MethodsX 6 (2019) 1054–1067

Contents lists available at ScienceDirect

MethodsX

journal homepage: www.elsevier.com/locate/mex

Method Article

# Attribute value extraction mechanism of Constructed Wetlands information

Mauricio Andres Nevado Amell[a], Muhammad Awais[a], Sowmiya Ragul[a], Kurt Brüggemann[b], Tamara Avellán[b,*]

[a] Technische Universität Dresden, Germany
[b] United Nations University Institute for Integrated Management of Material Fluxes and of Resources (UNU-FLORES), Dresden, Germany

A B S T R A C T

Constructed Wetlands (CWs) are a nature-based solution for the treatment of wastewater. The CWetlands – the Constructed Wetlands Knowledge Platform – intends to help understand how CWs can support achieving the Sustainable Development Goals. The platform is based on more than 100 attributes of CWs including criteria for design criteria, operation, efficiency, climate and other geographical factors. This study aims at developing an attribute value extraction mechanism tool in R to extract meaningful information from peer-reviewed journal articles in a reliable and fast way.

- The tool focuses on the extraction of eighteen different extractable attributes gathered in 4 classes, which describe the main characteristics of CW systems.
- The process contains 4 sub-processes: 1–2) the papers are accessed and pre-processed, 3) the attributes are extracted by two data mining techniques: Keyword Match and Web Scrap, and 4) the values are exported to a database.
- For the development and testing of the tool, 13 articles were used. The tool achieved a mean success rate of 79% in 30 min; less compared with the 480 min needed with a manual approach. In further versions, the tool is expected to obtain a higher success rate in all attributes.



A R T I C L E I N F O
*Method name:* Attribute value extraction mechanism
*Keywords:* Nature-based solutions, Text mining, Natural processing, R, Database
*Article history:* Received 3 December 2018; Accepted 16 April 2019; Available online 30 April 2019

* Corresponding author.
*E-mail address:* avellan@unu.edu (T. Avellán).

https://doi.org/10.1016/j.mex.2019.04.017

**Specifications Table**

| | |
|---|---|
| Subject Area: | Environmental Science |
| More specific subject area: | Constructed wetlands<br>Text mining |
| Method name: | Attribute value extraction mechanism |
| Name and reference of original method: | There is no specific method that was originally developed and then modified. Rather we used methods from other disciplines – i.e. linguistics and text mining – for the development of a tool for value extraction in peer-reviewed journal articles on constructed wetlands<br>Original sources include:<br>F. Ronen and S. James, *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. 2006<br>J. Tiedemann, "Improved Text Extraction from PDF Documents for Large-Scale Natural Language Processing," *Computational Linguistics and Intelligent Text Processing, Cicling 2014, Pt I,* vol. 8403, pp. 102–112, 2014<br>K. Welbers, W. Van Atteveldt, and K. Benoit, "Text Analysis in R," *Communication Methods and Measures,* vol. 11, no. 4, pp. 245–265, 2017/10/02 2017<br>I. Feinerer, K. Hornik, and D. Meyer, "Text mining infrastructure in R," *Journal of Statistical Software,* vol. 25, no. 5, pp. 1–54, Mar 2008 |
| Resource availability: | The source code is openly accessible on GitHub and can be easily modified and used for other research applications for database development. The tool was developed in R studio version 3.5.0 on Windows 10, 64-bit operating system. Necessary installation programs are PostgreSQL 9.5.13 64 bits, PhantomJS 2.1.1 64-bits, and Java 64-bits. Relevant R packages were installed and loaded as per the requirements of R . |

## Method details

We developed a tool in R to extract information on eighteen different extractable a ttributes (see Table 1) from peer-reviewed journal articles related to Constructed Wetlands (CW) using natural processing and text mining tools and exported these via PostgreSQL for display on maps. R is widely accepted in the natural processing of text clustering and text classification [1]. The source code is openly accessible on GitHub https://github.com/CWetlands/Inputs-to-CWetland-using-R and can be easily modified and used for other research applications for database development.

## Procedure

The tool was developed in R studio version 3.5.0 on Windows 10, 64-bit operating system. Necessary installation programs are PostgreSQL 9.5.13 64 bits, PhantomJS 2.1.1 64-bits, and Java 64-bits. Relevant R packages were installed and loaded as per the requirements of R.

The code was validated by contrasting its results with the data of a pre-existing database that was developed manually by the team at UNU in MS Excel containing data from approx. 100 English language peer-reviewed journal articles. For the development of the code, thirteen articles [2–14] having (i) more than three of the attributes from technical specifications class (see Table 1), and (ii) published in journals with the highest number of publications about constructed wetlands (e.g. Ecological Engineering, Water Science and Technology, Bioresource Technology) were chosen from that pre-existing database. Findings of the code were compared with the data from the database to test the reliability of the code.

The tool *Inputs to CWetlands using R* is formed by 4 sub-folders as shown in Fig. 1. The relevance of each sub-folder in the different processes are presented in the Graphical Abstract and how the users should edit/input information to obtain adequate results from the tool is explained as follows:

- **Phantom** – contains the program files of PhantomJS. The tool uses that program for accessing Java components of HTML pages. The excel file *HTML_Links* should be filled with the links from where the documents used in the process *Screen & pre-process peer-reviewed papers* were downloaded. The information in this folder is a pre-requirement for carrying out the sub-process *Web Scrap*.

