IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 25, No. 1, February, 2020, pp. 11-19
Journal homepage: http://iieta.org/journals/isi

Information Extraction Model to Improve Learning Game Metadata Indexing

Morie M. Wielfrid[1*], Marfisi-Schottman Iza[2], Goore Bi Tra[1]

[1] Institut National Polytechnique Felix Houphouët-Boigny (INPHB), 1093 Yamoussoukro, Côte d'Ivoire
[2] Le Mans Université, EA 4023, LIUM, 72085 Le Mans, France

Corresponding Author Email: maho.morie@inphb.ci

https://doi.org/10.18280/isi.250102

Received: 12 October 2019
Accepted: 2 January 2020

Keywords:
educational ontology, information extraction, game indexing, learning games, semantic web

ABSTRACT

The use of Learning Games (LGs) in schools is a success factor for students. The benefits they bring to the learning process should be widely disseminated at all levels of education. Currently, there are thousands of LGs that cover a large variety of educations fields. Despite this large choice of LGs, very few are used by teachers, due to the difficulty of finding and selecting suitable LGs. The aim of this paper is to propose an extraction model that will automatically collect the information about LGs directly from their web pages, in order to index them in a catalogue. The proposed ADEM (Automatic Description Extraction Model), browses the web pages describing LGs and does a first cleaning to remove any unnecessary information. Then a detection of description blocks, based on a certain number of criteria, identifies the regions containing the LG description text. Finally, an indexing on specific fields is performed. ADEM made it possible to automatically process 785 web pages to extract LG metadata indexing information. The results of this extraction process were validated by 20 teachers. This model therefore offers a promising starting point for better LG indexing and the creation of a complete catalogue.

1. INTRODUCTION

The introduction of Learning Games (LGs) in schools has shown the great potential of games for education [1-3]. LGs have hence become increasingly known to teachers and students from kindergarten to higher education [4, 5]. The development of digital LGs in particular, has expanded considerably these past years, due to the popularity of computers, tablets and smartphones [6]. However, even if teachers are aware of the existence of LGs and want to use them, very few do. Indeed, they encounter difficulties in selecting LGs for their teaching activities. Looking for LGs with classic search engines is very time consuming and brings little satisfaction [7, 8]. In addition, there are very few catalogues that offer a wide range of LGs and that are equipped with a filtering system that allows teachers to find the LGs that meet their specific needs (Table 1).

In addition, these catalogues are updated manually [9, 10]. This means that it is a human who adds the LGs to the catalogues and fills in the metadata (i.e. name of the LG, subject taught, level of study) that will be used to filter them. This indexing task is tedious [6] and, when performed by humans, can include errors. Automatic or semi-automatic indexing would allow more LGs to be considered and would facilitate the work. The insertion of new LGs could be done automatically. But, how to index these LGs when we know that the information provided on the designers' webpages is neither standardized nor structured in the same way [11]? How to extract relevant information such as the domain of the LG, the platform or the learning level for which it is intended?

We try to answer this question by proposing an Automatic Description Extraction Model (ADEM). First, the model goes through the web in search of LG web pages. Then, ADEM extract the information that describes the games on these websites and extracts the metadata useful for the automatic indexing of these websites.

In this article, we present the work done on tools and methods for extracting information from websites in section 2. In section 3, we propose a model for automatic extraction of metadata from web pages. This section explains the different steps of the ADEM model design. Section 4 presents the experiments carried out with the ADEM model. In the experimentation part, we discuss the model's performance on a selection of LG websites and the evaluation of the results obtained. To conclude, we discuss the contributions of the model and its concrete use in a LG catalogue.

Table 1. List of learning games catalogues

Catalogue	All Games	Nb of LGs	Update freq
Serious Game Classification	3,300	402	+1 / Day
Moby Games	110,558	260	+3 / Day
Serious Games Fr	183	74	+1 / Month
MIT Education Arcade	8	7	On Project
Vocabulary Spelling City	42	42	On Project

- Nb of LGs: total number of LGs of catalogues
- Update freq: frequency with which the LGs are added to the catalogues
-The URLs of each catalogue can be found in the appendix.

2. RELATED WORK

Current LG catalogues use manual indexing, which consists in asking a human to analyze and extract all the relevant information on the LGs' webpage. The people who are in

charge of this task are LG experts or enthusiasts who have a good level of knowledge about the LGs cited [11]. They search on social media feeds, blogs or directly on the webpage of companies that produce LGs, in order to find new LGs and index them in their catalogues, according to their own classification model [12, 13]. The description information about these LGs is either copied as it is or formatted according to the classification model used by the catalogue [14]. This information formatting requires a phase of familiarization, analysis and translation of the original documents and the LG itself. For example, the Serious*GameClassification* and *MobyGames* platforms [14, 15], which have 20 years of existence, count more than 100 contributors.

