fr.org". WebT-IDC outperforms all the considered tools. It gives a noiseless dataset with 1OO% precision and recall and obtains the best execution time. For this test the crawling mechanism was disabled for WebT-IDC since the input seed URLs were fixed.

Table 2. WebT-IDC performance evaluation using DyCorC experiment setting [15]

	Nutch	Heritrix	BootCaT	DyCorC	**WebT-IDC**
Precision	62,68%	**100,00%**	33,46%	**100,00%**	**100,00%**
Recall	89,79%	56,58%	98,65%	96,97%	**100%**
Crawling Time (ms)	143	909	70	70	**15**
Scraping Time (ms)				107	**25**

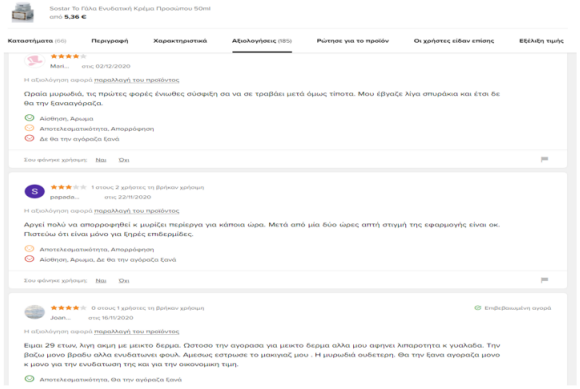
In another example of WebT-IDC performance, with a topic such as "airlines services", we obtained a global number of 44348 reviews dispatched as follows: 17083 comments are expressed in French, 16668 in English, 2240 in Spanish, 3300 in Italian and 5057 in German. The variance observed between languages could be partly explained by the fact that the number of URLs per language is not evenly distributed on the World Wide Web, particularly for the websites of critics who are highly dependent on different cultures.

Finally, to evaluate WebT-IDC's ability to eliminate the noise data during the extraction process, we use a purity measure p that computes the number of reviews containing noisy data proportionally to the total number of extracted reviews as expressed in equation 1.

$$p = \frac{Number \quad of \quad reviews \quad containing \quad noise}{Total \quad number \quad of \quad extracted \quad reviews} * 100 \tag{1}$$

For the evaluation of noise filtering we compare the automatically constructed corpus of feedbacks on airline services produced by WebT-IDC, to a manually created gold standard corpus from a subset of the crawled output URLs. We get a p-measure equals to zero, which means that our scraping tool matches exactly the target data and generates completely noiseless data.

(a)



Review	User	Score	Date
Ωραία μυριδιά, τις πρώτες φορές ένιωθες σύσφιξη σα να σε τραβάει μετά όμως τίποτα. Μου έβγαζε λίγα σπυράκια και έτσι δε θα την ξανααγόραζα.	Maria.Pa.	4	02/12/2020
Αργεί πολύ να απορροφηθεί κ μυρίζει περίεργα για κάποια ώρα. Μετά από μία δύο ώρες απτή στιγμή της εφαρμογής είναι οκ. Πιστεύω ότι είναι μόνο για ξηρές επιδερμίδες.	Αναστασίαρ adakistylianiа	3	22/11/2020
Είμαι 29 ετων, λιγη ακμη με μεικτο δερμα. Ωστοσο την αγορασα για μεικτο δερμα αλλα μου αφηνει λιπαροτητα κ γυαλαδα. Την βαζω μονο βραδυ αλλα ενδυτωνει φουλ. Αμεσως εστρωσε το μακιγιαζ μου . Η μυριδιά ουδετερη. Θα την ξανα αγοραζα μονο κ μονο για την ενυδατωση της και για την οικονομική τιμη.	Joanna_J0	4	16/11/2020

(b)

Review	User	Score	Date
Дневной крем для лица Биокон "Натуральный уход" 25+ мне подошел идеально. Начнем с упаковки. Это, конечно, не очень удобная баночка, так как, когда откручиваете крышку, то большая поверхность крема контактирует с воздухом. В общем, не...	LyubaGrom	5	23/10/2018
Новая серия ТМ Биокон "Натуральный уход" привлекла меня прежде всего своим названием- при выборе косметики я стараюсь отдавать предпочтение натуральным средствам. И вот перед Новым годом сделала я себе небольшой подарок- купила три средства из...	NIKITA83	5	07/03/2014

Fig. 6. Examples of reviews extraction from (a) a Greek web page (b) a Russian web page. The top part shows the web page and the bottom part represents the extraction output.

Furthermore, we applied the WebT-IDC tool to extract reviews about beauty products. In this context, we considered web pages in different languages (French, English, Greek, Russian, Italian, ...). Figure 6 shows an example of the extracted output information from respectively Greek and Russian websites.

In addition to information extracted by our tool (the review text, the ranking score, the publishing date and location), we also pre-labelled each review with the key topics deducted automatically from the thematic queries given to the crawler component along with the language label. Hence, each dataset entry is associated with a product category, a polarity label deduced from the ranking score attribute (when the ranking score is higher than 3, we estimate that the reviewer has a positive impression, negative otherwise) and a label defining the language used to express the review. To ensure that the output dataset suits well to machine learning tasks, we prepared a model ready to perform a multi-label and multi-class classification task. It relies on "bert-base-multilingual-uncased" BERT [4] (Bidirectional Encoder Representations from Transformers) pre trained model able to deal with multilingual encoding stacked to three linear layers that respectively encode three different classes: the language of the review, the product category, and the reviewer's impression (whether positive or negative).