**Table 1**
List of extractable attributes from peer-reviewed journal articles through the developed code, as well as attribute names and entity names as per the nomenclature used in the CWetlands platform (cwetlands.net).

| Extractable Attribute Description | Extractable Attribute name | Entity | Classes/Groups used by CWetlands-Extraction tool |
|---|---|---|---|
| Title of the article | TITLE | LITERATURE | metadata |
| Year of Publication | YEAR | LITERATURE | metadata |
| DOI of the article[a] | – | – | metadata |
| Journal name | JOURN_NAME | LITERATURE | metadata |
| Publisher | PUBLISHER | LITERATURE | metadata |
| Name of the author/s | AT_FIRST_N | AUTHORS | metadata |
| Last name of the author/s | AT_LAST_N | AUTHORS | metadata |
| Country name | CTRY_NAME | COUNTRY | location |
| Name of the municipality[b] | MUNI | SITES | location |
| Latitude | LAT | SITES | location |
| Longitude | LONG | SITES | location |
| Wastewater type | WW_TYPE | SITES | technical specifications |
| System area | S_AREA | SYSTEMS | technical specifications |
| Plant species | PLANT_SPEC | C_PLANTS | technical specifications |
| Plant genus | PLANT_GENUS | C_PLANTS | technical specifications |
| Type of Constructed Wetland | C_TYPE | CELLS | technical specifications |
| Average $BOD_5$ inflow concentration | C_BOD_IN | C_ORG | treatment performance |
| Average $BOD_5$ outflow concentration | C_BOD_OUT | C_ORG | treatment performance |

[a] Not represented as an individual attribute and, hence, not included in a certain entity, but it becomes part of the attribute CITATION.
[b] A city name can also be part of the attribute *MUNI*, in case the definition of municipality and city differs.

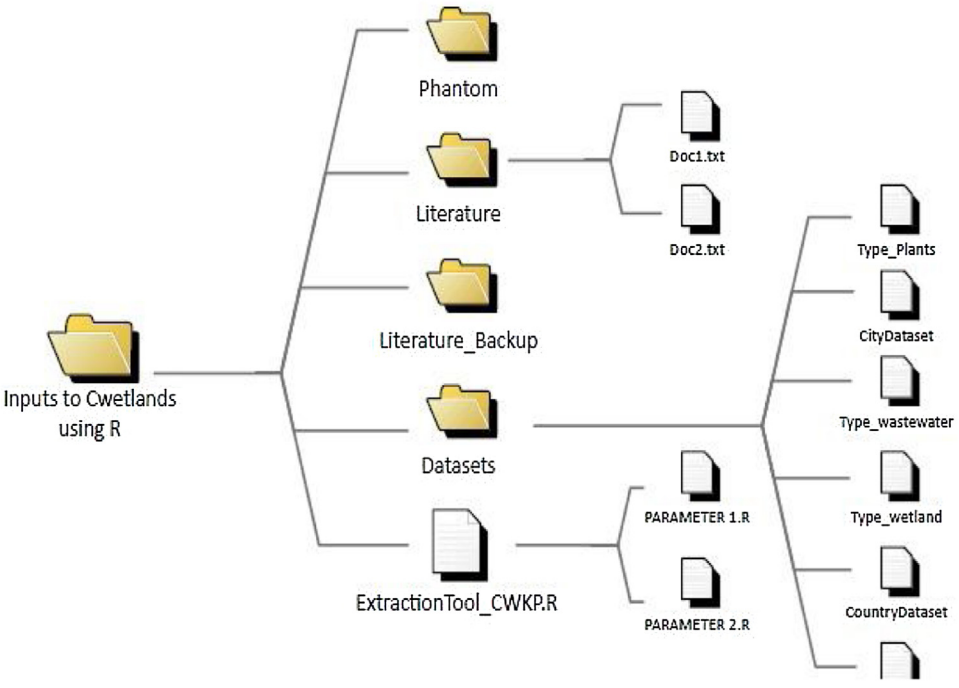


**Fig. 1.** Overview of structure of folders for tool.

- **Literature** – the .txt files created in the process *Screen & pre-process peer-reviewed papers* should be saved in this folder. The tool reads the files from this folder to carry out further processes.
- **Literature_backup** – after the last process *Export information to PostgreSQL*, the tool copies the files in the folder *Literature* to have a backup. Later the tool eliminates the files in the folder *Literature*. If the user wants to run the tool again the next time, the folder *Literature* is already empty, so there is no double analysis of the files from the previous run of the tool.
- **Datasets** – the available datasets e.g. *CountryDataset*, *CityDataset*, *Type_wastewater* are saved in this folder. The tool reads the files from this folder as a requirement to carry out the sub-process *Keyword Match*. Users can modify the datasets files in excel format. The files in .txt (*CountryDataset*, *CityDataset*) should remain unmodified.
- **ExtractionTool_CWKP.R** – the Main Source code in the R format.

The procedure below follows the steps outlined in the graphical abstract.

1 Screen & pre-process peer-reviewed papers

The peer-reviewed journal articles were accessed and downloaded from the citation indexing service Web of Science using access from TU Dresden. In theory, this tool can also work for other online documentation that is downloadable such as reports and books after several adjustments.

Peer-reviewed journal articles were converted from .pdf to .txt files using MS Word and stored as *doc[#].txt* e.g. *doc1.txt*. Other platforms can be used to convert the files to text format. The documents were saved in the *Literature* folder as shown in Fig. 1.

2 Cleaning and Division

The files in the .txt format produced were further processed to a) remove special characters that otherwise hinder the text mining, and b) divide it into sub-sections to allow for more targeted word searches. The original peer-reviewed journal papers have a set of unstructured text such as tables, equations, and figures, which during the conversion to .txt appear as a combination of special characters without a linguistic meaning, e.g. *><|-. Those disordered strings, as well as punctuation characters e.g. *;?!*, multi-white spaces (when there is more than one space between consecutive words) and stop words (word connectors as *"some"*, *"each"*, *"when"*), make processing of regular expressions more complicated thus need to be erased.

The following function of the code takes care of the removal of special characters: *removeSpecialChars<-function(x) gsub("[^a-zA-z0-9.]","",x)*. The expression in brackets preserves all the numbers, letters and points in the text, and eliminates the rest of the characters e.g. special characters and other punctuation characters. Points are maintained because the information in numeric decimal format e.g. *4.5*, has as decimal separator a point. The elimination of the multiple whitespaces is carried out using the code line: *mydocs<-tm_map(mydocs, stripWhitespace)*, which assures that between each word there is just one whitespace. The elimination of stop words is carried out by using the line: *mydocs<-tm_map(mydocs,removeWords,stopwords('en'))*. Fig. 2 shows an example of the text before and after the cleaning step.