The problem with this method is the heaviness of the task. Moreover, it can only be done by an expert who knows where to find new LGs and who knows the catalogue description model [16, 17]. In addition, most of these catalogues offer all types of games, learning and non-learning and or not always up to date [11, 18, 19]. Teacher who are looking for LGs will therefore have to browse several catalogues before finding the appropriate one. Table 1 presents statistics on the seven biggest (most LGs) and most updated catalogs we found in the literature [11, 20].

In order to create a LG catalogue that covers all types of levels and educational fields and that is automatically updated with new LGs, it is necessary to reduce human intervention and switch to an automatic method that will scan the pages of the LG editors' webpages to retrieve the necessary information, analyze it and format it according to an indexing standard. This is where the fist difficulty appears: LG editors do not follow standards such as LOM (Learning Object Metadata) [21-23], MLR (Metadata for Learning Resources) [24] or ontology-based systems [25, 26] to define their games [27]. This greatly complicates the automatic indexing task, since the system cannot immediately understand the information.

Early research, that deals with the automatic analysis of web pages, analyze the pages' DOM (Document Object Model) tree, to extract the HTML (*Hypertext Markup Language)* tags that potentially contain useful information [28, 29]. This process is only possible if the webpage structure is known [30, 31]. The problem is therefore the same, since it involves human intervention to analyze the structure of the page, inducing potential errors and a slow indexing process [32]. The page analysis therefore must be fully automated for the extraction of information on the LGs. One possible solution could be to make a statistical analysis of the weights of the information contained in the branches of the web page's DOM tree, in order to identify the regions where the important information, concerning the LG, could be. The work carried out by Velloso et al. [33], which uses signal processing techniques to perform this regions analysis, is interesting because it allows to determine approximately which parts of the web page contain the information describing the LG. However, this technique has the particularity of bringing a lot of noise (i.e. irrelevant information), such as the content of headers and side sections of the webpage. This technique must be combined with further processing to analyze the collected data [34].

The implementation of a system that will be trained to identify regions in the DOM tree path that contain the required information [10] can also be interesting. This system seems especially relevant for extracting information on platforms that host multiple LGs with the same presentation pattern for each LG page. Indeed, once the first pages are processed, identifying the regions of the DOM tree on similar pages will be easy. However, this learning phase needs to be done for all the discovered LG webpages.

As we can see, current methods do not allow us to move closer to our initial objective of automating the extraction of information describing LGs, or to reduce human influence in their indexing. Using keyword recognition would not work any better, since it will have pick up text in the advertisements and related articles [29, 34]. The information we want to extract from the webpages is only the information that describes the LGs. The system should therefore be able to identify the regions of the page that contain this description information, clean it, find the terms identifying the attributes that will be used to index the LGs, and all this, automatically.

3. AUTOMATIC METADATA EXTRACTION MODEL DESIGN

3.1 Webpage browsing

Our objective is to limit the expert's intervention to the minimum. Thus, in addition to automatically extracting the keywords used to index LGs, the ADEM system must be able to automatically collect the web pages of these games. To do so, ADEM uses a list of URLs pointing to teachers' blogs, catalogues of LG publishers and websites specialized in learning resources (Table 1). This is not ideal in itself, but as a base, it allows us to find LGs that correctly answer the vast majority of users' needs, as these catalogs are part of major projects in the world of LGs. In addition to facilitate the automatic inclusion of new LGs, it will be easy to browse through new catalogues with small parameter tweaks. To collect only links that deal with LGs, the system ignores links to a domain name that is different from the one of the analyzed websites. Then, links that do not contain the words related to the game title are left. Finally, the remaining links are analyzed. For example for the *SeriousGamesGlassification* platform, we start from the link of its link (Appendix, Table 5), on which we retrieve all games whose link starts by "*http://serious.gameclassification.com/FR/games/*" and contains the title of the game (i.e. the object of the link) with a hyphen instead of spaces ({/18480-10-Minute-Solution/index.html} for *10 Minute Solution* LG). Web pages are documents structured with *Hypertext Markup Language* (HTML) tags, which frame the content that will be displayed or executed by the browser. The source code of the page should respect specific conventions defined in the documentation [35]. This HTML source code allows browsers to build the DOM tree of the web page. This Document Object Model (DOM) represents the document asset of nodes and objects with properties and methods.

The DOM route allows you to select specific HTML tags to reach a region of the web page. Most web pages are built the same way: there is a header containing the site name, navigation menus, advertising areas, a main area that contains web page information and a footer. In some cases, we may have web pages that do not respect this global structure, but this is not a problem as the HTML tags are universal [36]. The construction of web pages always follows the same semantic. For example, <h1> tags are used to give a title to the web page and <td> tags are used for tables. There are three types of HTML tags: block tags such as , <p>, <header> which are used for visual organization [37, 38], line tags such as

, which are used to format the text and inline-block tags that are used for optional content [39]. The fact that each tag has a specific meaning, even if they are not always used according to the HTML 5 recommendations, makes the content extracting easier.