We used an input dataset that contains beauty-related reviews to feed the proposed machine learning model. It is a multilingual dataset composed of 45000 English reviews, 36000 French reviews and 24000 Italian reviews. The input dataset was split into a training set and a validation one, with a ratio of 8:2. In this example, Bert gives flawless classification results for the review's language and reviewers' impression classification tasks with 100% accuracy.

For product category classification 89% of accuracy is obtained. This task is challenging compared to the previous ones, since similar product categories exist and similar keywords are employed to describe the characteristics of the reviewed products, and thus creating semantic ambiguities such as skincare and suncare products.

5. Conclusion

In this paper, we presented *WebT-IDC*, an automatic web chain process tool able to gather online feedback of reviews and comments from forums and blogs. It produces a clean corpus, ready-to-use featured dataset for further machine learning tasks. We have shown that our method combining both crawling and scraping components defends itself well against state-of-the-art methods. *WebT-IDC* is independent of the inherent web page architecture and thus adaptable to different forums and blogs containing the pagination element. *WebT-IDC* generates clean thematic multilingual corpus since it matches only relevant target data. The use of multithreading processing and direct access to relevant web page part is a fundamental factor that boosts further the global performance.

Future work incorporates machine learning tasks to improve the identification process for the relevant data region and automatically generate training datasets for a selected domain. Moreover, we propose to add another functional layer for shallow semantic parsing annotation to the extracted corpus as accurately as possible in order to improve the dataset's features.

References

[1] Baroni, M., Bernardini, S., 2004. Bootcat: Bootstrapping corpora and terms from the web., in: LREC, p. 1313.
[2] Baroni, M., Kilgarriff, A., 2006. Large linguistically-processed web corpora for multiple languages, in: Demonstrations. URL: https://www.aclweb.org/anthology/E06-2001.
[3] Clark, J., DeRose, S., et al., 1999. Xml path language (xpath) version 1.0.
[4] Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 .
[5] Evert, S., Kilgarriff, A., Sharoff, S. (Eds.), 2008. Victor: the Web-Page Cleaning Tool.
[6] Gatto, M., 2011. The 'body' and the 'web': The web as corpus ten years on, pp. 35–58.
[7] Htwe, T., Kham, N.S.M., 2011. Extracting data region in web page by removing noise using dom and neural network, in: 3rd International Conference on Information and Financial Engineering.
[8] Johnson, F., Gupta, S.K., 2012. Web content mining techniques: a survey. International Journal of Computer Applications 47.
[9] Kehoe, A., Gee, M., 2007. New corpora from the web: making web text more 'text-like', VARIENG.
[10] Khare, R., Cutting, D., Sitaker, K., Rifkin, A., 2004. Nutch: A flexible and scalable open-source web search engine. Oregon State University 1, 32–32.
[11] Kilgarriff, A., 2007. Googleology is bad science. Comput. Linguist. 33, 147–151. URL: http://dx.doi.org/10.1162/coli.2007.33.1.147, doi:10.1162/coli.2007.33.1.147.
[12] Kilgarriff, A., Grefenstette, G., 2003. Introduction to the special issue on the web as corpus. Computational Linguistics 29, 333–348. URL: https://www.aclweb.org/anthology/J03-3001, doi:10.1162/089120103322711569.
[13] Kumar, A., Singh, R.K., 2017. A study on web structure mining. International Research Journal of Engineering and Technology (IRJET) 4, 715–720.
[14] Malik, S.K., Rizvi, S., 2011. Information extraction using web usage mining, web scrapping and semantic annotation, in: 2011 International Conference on Computational Intelligence and Communication Networks, IEEE. pp. 465–469.
[15] Manad, O., Pappa, A., Bernard, G., 2018. A cleaning algorithm for noiseless opinion mining corpus construction. doi:10.1109/AICCSA.2018.8612867.
[16] Meyer, C.F., Grabowski, R., Han, H.Y., Mantzouranis, K., Moses, S., 2003. The World Wide Web as Linguistic Corpus. Brill — Rodopi. volume 46 of *Language and Computers*. p. 241–254. doi:10.1163/9789004334410_014.
[17] Mozenda, 2017. https://www.mozenda.com/.
[18] Myers, D., McGuffee, J.W., 2015. Choosing scrapy. Journal of Computing Sciences in Colleges 31, 83–89.
[19] Pomikálek, J., 2011. Removing boilerplate and duplicate content from web corpora. Ph.D. thesis. Masarykova univerzita, Fakulta informatiky.
[20] Roh, Y., Heo, G., Whang, S.E., 2018. A survey on data collection for machine learning: a big data - ai integration perspective. ArXiv abs/1811.03402.
[21] Spousta, M., Marek, M., Pecina, P., 2008. Victor: the web-page cleaning tool, in: 4th Web as Corpus Workshop (WAC4)-Can we beat Google, pp. 12–17.
[22] Tschacher, N., 2015. Googlescraper. https://github.com/NikolaiT/GoogleScraper.
[23] Woon, Y.K., Ng, W.K., Lim, E.P., 2005. Web usage mining: Algorithms and results, in: Web Mining: Applications and Techniques. IGI Global, pp. 373–392.