Some parameters had several values with the same keywords in the complete text, where only one correct value for the parameter is meant to be extracted. For example, in the case of country name, the *Introduction* part quotes related papers referring to investigations carried out in other countries, which differ from the name of the country where the research of the peer-reviewed article was carried out. The *Materials and Methods* section is eventually the part that mentions the actual country name where the analysis was done. Thus, it becomes necessary to divide the text into different parts to refine the search of the parameter information, and to avoid inconsistent results. The tool divides the text into 4 main parts: *Introduction*, *Abstract*, *Materials and Methods*, *and Results*.

The division in the above format was selected because it follows a general sequential writing structure of the papers i.e. after *Abstract* always comes *Introduction*, followed by *Methods*, then *Results* and finally *Conclusions*, making it easier to define a set of functions in R for doing the task. The structure of the code is as follows:

| | |
|---|---|
| *This work focuses on the performance evaluation of two full-scale horizontal subsurface flow constructed wetlands (H-SSF CWs) working in parallel, which have an almost equal surface area (about 2,000 m2) but with different operational lives: 8 and 3 years. Both H-SSF CWs, located in Southern Italy (Sicily), are used for tertiary treatment of the effluent of a conventional wastewater treatment plant. This study evaluates and compares H-SSF CW efficiency both in terms of water quality improvement (removal percentage) and achievement of Italian wastewater discharge and irrigation reuse limits.* | *This work focuses performance evaluation two full-scale horizontal subsurface flow constructed wetlands H-SSF CWs working parallel almost equal surface area 2000 m2 different operational lives 8 3 years. Both H-SSF CWs located Southern Italy Sicily used tertiary treatment effluent conventional wastewater treatment plant. This study evaluates compares H-SSF CW efficiency terms water quality improvement removal percentage achievement Italian wastewater discharge irrigation reuse limits.* |

**Fig. 2.** Document example before cleaning (left) and after cleaning (right).

```
## Step 6: To clean information before the word "Abstract"
removeAbstract<-function(x) gsub("(.*?)A(?i)bstract(?-i)","",x)
mydocs<-tm_map(mydocs,removeAbstract)
## Step 7: To clean information after the word "Conclusions"
removeConclusions<-function(x) sub("C(?i)onclusion(?-i)(.*)","",x)
mydocs<-tm_map(mydocs,removeConclusions)
## Step 8: To divide the text into four main parts: Abstract, Introduction, Materials and Methods and Results
## Step 8.1: Abstract
removeIntroduction_down<-function(x) sub("I(?i)ntroduction(?-i)(.*)","",x)
str1<-tm_map(mydocs,removeIntroduction_down)
## Step 8.2: Introduction
removeIntroduction_up<-function(x) sub("(.*?)I(?i)ntroduction(?-i)","",x)
str2_fs1.1<-tm_map(mydocs,removeIntroduction_up)
removeMethods_down<-function(x) sub("M(?i)ethods(?-i)(.*)","",x)
str2_fs1.2<-tm_map(str2_fs1.1,removeMethods_down)
removeMethods_down<-function(x) sub("M(?i)aterial(?-i)(.*)","",x)
str2<-tm_map(str2_fs1.2,removeMethods_down)
## Step 8.3: Materials and Methods
removeMethods_up<-function(x) sub("(.*?)M(?i)aterial(?-i)","",x)
str3_fs1.1<-tm_map(mydocs,removeMethods_up)
removeMethods_up<-function(x) sub("(.*?)M(?i)ethods(?-i)","",x)
str3_fs1.2<-tm_map(str3_fs1.1,removeMethods_up)
removeResults_down<-function(x) sub("R(?i)esults(?-i)(.*)","",x)
str3<-tm_map(str3_fs1.2,removeResults_down)
## Step 8.4: Results
removeResults_up<-function(x) sub("(.*?)R(?i)esults(?-i)","",x)
str4<-tm_map(mydocs,removeResults_up)
```

These functions search in the text for the specific words/sentences, which characterize each of the parts e.g. *Introduction*, *Methods/Materials*, and *Methods*, and then eliminate all the information before the word/sentence e.g. *Introduction*, and all the information after the next word/sentence e.g. *Methods* and *Methods*, to divide the text into four main parts. The part *Conclusions* was not considered as the required information is already extracted from the other parts.

The outcome of this process is a cleaned and divided text in an R text data structure called *VCorpus*, which is compatible with the functions used for the subsequent steps.

## 3 Extraction of Attributes

Attributes related to CWs mentioned in journal articles describe different aspects of constructed wetlands. Those aspects can be grouped in different classes as technical specifications (e.g. *Plant species, Plant genus, Type of Constructed Wetland, Wastewater type, System area*), treatment performance (e.g. *Average BOD5 inflow concentration, Average BOD5 outflow concentration*), metadata (e.g. *Title of the article, Name of the author, Journal name, Year of Publication*), and location (e.g. *Country name, Name of the municipality, latitude, longitude*). In its first version, the tool focus on those classes.

Other classes as operational (e.g. *Technology of previous treatment, Technology of posterior treatment, Maintenance frequency*) and economic aspects (e.g. *Investment cost, Maintenance cost*) could be included in a further version of the tool.

To develop the CWetlands-extraction tool, a small subset of attributes (see Table 1) of the CWetlands Platform was chosen. Those attributes were selected because (a) they are the most preferred attributes for potential users of the platform [15], (b) have a high rate of appearance in the peer-review journal papers, and (c) are expected to be extractable in a reliable way by using the text mining principles. Some other attributes as Total Suspended Solids (TSS) fulfilled the selection criteria to be extracted by the tool too. They were not included in the tool in the first place because the main purpose of this first version was to develop and validate the extraction method. Therefore, a reduced amount of attributes was used to achieve that purpose more easily. In further versions of the tool, an expansion of the list of Extractable Attributes is planned.

The process *Extract Attributes* is carried out in two different ways depending on the attribute:

1 by Keyword Match and
2 by Web Scrap

For the first pathway, the tool looks for matching values of a Text Document Matrix and a dataset of expressions in the folder *Datasets* e.g. *Type Plants*. In the second pathway, the tool looks for the information directly in the source webpage where the peer-reviewed article is published. In both ways, the results are subsequently refined by regular expressions to obtain the actual value of the attribute.