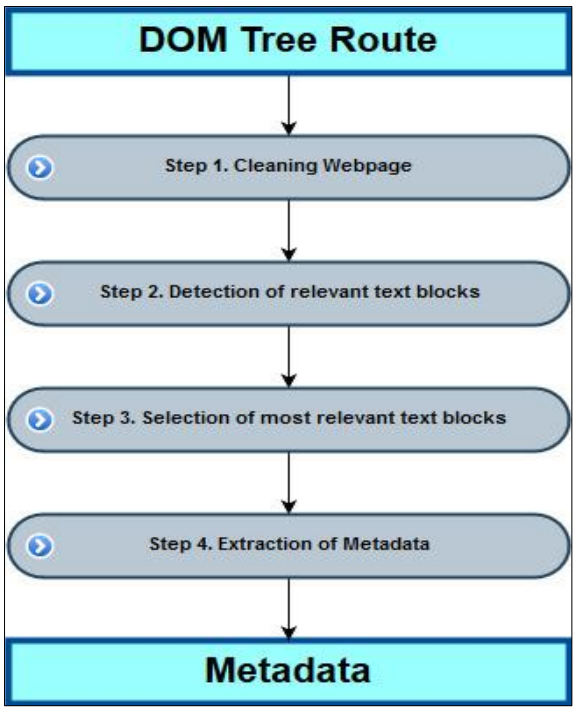

Figure 1. Activity diagram of ADEM steps

The ADEM model we propose, consists of four steps (Figure 1):

Step 1: Clean the web page in order to keep only the regions that contain text describing the LG.

Step 2: Detect the text blocks containing the description of the LG and retrieve the keywords for classification attributes.

Step 3: Selection of most relevant text blocks.

Step 4: Extract terms of Metadata from description text analysis.

3.2 Step 1: Webpage cleaning

In step 1, the web page is cleaned by removing unnecessary regions. Everything outside the Body tag is first deleted. Then, the header and footer areas with the tags <header> and <footer>, and tags with attributes of this type, are also deleted. Non-HTML tags are also deleted. Then, the menu tags such as <aside> and <nav> and the form design tags such as <form> and <input> are removed. In fact, the most important tags in a web page, that contain main content, are <article> and <section>. However, in addition to older web pages created before the implementation of HTML 5 [36], some web site designers do not use the types of tags recommended by the HTML 5 standard to describe their web page content and they still use tags for all types of content. Thus when analyzing a web page, ADEM will first looks for the new semantic tags defined above and, if it does not find them, it will then look for the values of the "id" and "Class" attributes of the tags that are semantically close to the content tags of HTML 5. For example, in Figure 2, the web page does not contain a specific HTML 5 tag, but we have tag "id" attributes that contain words like *navbar*, *sidebar*, *content*... which are close to the <nav> and <aside> tags in HTML 5. Words "Content" and "main" are searched for because they are used to define the main content of the web page [40].

Tags with empty content are also deleted along with image or graphic representation tags and title tags <h(i, ∀i∈{1...6})>. For example, for the *Supercharged* webpage (Figure 2), ADEM removes the background image, the header, the footer and *navbar* menu. Only the main content is left (Figure 3).

After this cleaning, if a <*div*> tag is contained in another <*div*> tag, we separate this tag from parent tag and so on. Finally, we have only the <*div*> tags which do not contain another <*div*> tags.