In practical terms, extracting information from the web page i.e. *Web Scrap* is easier and more reliable than extracting the information from PDF files i.e. *Keyword Match*, because the conversion from .pdf to .txt is more time-consuming (see step 2). The conversion can also result in disordered expressions/characters, which difficult the implementation of reliable regular expressions to find the required information. The structure of tables is also lost during the conversion from one format to the other, making extraction of meaningful information inside them more difficult. On the other hand, HTML files, besides being comprehensible for any person, are organized in structures that are easily accessible for a programming language such as R by a source HTML code, which avoids most of the disadvantages of the PDF approach. The reason why it is not used throughout is that, unfortunately, not all journals offer their peer-review articles in HTML versions. In most cases, the abstract and the metadata are the only information that is shown in the HTML version. In that sense, *Keyword Match* complements the information that cannot be extracted by *Web Scrap*. Hence, the most effective way to extract information from as many sources as possible is by integrating both approaches.

1 Keyword match

In this process, the tool matches the Text Document Matrix, created by the tool based on two pre-established criteria: a) Number of Words and b) Text Section (see 1 below), and a dataset of keywords (see 2 below and Table 2), which was created by the authors. The attributes extracted in this manner are part of the group *Technical specifications* and the group *Location* used by CWetlands.

This process is divided into the following three sub-processes (see Fig. 3):

1 Define criteria: The aim is to define the criteria for the creation of the Text Document Matrices: Those matrices are generated by a tokenization process, which divides the text saved in the structure *VCorpus,* into sequential strings of N words.
   a Number of Words: This criterion is a range for the variable N for each attribute. The definition of N depends on the attributes, whose possible values are strings of several words. For example, the attribute *COUNTRY NAME* can be the name of a country confirmed by two words as 'South Africa' or just one word as 'Colombia'. This criterion was defined for each attribute by counting the number of words of each of the values extracted from the sample of 13 documents and then identifying the minimum and the maximum number of words. In the example of COUNTRY

**Table 2**
Keywords' attributes.

| Attribute Description | Keywords | OUTPUT column | File[a] |
|---|---|---|---|
| *Location* | | | |
| Country name | Dataset with the name of the countries in the world. | – | *CountryDataset.txt* |
| Name of the Municipality | Dataset with the name of the municipalities/cities in the world. | – | *CityDataset.txt* |
| *Technical Specifications* | | | |
| Plant species/Plant Genus | *Typha latifolia,Typha, latifolia, Phragmites australis, Phragmites, australis, Cyperus alternifolius, Cyperus, alternifolius, Arundo donax, Arundo, donax, etc.* | – | *Type_plants.xlsx* |
| Type of Constructed Wetland | HFCW, Horizontal Flow, Horizontal Flow Constructed Wetland, horizontal flow, surface flow constructed wetlands. | *HORIZONTAL SURFACE FLOW* | *Type_wetland.xlsx* |
| | Vertical Flow System, VFCW, Vertical Flow Constructed Wetland, vertical flow, vertical flow wetland | *VERTICAL FLOW* | |
| | Horizontal Subsurface Flow system, horizontal subsurface flow, Horizontal subsurface flow, subsurface flow wetland | *HORIZONTAL SUBSURFACE FLOW* | |
| Type of wastewater | Municipal raw wastewater, municipal wastewater, domestic wastewater | *MUNICIPAL* | *Type_wastewater.xlsx* |
| | fFood processing industry, Dairy milking parlor, Potato starch processing, eutrophic lake water, industrial. | *INDUSTRIAL* | |
| | urban runoff agriculture, agricultural, swine urine. | *STORMWATER AGRICULTURAL* | |

[a] Excluding *CityDataset.txt* and *CountryDataset.txt*, the files can be modified by adding new keyword expressions.

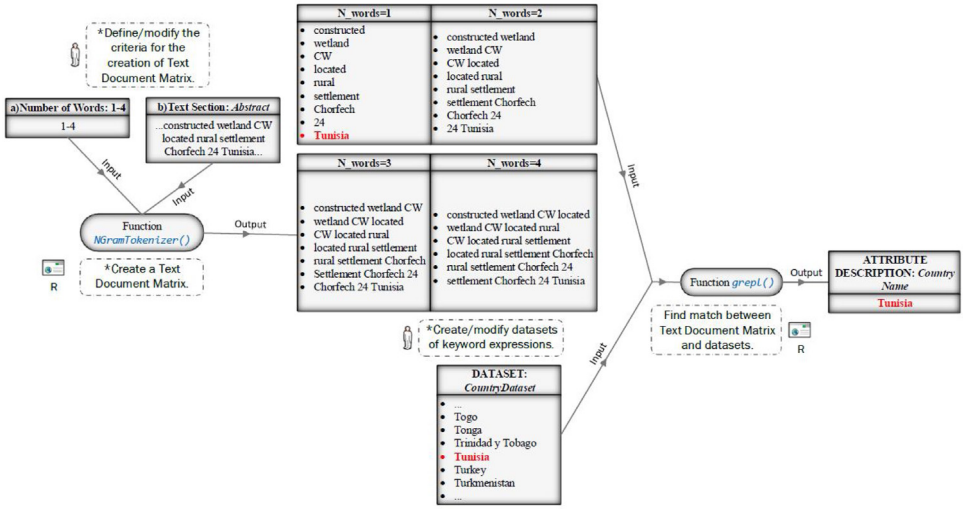


**Fig. 3.** Outcome of the sub-process Keyword Match (ref. Graphical Abstract). Example from [13].

NAME, the range was set up in 1–4, which means that this attribute can take values confirmed by 1, 2, 3 or 4 words.

b Text Section: This criterion is the text section i.e. *Introduction, Abstract, Materials and Methods, Results*, from which the matrix is being created. This criterion was defined by identifying for each

attribute, in which section the value was found in each of the 13 documents. For example, *COUNTRY NAME* is most likely to be found in the *Abstract, Materials & Methods* section.