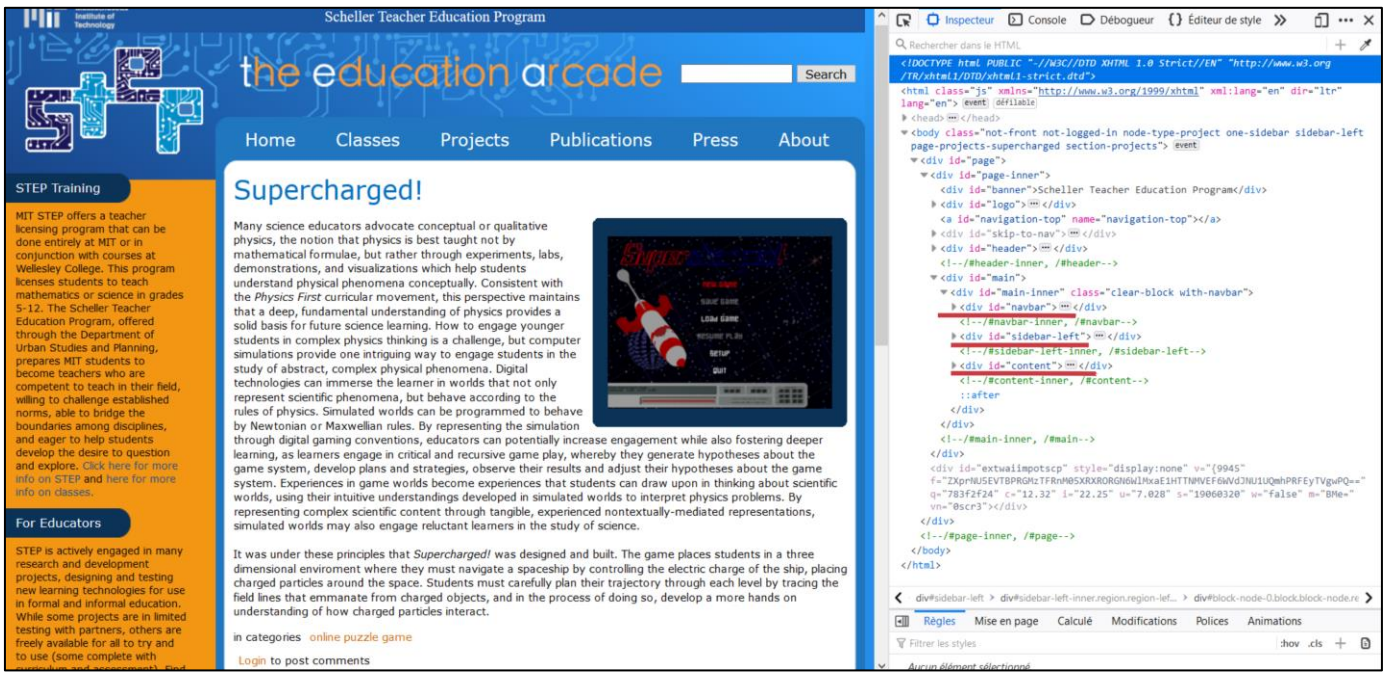

Figure 2. "*Supercharged!*" Game webpage with its code source

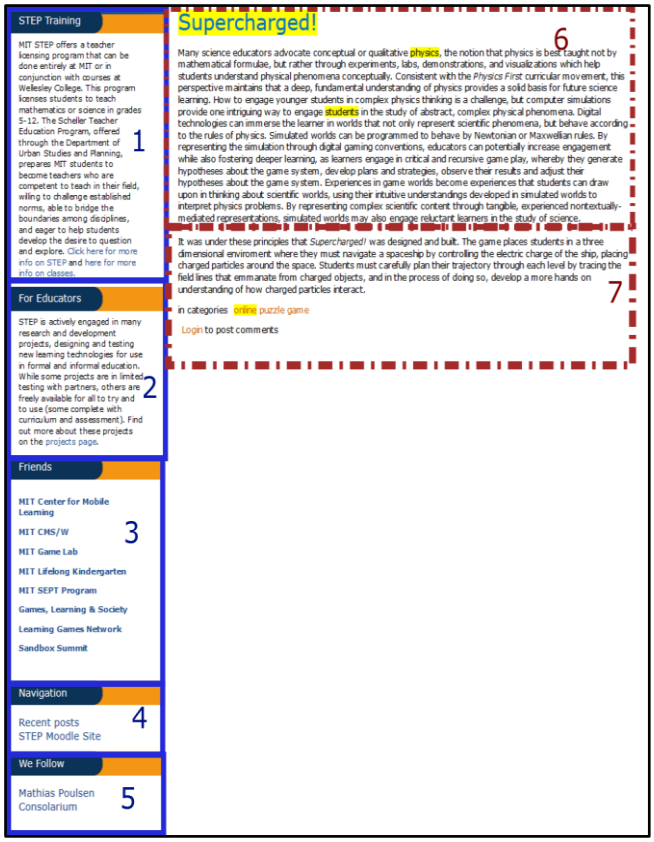

Figure 3. *"supercharged!"* webpage relevant text blocks detected

3.3 Step 2: Detection of relevant text blocks

The detection of the relevant text blocks is done with the following criteria:

- The description texts are framed by paragraph tags <*p*>.
- In most cases, regions containing descriptive texts also contain the least Hypertext links, i.e. the tag <*a*>.
- The texts describing the LGs are the ones that contain the most inline tags (e.g. <*span*>, <*em*>, <*br*/>...)

Thus, a ratio calculation is performed on all the remaining blocks containing text, after the cleaning step, according to the above criteria.

For the ratio calculation of a given <*x*> tag, *Rx* represents the number of <*x*> tags in each block tag except the <*p*> tags, which do not contain other block tags represented by *TagRi* (*x*), on the total of this <*x*> tag of the remaining regions after cleanup, *TagPage* (*x*). At this level all calculations are done on the remaining areas after the cleaning phase. With this ratio calculation formula, we calculate the proportion R_p of tags <p>, R_a of tags <a> and R_{in} of the inline type tags defined by <in> contained in each block tag and compare it to the entire web page. Propotions R_p (Eq. 2), R_a (Eq. 3) and R_{in} (Eq. 4) is from Eq. 1 where *x* is respectively equal to tag <*p*>, tag <*a*> and all inline tags, <*in*>.

$$R_x = \frac{TagR_i(x)}{TagPage(x)} \tag{1}$$

$$< for\ x\ =\ p\ >:\ R_p = \frac{TagR_i(p)}{TagPage(p)} \tag{2}$$

$$< for\ x\ = a\ >: Ra\ = \frac{TagR_i\,(a)}{TagPage(a)} \tag{3}$$

$$< for\ x\ =\ in\ >: Rin\ = \frac{TagR_i\,(in)}{TagPage(in)} \tag{4}$$

Our hypothesis is that the text block containing the LG description, should have the highest R_p and R_{in} ratios found in the DOM tree, since these tags are used to format the texts on which the user should focus. On the other hand, the R_a ratio should be lowest because it is in the long text regions that there are the least Hypertext links.

To determine the $AreaT_i$ region, which contains the text used to describe the LG, the three ratios will be examined by priority, namely the highest R_{in}, then the highest R_p, and finally the lowest R_a. Another criterion that comes into play and which is very important is the weight P_{words} of the words in *TagRi*, so all tags in *TagRi* are eliminated and the number of words remaining is counted. Finally, we can reduce the scenario to an optimization problem where we look for *AreaTopt* regions that correspond to an optimal situation, i.e. that have their V_R value above average. $Avg(V_R(AreaTi))$ correspond to average value of all V_R of all page block.

$$\mathrm{V_R} = R_{in} + R_p + \mathrm{P}_{words} - R_a \tag{5}$$

$$\forall\ AreaT_i \in AreaT, AreaT_i \in AreaT_{opt}\ /\ V_R(AreaT_i) > Avg(V_R(AreaTi)) \tag{6}$$

For example (in Figure 3), the *Supercharged!* Webpage contains 7 remaining blocks. In the remaining tags, we have a total of 17 <a> tags, 20 <p> tags, 7 inline tags and 496 words. Thus, we selected blocks 1, 6 and 7 according to the scores obtained by each of the blocks meeting the criteria of formula 6.

Table 2. Supercharged webpage Blocks ratio score

BLOCK	R_{in}	R_p	P_{words}	R_a	V_R
1	0	0.1	99	0.11	**98.99***
2	0	0.1	60	0.055	60.05
3	0	0.4	25	0.47	24.93
4	0	0.1	05	0.11	4.98
5	0	0.1	03	0.11	2.88
6	0.85	0.1	220	0	**220.95***
7	0.28	0.1	84	0.11	**84.27***
Average (V_R)					**70.86**

*V_R Scores greater than Average of V_R 70.86 is selected

The $Avg(V_R(AreaT))$ value that corresponds to the average of VR in Table 2, gives a value of 70.86, and on the data in this table, we only have blocks 1, 6 and 7 with a score greater than this average.

3.4 Step 3: Selection of most relevant text locks

If several text blocks are in an optimum situation, i.e. at least two text boxes have been selected and therefore could contain the description information, only the blocks or nodes, as defined by the DOM, that have the same immediate parent will be considered. In Fig 5, the blocks 1, 6 and 7 are in an optimum situation. However, only block 6 and 7 are together in the same parent node, thus, block 1 is not selected. This discrimination is important because it allows to ignore content that is not related to the description of the game. As we could see, if we had kept Block 1, that gives the general objectives and missions of the MIT STEP project, terms like "Mathematics"

and "grades 5-12" would have created many false positives.

3.5 Step 4: Extraction of metadata

After retrieving the description text of each detected area, an analysis of the text will begin for the domains to which each game is related. To do this, it is first important to know the text language. In this, the system analyzes *lang* attribute of the HTML tag of the web page -in this work we considered the English and French languages-, if this is not specified, it is the text that is analyzed to determine its language by measuring the frequency of characters and reference words [41]. From the knowledge of the language, the text is freed of Stop Words, which consists of deleting words that have no syntactic interest. Once this step is carried out, we lemmatize the text, then the major terms are grouped according to their similarity by their common synonyms. This grouping reduces the size of the remaining word vector that will be used to determine the scope of LGs according to an educational ontology.

To extract terms describing the level for LGs, ADEM uses the thesaurus of the *European schoolnet Vocabulary Bank for Education* [42]. The terms of this vocabulary bank that match the terms of the text the closest are chosen. If no match is found, ADEM considered the LG is for the general public.

Regarding the platform on which the LG can run, and since we focus on LGs usable in a classroom, ADEM only keeps LGs that can run on the following platforms; PC, Tablet, Smartphone, Mac, and on the following operating systems; Windows, Linux, MacOs, iOS, Android, Windows Phone and online.