2 Database of keyword expressions: a dataset of keyword expressions for each of the attributes was developed (see Table 2). Those datasets are a list of possible values that an attribute can take. They are based on the analysis of the selected 13 peer-reviewed articles. For example, in the case of the attribute *TYPE OF PLANTS*, the expressions which referred to plant name were highlighted in each article and then inserted into the file *Type_plants.xlsx*. The different datasets found in the folder *Datasets* can be easily expanded with more expressions extracted from the analysis of more peer-reviewed articles in the future. Those expressions should be added respecting capitalization to obtain a match in the sub-process *Keyword Match* because the tool considers uppercase and lowercase characters as different e.g. *Typha* is different from *typha*. Only the datasets *CountryDataset* and *CityDataset* must remain unmodified because the name of the cities and countries in the world are not expected to change in the close future.

3 In the final sub-process, the tool uses the function *NGramTokenizer()* to create the Text Document Matrix based in the criteria, and the function *grepl()* looks for matching values of the Text Document The created matrix and the dataset of keyword expressions are saved in the folder *Datasets*.

```
## Step 3: To extract the country name
## Rules
## 1- SP's: Abstract, Materials, and Methods
## 2- Tokenizer: min=1, max=4
 for (m in 1:N.docs){
     Tokenizer<-function(x) NGramTokenizer(x, Weka_control(min=1,max=4))
     head(NGramTokenizer(VCorpus_docs3[m],Weka_control(min=1,max=4)))
     dtm3<-TermDocumentMatrix(VCorpus_docs3[m],control=list(tokenize=Tokenizer,tolower=FALSE))
```

In the example of *COUNTRY NAME*, the first criterion was set to 1–4 i.e *Weka_control(min = 1, max = 4)*, and the second criterion was set to *Abstract, Materials & Methods* i.e. *VCorpus_docs1[m]* refers to Abstract and *VCorpus_docs3[m]* refers to *Materials and Methods*. An overview of the keywords extracted from the literature and used in the development process can be seen below.

The different authors of the peer-reviewed articles can express the same attribute value e.g. *HSSF-Horizontal Subsurface Flow* using different expressions e.g. *subsurface flow wetland* or *horizontal subsurface flow*. That is a characteristic of the natural language. To obtain a consistent and uniform attribute value, it is necessary to group the expressions representing the same value in a cluster group. The datasets of keywords corresponding to the attributes *Type of Constructed* Wetland and *Type of wastewater* contain an additional column called *OUTPUT*, which contains the names of the cluster groups defined by the authors (see Table 2). The tool finds first the matching keyword expression after the sub-process *Keyword Match*, then looks for the corresponding cluster group name and saves it as the attribute value.

Web scrap

In this process, the information is extracted directly from the web pages. In HTML, the information in a webpage, e.g., tables, graphs, images, body text, headlines, is identified by a query language called *XPath* through *XPath expressions*. By knowing those expressions, several functions are used in R to obtain the data. The parameters extracted by this approach are part of the group *Metadata* e.g. *Journal name, Title of the article, Name of the author* and the group *Performance* e.g. *Average BOD5 inflow concentration, Average BOD5 outflow concentration*.

This process is divided into the following 3 sub-processes:

1 The user must insert the links of the pages from where the peer-reviewed articles were downloaded as well as the name of the corresponding documents e.g. *Doc1*, *Doc2*, in the file *HTML_links.xlsx* of the folder *Phantom*.

2 The specific *XPath expressions* (see Table 3) were searched for, which identify the information that we wanted to extract. For example, for the page *Elsevier*, the *XPath="span.title-text"* identifies the name of the title of the peer-reviewed article. Each web page has different *XPath expressions* for extracting the same information. In this case, we used expressions of four sites namely: *Elsevier, NCBI*, *IWA*, and *ResearchGate.*

3 The R function *html_nodes()* was used to access the information identified by a specific *XPath expression* from a specific link. The function *html_text()* extracts the information previously accessed when it is text, and the function *html_tables()* when it is a table. In the case of the last function, sometimes the web pages have the information encoded by Java components, therefore it is necessary to call a program called *PhantomJS*, to access the tabular information. The program does not need to be installed and its files are found in the folder *Phantom*. It is worth to mention, that the code used for calling that program in R is configured for working just with the operating system Windows and that the files in the folder were downloaded for a 64 bits version.

**Table 3**
XPath and Regular Expressions.

| Attribute Descriptions | XPath Expression | Regular Expression/s |
|---|---|---|
| *Web Scrap* | | |
| Title of the article | *span.title-text* | – |
| Year of Publication | *div.text-xs* | *grepl(pattern="(?i)volume(?-i)")*<br>*sub("â\|u0080\|u0093","-")*<br>*sub("(.*?)[,]","")*<br>*sub("[,](.*)","")* |
| DOI of the article | *a.doi* | *sub("(.*?)doi.org","doi:")* |
| Journal name | *h2#publication-title.publication-title* | – |
| Publisher | – | *grepl("sciencedirect")* |
| Name of the author/s | *span.text.given-name* | *sub("Ã","ä")*<br>*sub("Ãµ","õ")*<br>*sub("Ã\|u009c","Ü")* |
| Last name of the author/s | *span.text.surname* | *sub("Ã\|u0096Ã¶","Öö")* |
| *Keyword Match* | | |
| Latitude | – | *sub(".*(\|\|d+{2}).*$", "\|\|1")*<br>*sub("([0-9]{2}).*", "\|\|1")*<br>*sub("(^[^0-9]*)(\|\|d+)([^0-9].*)", "\|\|2")*<br>*grepl(pattern="^[0-9]{5}W")*<br>*grepl(pattern="^[0-9]{5}E")* |
| Longitude | – | *sub(".*(\|\|d+{2}).*$", "\|\|1")*<br>*sub("([0-9]{2}).*", "\|\|1")*<br>*sub("(^[^0-9]*)(\|\|d+)([^0-9].*)", "\|\|2")*<br>*grepl(pattern="^[0-9]{5}W")*<br>*grepl(pattern="^[0-9]{5}E")* |
| Wastewater type | – | – |
| System area | – | *grepl(pattern="[[0-9]m2$")*<br>*grepl(pattern="[0-9] m2.$")*<br>*grepl(pattern="[0-9] m2$")*<br>*grepl(pattern="[0-9]m2.$")*<br>*sub("[0-9].*","")*<br>*sub("m(.*)","")*<br>*sub("[[: space:]]+ ","")* |
| Average $BOD_5$ inflow concentration | – | *grepl("(?i)influent(?-i)")\|grepl("(?i)inflow(?-i)")*<br>*sub("(?i)Â(.*)(?-i)","")*<br>*grepl("BOD")*<br>*grepl("(?i)deviation(?-i)")*<br>*grepl("(?i)in(?-i)")* |
| Average $BOD_5$ outflow concentration | – | *grepl("(?i)efluent(?-i)")\|grepl("(?i)outflow (?-i)")*<br>*sub("(?i)Â(.*)(?-i)","")*<br>*grepl("BOD")*<br>*grepl("(?i)deviation(?-i)")*<br>*grepl("(?i)out(?-i)")* |