For example, for the LG in Figure 3, the words "physics", were identified in the description text in addition to other elements such as platform, gender, and domain, which gives "online", "puzzle" and "student" respectively (words highlighted in yellow in Figure 3).

As a result, ADEM automatically collected 785 LG web pages. The number of LGs collected is more than enough compared to the number of games in the *SeriousGameClassification* catalogue. Moreover, each time a new LG is added to the catalogue, the ADEM system will automatically add it to the catalogue. Another important point is that ADEM indexes only LGs which removes all noise in the selection of these LGs by the users as seen in Table 1 showing a major gap between the total number of games and the number of LGs collected by these catalogs.

4. EXPERIMENTATION AND RESULTS

4.1 Experimental design

The objective of this first experimentation is to validate the fact that ADEM can automatically extract relevant information about LGs, by analyzing the content from their editors' web page. To determine the relevance of the system, the evaluation was conducted with 15 teachers. Indeed, we want to know if ADEM can extract the description information of LGs and with what level of accuracy. To do this, we asked teachers to measure the accuracy of the extracted information, because they are the first target of LG libraries. The profile of these teachers is diverse in terms of the subjects they teach (3 language, 6 science, 1 sport, 5 technology) as well as the level of teaching (2 in primary, 8 in secondary, 7 in higher education).

Table 3. Learning Games keywords automatically extracted from ADEM

LEARNING GAMES NAME	DOMAIN	PLATFORM	LEVEL
Lure of the Labyrinth	Math, Algebra	Online	College
Supercharged	Physics	Online	Student
StarLogo	Programming system	Online	Age +10
MecheM	Chemistry	Online	Age -17
Foldit	Science, biology	PC, online	All
Robot Tueur	Mechanic	Online	Student
Fuite Fatal	Electricity	Online	Student
Estimation du bien-être en entreprise	Statistic, Math	Online	Student
Mission a Emosson	Mechanic	Online	Student
Learn Japanese To Survive! Hiragana Battle	Japanese	PC	All
codecombat	Code python, JavaScript	Online	College
Prog & Play	programming language	Pc	College
taxman	number, math	Online	All
Algo bot	Programming	Online	College
Typing of dead	Computer	PC	All
Cranky	Math	Online	All
10 minutes Solution	Sports, body exercises	Online	3rd person
Zombie division	math	Online	College
Poubelle ecologique	Tri dechets	Online	age -18
Algebots	Math	Online	age -12
english taxi	English, Chinese	Online	Student
Reconnaître les déterminants	Langue, Français	Online	age -11
check.io	programming	Online	Student
Color memory games	Memory game	Tablet	Kids

Out of the 785 LGs extracted by ADEM, we provided these teachers with a selection of 24 LGs. In order to assure ADEM worked on all types of webpages, we selected well formatted webpages, poorly formatted webpages, webpages with presentation popup, platform webpages, webpages with several LGs and webpages with Flash animations. The metadata extracted by ADEM for these 24 in Table 3. This tool was design in the computer language *Python* with the library *Beautifulsoup*. We conducted the tests on a computer powered by an *8-core Intel Core i7 9th* generation process, a *Nvidia*

GetForce GTX 1050 GPU and a 16 Gb RAM memory. The extraction of metadata was done in 2 steps. First, connecting to web pages to retrieve the source code was done with the python library *Beautifulsoup*. The second step was to analyze the source code of the web pages to extract the metadata. This experimentation computation speed averaged around 2000 milliseconds for each LGs web page. However, this running time is essentially due to the first step with *Beautifulsoup* 4 connecting to the web page with internet to retrieve and put them in object form with 1600 milliseconds on average [43]. For each LG, we trying to find by ADEM metadata of the LGs domain, platform requirements and public level required to use them (Table 3). Thus, the metadata obtained on the web pages of these 24 LGs by ADEM tool are recorded in Table 3. For example, for LG "*Supercharged!*", we extracted for domain, platform and public level metadata respectively "physics", "online" and "student" such as in Table 3.

For each LG, the teachers assigned a score between 0 and 5, depending on the keywords ADEM extracted in relation to those they would have chosen on the web site. In addition, they gave a percentage of precision for each extracted description text. They compare ADEM extraction with what they see on LGs webpage. They assign the score according to matching level succeed.

This experiment was carried out over a period of 2 months. In practice, difficulties identified by the teachers concerned the accuracy of the text, especially on web pages with a lot of diverse information such as *Foldit* and *Prog & play* web pages. the analysis of these web pages required more time and reflection from them. This is precisely the problem that ADEM wants to resolve.

4.2 Result

In addition to the result of the teachers' evaluation that defines the general accuracy (scale of 1 to 5) and percentage of precision of the keywords, the validity of the model was measured by the noise metric that is widely used in the field of content extraction [36, 37] and represents the level of unnecessary texts collected by the system.

$$Noise = 1 - \frac{Pw}{Mw} \quad (7)$$

Mw is the number of words of the text obtained by the model, Pw is the number of words of the description text obtained by the teacher on the LG webpage.

The results of the evaluation, in Table 4, show that ADEM extracts the description content of the LGs on their web page with an average of 85% accuracy, and never lower than 80%, which is considered a good threshold [35]. In terms of keyword retrieval for the metadata phase, we have an average score of 4/5 with a minimum of 3/5.

However, there are problems for the LG such as *Foldit* with metrics of 0, cause of the description of the LG which is not on the LG page but rather a lot of information about the ecosystem of the LG. In addition, there is a lot of noise in the recovery of description content, for some games, which is explained by the poor organization of the pages concerned with the information scattered in several unrelated nodes with the use of HTML tags that do not respect defined logic. For web pages with a good content structure, the noise level is around 0%, like those of "*Cranky"* and "*10 Minutes solution"* which have their web pages built on the same model the MobyGames platform. The noise level is much higher for "*Robot tueur"*, "*Fuite Fatal*", "*Estimation du bien-être en entreprise*" and *"Mission à Emosson"* because these games are described on the same page "*Les Ecsper"* which contains 6 games for this project, with a pop-up that appears when you click on one of the LG (Figure 4).

Table 4. ADEM evaluation results

LEARNING GAMES	[0-5] Keywords	% Text Precision	% Noise
Lure of the Labyrinth	3,2	98,13	28,4
Supercharged	5	99,38	49,6
StarLogo	4,3	93,75	23,9
MecheM	3	96,88	18,54
Foldit	3	0	-
Robot Tueur	4,6	73,75	82,9
Fuite Fatal	4,6	73,75	83,3
Estimation du bien-être en entreprise	4,5	73,75	86,2
Mission a Emosson	4,7	73,75	85,1
Learn Japanese To Survive! Hiragana Battle	4,2	100	44,9
Code combat	4,5	74,38	63,2
Prog & Play	4,1	88,75	11,5
taxman	4,5	92,5	26,5
Algo bot	3,1	92,5	38,4
Typing of dead	2,1	93,75	10,12
Cranky	3,8	89,38	0
10 minutes Solution	4	95,63	1,6
Zombie division	4,7	100	12
Poubelle ecologique	4,6	100	3
Algebots	4,9	77,5	1,4
English taxi	4,9	81,88	3,3
Reconnaître les déterminants	4,5	100	1,9
check.io	5	84,38	14
Color memory games	4,9	100	0

Figure 4. Les Ecsper Webpage show pop-up when we click on one LG

Thus, the observed noise represents the description

information of the other LGs that the model retrieves from the webpage. Despite these high noises, we still have a high level of relevance with a threshold of 70%, which is very good [35] because it is based on this relevance that the information describing the LGs can be analyzed for other types of applications.

5. CONCLUSION

The idea of automatically extracting information describing Learning Games (LG) aims to solve time consuming and error-prone manual indexing of LGs. The proposed Automatic Description Extraction Model (ADEM) can automatically extract the relevant information that describes a LG and the keywords for metadata from a webpage, without prior knowledge of the organization and structure of its DOM tree. ADEM meets the objectives set with good precision both in the extraction of relevant text and in the search for keywords. The main difficulty for extracting information automatically come from the fact that the text containing the LGs' description is sometimes in the middle of insignificant text blocks, which increases the noise found in the extracted text. To reduce this noise level, ADEM proceeds to several steps that clean and select of the potentially interesting text blocks, before performing a syntax analysis to identify keywords used for LG description.

ADEM can be used by LG catalogs to improve and accelerate indexing tasks by simply specifying game pages for automated indexing. To improve the system, we could involve humans at a low level, especially for pages that are very poorly formatted or contain a lot of flash content. For the keywords of the metadata phase having a corpus of domain terms can be a good perspective in improving the system.

Currently, ADEM presents some limitations. First, relevant text areas could have special CSS formatting styles. However, the current HTML DOM parsing does not consider extra styles such as CSS tags. This could be done by scanning the CSS files to analyze HTML areas that receive special processing from attribute tags. The challenge here is to create an efficient processing for text blocks in CSS. Another limitation of ADEM is the very basic keyword search. The use of a collaborative domain ontology of LGs could improve this phase and include more languages easily.

Finally, ADEM could be used to automatically index all kinds of items on webpages. We intend on providing users with an interface where they will be able to adjust parameters in ADEM, according to their objectives and desired keywords.