The information extracted by *Keyword Match* or *Web Scrap* might need further processing to be ready for export to the PostgreSQL database. Some unnecessary words/or expressions need to be removed. For doing that, regular expressions through the R function *gsub()* are used. This function searches for the unwanted words/characters by using the search patterns defined by the regular expressions and then eliminates these.

For example, in the case of the attribute *System Area*, the process *Keyword Match* produces the following set of expressions: *"area* 5000 $m^2$." and *"single* 806 $m^2$", from which just the numeric characters should be extracted. For doing so, the tool follows the next steps: (1) If any of the expressions have just letters as characters, it is discarded by using the regular expression *"[0–9].*"* i.e. in this example, both expressions are preserved because they include the numbers *5000* and *806*. (2) The expressions remaining are refined by eliminating the unit e.g. m2, whitespaces and other words e.g. *"single"*, by using the regular expressions *"m(.*)" and "[[: space:]]+"*. The outcome of the process is: *"5000–806"*.

The regular expressions were defined by analyzing the results of the process *Extract Attributes* in the sample of 13 documents (see Table 3). The regular expressions can be further improved depending on the attributes and characteristics found in further articles.

1 Export information to PostgreSQL

The aim of this process is to export the attribute values extracted to a database in PostgreSQL: an open source object-relational database management system connected to CWetlands. The structure of the database was developed by the team of CWetlands at UNU, and it consists of 34 tables which are correlated by Foreign Keys i.e. column's information shared by various tables. It was decided to use that relational database structure because it splits the information into different entities, which makes it easy to navigate/modify the database in PostgreSQL.

This process is divided into two sub-processes:

1 The tool generates a data frame i.e. a tabular structure in R, where the information that was refined in the step *Use Regular Expression* is saved along with Derived Attributes (see Table 4) and Primary Keys. The last ones are columns in the PostgreSQL database tables, which uniquely identify each row in the table. The following features have to be observed: each row must contain a unique value and it cannot contain null values. For example, in the table *LITERATURE*, the Primary Key is the column *LIT_ID*, which can take values as *LIT_1*, *LITE_2*, *LIT_3*. The values of the Primary Keys for the new information that is going to be exported need to be determined following the sequence of the information already in the PostgreSQL database. For example, if the last Primary Key in the PostgreSQL table *LITERATURE* is *LIT_8*, then the new information Primary Keys must start from *LIT_9*. For doing so, the R function *dbGetQuery()* is used, which reads PostgreSQL tables into R tracking the current sequence of the Primary Keys. Additionally, Derived Attributes are attributes that were not obtained in the process *Extract Attributes*, but that are created by joining or combining

**Table 4**
List of Derived Attributes, as well as attribute names and entity names as per the nomenclature used in the CWetlands platform.

| Derived Attribute Descriptions | Extractable Attributes | Derived Attribute names | Entities | Classes/Groups used by CWetlands-Extraction tool |
|---|---|---|---|---|
| Citation of the peer-review article | TITLE | LIT_CITE | LITERATURE | metadata |
| | YEAR | | | |
| | DOI | | | |
| | JOURN_NAME | | | |
| | PUBLISHER | | | |
| | AT_FIRST_N | | | |
| | AT_LAST_N | | | |
| Country code | CTRY_NAME | CTRY_NAME | SITES | location |
| Latitude | MUNI | LAT | SITES | location |
| Longitude | MUNI | LONG | SITES | location |

the extracted ones. For example, the attribute *CITATION* is derived by joining the attributes in the entity *LITERATURE* and *AUTHORS* using the function *paste0()*. Also, Derived Attributes can be obtained by inserting values of an Extractable Attribute into a function. For example, the attributes *LAT* and *LONG* are derived by using the function *getGeoData()*, which gives the value of the coordinates by inserting the extracted attribute MUNI. The tool uses the last approach when there is no information about the coordinates in the peer-reviewed articles.

2 The information in the data frame created in the first sub-process is copied into eight tables in R which are equivalent to the tables in PostgreSQL. Those tables are *SITES, COUNTRY, SYSTEMS, CELLS, C_PLANTS, LITERATURE, C_ORG*, and *AUTHORS*. Then the tables in R are exported to PostgreSQL by using the function *dbWriteTable()*.

## Efficiency check

The tool uses keywords and data extracted from 13 articles and it was checked for the same articles. The average time consumed by the automatic approach (tool) was compared against the one consumed by a manual approach by stop watching the time used by both approaches to process the same peer-reviewed journal articles. The automatic process took 30 min for all 13 articles (processing time) whereas the manual process of extraction took around 480 min i.e. the time needed by one person of the team of UNU to process manually the 13 articles. The efficiency of each attribute corresponding each article in the sample is given in Table 5: the value *0* means that the tool could not extract the attribute or that the extracted value was not the same as the one noted down in the manual process. The value 1 means that the attribute values are equal in both processes. Finally, the value *N/A* means that there was no information about the attribute in the peer-reviewed article.

Success rates (values of 1 where attribute values were equal) vary from 57.14% as in the case of the attribute MUNI, to 100% as in the case of attributes *JOURNAL_NAME* or *PUBLISHER* (see Table 5 and Fig. 4). The overall success rate across all parameters and all articles was 87%. Attributes in the entity *LITERATURE, COUNTRY* and *C_PLANTS* show high success rates (90–100%), which means that the tool

**Table 5**
Efficiency performance by attributes.

| Entity | Attributes | Article | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | Efficiency |
| COUNTRY | Country name | 1 | 1 | 1 | N/A | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 100 |
| SITES | Name of the Municipality | 1 | 0 | N/A | N/A | N/A | 0 | N/A | N/A | N/A | 1 | 1 | 0 | 1 | 57.1 |
| SITES | Latitude & Longitude | 1 | 1 | N/A | N/A | N/A | 0 | 1 | N/A | N/A | 1 | 1 | 0 | 1 | 75 |
| SITES | Wastewater type | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | N/A | 1 | 1 | 1 | 66.7 |
| SYSTEMS | System area | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 76.9 |
| CELLS | Plant species/genus | 1 | 1 | 0 | 1 | 1 | N/A | 1 | 1 | 1 | 1 | N/A | N/A | 1 | 90 |
| CELLS | Type of Constructed Wetland | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 76.9 |
| AUTHORS | Name/Last name of the authors | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 100 |
| LITERATURE | Title of the article | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 100 |
| LITERATURE | Publisher | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 100 |
| LITERATURE | Year of Publication | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 100 |
| LITERATURE | Journal Name | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 100 |
| LITERATURE | DOI | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 100 |
| S_ORG | BOD inflow | 0 | N/A | N/A | N/A | 0 | N/A | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 22.2 |
| S_ORG | BOD outflow | 0 | N/A | N/A | N/A | 0 | N/A | 1 | 0 | 1 | 0 | 0 | 0 | N/A | 25 |

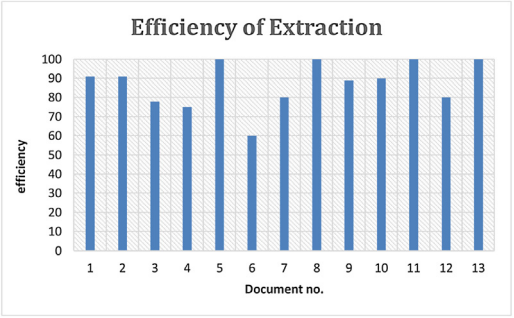


**Fig. 4.** Efficiency performance by literature in the sample.

works well in extracting the information in those entities. Attributes in the entities *SITES* and *S_ORG* show low success rates (75–22.2%) and attributes in the entities *SYSTEMS* and *CELLS* show medium success rates (90–76.9%).

Currently, there are certain limitations to the tool which affects the overall efficiency of extraction;

1 One of the reasons for the lower success rate in some attributes is that during the process of conversion to .txt (*Screen and pre-process pre-review articles*) some information originally contained in the PDFs (e.g. images or equations) is lost. For example, the latitude and longitude information in some peer-review articles can be in the format of latex expressions which then are converted to disordered expressions when the format is changed. Even after the process of cleaning (*Clean and Divide*), the remaining syntax is still too particular and does not follow a general structure. Complex regular expressions to extract a very specific disordered syntax is thus used in the tool for some cases e.g. *sub("(^[^0-9]*)(||d+)([^0-9].*)", "||2")*. The tool can be further improved by employing more complex regular expressions.
2 Another reason is that in some cases more than one value is assigned for each attribute. Hence, after the process *Use Regular Expressions*, if an attribute results in more than one value, a conditional statement sets the value for this parameter to *N/A* (if the there is more than one match, then put the value equal to *N/A*). For example, in the attribute *MUNI*, the name of the Italian city *San Michele di Ganzaria* should be found, but there is also a city in Italy called *Plant*, and this word is found in the normal context of the article. The tool finds two matches i.e. *San Michele di Ganzaria* and *Plant* and saves the attribute value as *N/A* instead of *San Michele di Ganzaria*.
3 Finally, another reason is that the datasets *Type_plants.xlsx, Type_wastewater.xlsx* and *Type_wetland.xlsx* were formed with a sample of 13 kinds of literature. These datasets certainly do not cover the whole database of possible values that either attribute can take. Information from more peer-review articles will increase the size of the datasets. However, it is not easy to define the number of peer-review articles needed to be analyzed to have reliable datasets, without compromising the goal of the tool to reduce the human time needed in the extraction operation.

On the other hand, the overall efficiency of extraction between the documents was found very varied. Some conditions can be stated which might explain the variance:

1 The HTML version of the document is available or not in the journal page: The condition allows to process completely the text of the document by the process *Web Scrap* (See 3.1.2). Therefore, the limitations encountered in the process *Keyword Math* (See 3) are diminished and the overall efficiency of extraction is expected to be higher.
2 The document reports or not some data attribute values: Some authors provide more detailed information about their research. Therefore, documents, where the data attribute information is

reported, are expected to yield higher efficiency of extraction compared to ones where the data information is not provided.

As an example, document 6 presented a low overall efficiency of extraction because (1) the HTML version of the document was not available on the journal page, and (2) the document does not report some of the data attribute values i.e. represented by N/A values.

Further refinement of the tool may improve its performance. Some recommendations for the improvement of this tool are provided below:

1 1 **Define an extraction efficiency threshold for the extractable attributes**: As the tool would not achieve a 100% efficiency of extraction, a minimum acceptable threshold must be defined. The criterion could be based on the relevance of a specific extractable attribute to the purposes of the possible users of the platform. For example, the main purpose of the platform CWetlands is to: *Compare data for feasibility studies and/or during the design phase* [25]. Attributes highly correlated with that purpose as *Average BOD5 outflow concentration* should have a high efficiency of extraction e.g. more than 80%.

2 **Calculate a range of efficiency:** The tool will extract attributes values from a population of 7000+ peer-reviewed journal papers. Based on the standard deviation and the mean of the extraction efficiency obtained from the sample used to test the tool, the null hypothesis that the extraction efficiency of the Extractable Attributes for the whole population are the values defined previously (see 1) could be tested for an expected value of $\alpha$ (the risk of rejecting a true hypothesis) and $\beta$ (the risk of accepting a false null hypothesis when a particular value of the alternative hypothesis is true). The values of $\alpha$ and $\beta$ depend on the sample size, therefore an estimate number of peer-review articles that should be further included to test the tool efficiency should be calculated to obtain those values.

3 **Evaluate the range of efficiency**: if the range is below the acceptance threshold, then decide between the following options.

a) Analyze more peer-review articles to increase the coverage of the datasets and add/modify specific regular expression/functions in the tool code, which can increase the attribute efficiency. *Approach used when there are indications that a representative increase in efficiency can be achieved.*

b) Take off the attribute from the tool because the reliability is not good enough for the Constructed Wetlands Knowledge Platform user's purposes. *Approach used when there is evidence that a representative increase in efficiency cannot be achieved by applying the option 2.1.*

4 **Add new attributes**: Identify new attributes to add to the tool. A starting point: look for attributes with similar features to the attributes with high efficiency already in the tool. *Then repeat the steps 1–3 for the new attributes.*

5 **Adapt tool for peer-review articles referring to wetlands conformed by multiple cells**: The first version works for peer-reviewed articles, which refer to constructed wetlands composed by just one unit/cell. For multiple-cell wetlands, the attributes are extracted in an aggregated version e.g. area equal to 20–30 $m^2$. In the following improvements, the tool should be able to disaggregate the information for each cell, so that the user of the platform can have more broad and detailed data about wetlands' attributes.

## Acknowledgments

The authors wish to thank their respective donors for making this work possible. Tamara Avellán thanks the German Federal Ministry of Education and Research (BMBF) and the Saxon State Ministry for Science and the Arts (SMWK) for providing research funding for UNU-FLORES. Mauricio Nevado wishes to thank the Deutsche Akademische Austauschdienst (DAAD) for supporting his studies at the Technische Universität Dresden (TUD). Muhammad Awais and Sowmiya Ragul would like to thank the Association of Friends and Sponsors of TU Dresden (GFF) for their extended support through stipend

during the study project work. The authors also thank Prof Liedl at TUD for organizing the Hydro-Science and Engineering Study Project, under which this research was carried out.

## Appendix A. Supplementary data

Supplementary material related to this article can be found, in the online version, at doi:https://doi.org/10.1016/j.mex.2019.04.017.

## References

[1] I. Feinerer, K. Hornik, D. Meyer, Text mining infrastructure in R, J. Stat. Softw. 25 (5) (2008) 1–54.
[2] A.C. Barbera, G.L. Cirelli, V. Cavallaro, I. Di Silvestro, P. Pacifici, V. Castiglione, et al., Growth and biomass production of different plant species in two different constructed wetland systems in Sicily, Desalination 246 (1) (2009) 129–136.
[3] A. Ghrabi, L. Bousselmi, F. Masi, M. Regelsberger, Constructed wetland as a low cost and sustainable solution for wastewater treatment adapted to rural settlements: the Chorfech wastewater treatment pilot plant, Water Sci. Technol. 63 (12) (2011) 3006–3012.
[4] B.C. Braskerud, Factors affecting phosphorus retention in small constructed wetlands treating agricultural non-point source pollution, Ecol. Eng. 19 (1) (2002) 41–61.
[5] E.J. Dunne, M.F. Coveney, E.R. Marzolf, V.R. Hoge, R. Conrow, R. Naleway, et al., Efficacy of a large-scale constructed wetland to remove phosphorus and suspended solids from Lake Apopka, Florida (vol 42, pg 90, 2012), Ecol. Eng. 52 (2013) 316.
[6] J. Vymazal, Horizontal sub-surface flow constructed wetlands Ondřejov and Spálené Poříčí in the Czech Republic – 15 years of operation, Desalination 246 (1) (2009) 226–237.
[7] K. Kato, T. Inoue, H. Ietsugu, T. Koba, H. Sasaki, N. Miyaji, et al., Design and Performance of Hybrid Reed Bed Systems for Treating High Content Wastewater in the Cold Climate, (2010) .
[8] M.P. Ciria, M.L. Solano, P. Soriano, Role of macrophyte Typha latifolia in a constructed wetland for wastewater treatment and assessment of its potential as a biomass fuel, Biosyst. Eng. 92 (4) (2005) 535–544.
[9] M.A. Belmont, E. Cantellano, S. Thompson, M. Williamson, A. Sánchez, C.D. Metcalfe, Treatment of domestic wastewater in a pilot-scale natural treatment system in central Mexico, Ecol. Eng. 23 (4) (2004) 299–311.
[10] M. Öövel, A. Tooming, T. Mauring, Ü. Mander, Schoolhouse wastewater purification in a LWA-filled hybrid constructed wetland in Estonia, Ecol. Eng. 29 (1) (2007) 17–26.
[11] S.Ç Ayaz, N. Findik, L. Akça, N. Erdoğan, C. Kınacı, Effect of recirculation on organic matter removal in a hybrid constructed wetland system, Water Sci. Technol. 63 (10) (2011) 2360–2366.
[12] S. Barbagallo, G. Cirelli, A. Marzo, M. Milani, A. Toscano, Hydraulic behaviour and removal efficiencies of two H-SSF constructed wetlands for wastewater reuse with different operational life, Water Sci. Technol. (2011) 1032–1039.
[13] S. Prost-Boucle, P. Molle, Recirculation on a single stage of vertical flow constructed wetland: treatment limits and operation modes, Ecol. Eng. 43 (2012) 81–84.
[14] S.Y. Chan, Y.F. Tsang, H. Chua, S.N. Sin, L.H. Cui, Performance study of vegetated sequencing batch coal slag bed treating domestic wastewater in suburban area, Bioresour. Technol. 99 (9) (2008) 3774–3781.
[15] K. Brüggemann, Analysing Data and Requirements for the Design of a Web-Accessible Database and GIS of Constructed Wetlands, Technische Universität Dresden, Dresden, 2018.
[25] K. Welbers, W. Van Atteveldt, K. Benoit, Text analysis in R, Commun. Methods Meas. 11 (4) (2017) 245–265.