REFERENCES

[1] Kiili, K., de Freitas, S., Arnab, S., Lainema, T. (2012). The design principles for flow experience in educational games. Procedia Computer Science, 15: 78-91. https://doi.org/10.1016/j.procs.2012.10.060

[2] Su, C.H., Cheng, C.H. (2015). A mobile gamification learning system for improving the learning motivation and achievements. Journal of Computer Assisted Learning, 31(3): 268-286. https://doi.org/10.1111/jcal.12088

[3] Tüzün, H., Yılmaz-Soylu, M., Karakuş, T., İnal, Y., Kızılkaya, G. (2009). The effects of computer games on primary school students' achievement and motivation in geography learning. Computers & Education, 52(1): 68-77. https://doi.org/10.1016/j.compedu.2008.06.008

[4] Sanchez, E., Ney, M., Labat, J.M. (2011). Jeux sérieux et pédagogie universitaire: de la conception à l'évaluation des apprentissages. Revue Internationale des Technologies en Pédagogie Universitaire, 8(1-2): 48-57. https://doi.org/10.7202/1005783ar

[5] Rawn, C.D., Fox, J.A. (2018). Understanding the work and perceptions of teaching focused faculty in a changing academic landscape. Research Higher Education, 59: 591-622. https://doi.org/10.1007/s11162-017-9479-6

[6] Marfisi-Schottman, I. (2012). Méthodologie, modèles et outils pour la conception de Learning Games., INSA Lyon, France.

[7] Omari, A., Shoham, S., Yahav, E. (2016). Cross-supervised synthesis of web-crawlers. In Proceedings of the 38th International Conference on Software Engineering, NY, USA, pp. 368-379. https://doi.org/10.1145/2884781.2884842

[8] Pesare, E., Roselli, T., Corriero, N., Rossano, V. (2016). Game-based learning and Gamification to promote engagement and motivation in medical learning contexts. Smart Learning Environments, 3: 5. https://doi.org/10.1186/s40561-016-0028-0

[9] Prensky, M. (2005). Engage me or enrage me: What today's learners demand. Educause Review, 40(5): 60-65.

[10] Gibson, D., Aldrich, C., Prensky, M. (2007). Games and Simulations in Online Learning: Research and Development Frameworks. Information Science Publishing.

[11] Alvarez, J., Plantec, J.Y., Vermeulen, M., Kolski, C. (2017). RDU Model dedicated to evaluate needed counsels for Serious Game projects. Computers & Education, 114: 38-56. https://doi.org/10.1016/j.compedu.2017.06.007

[12] Guemmat, K.E., Lahmar, E., habib B., Talea, M., Lamrani, E.K. (2015). Implementation and evaluation of an indexing model of teaching and learning resources. Procedia - Social and Behavioral Sciences, 191: 1266-1274. https://doi.org/10.1016/j.sbspro.2015.04.654

[13] Ratan, R., Ritterfeld, U. (2009). Classifying serious games. Serious Games: Mechanisms and Effects, 10-24.

[14] Djaouti, D., Alvarez, J., Jessel, J.P. (2011). Classifying serious games: The G/P/S model. Handbook of Research on Improving Learning and Motivation through Educational Games: Multidisciplinary Approaches. https://doi.org/10.4018/978-1-60960-495-0.ch006

[15] Adams, E. (2014). Fundamentals of Game Design. New Riders Publishing, Thousand Oaks, CA, USA.

[16] Wirth, C., Fürnkranz, J. (2015). On learning from game annotations. IEEE Transactions on Computational Intelligence and AI in Games, 7(3): 304-316. https://doi.org/10.1109/TCIAIG.2014.2332442

[17] Marfisi-Schottman, I., George, S., Tarpin-Bernard, F. (2011). Un profil d'application de LOM pour les Serious Games. In Environnements Informatiques pour l'Apprentissage Humain, Conférence EIAH'2011. pp. 81-94. Editions de l'UMONS, Mons 2011, Belgium.

[18] Fronton, K., Vermeulen, M., Quelennec, K. (2015). Les Ecsper : retour d'experience d'une etude de cas de type serious game en gestion de projet. In: e-Formation des adultes et des jeunes adultes. Equipe TRIGONE, laboratoire CIREL. Centre Interuniversitaire de Recherche en Education de Lille (EA 4354), Lille,

France.
[19] Mitamura, T., Suzuki, Y., Oohori, T. (2012). Serious games for learning programming languages. In 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Seoul, South Korea, pp. 1812-1817. https://doi.org/10.1109/ICSMC.2012.6378001
[20] Gottron, T. (2008). Combining content extraction heuristics: The combine system. In Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services, New York, NY, USA., pp. 591-595. https://doi.org/10.1145/1497308.1497418
[21] Freire, M., Fernández-Manjón, B. (2016). Metadata for serious games in learning object repositories. IEEE Revista Iberoamericana de Tecnologias del Aprendizaje. 11(2): 95-100. https://doi.org/10.1109/RITA.2016.2554019
[22] Neven, F., Duval, E. (2002). Reusable learning objects: A survey of LOM-based repositories. In Proceedings of the Tenth ACM International Conference on Multimedia, ACM, New York, NY, USA, pp. 291-294. https://doi.org/10.1145/641007.641067
[23] Rajabi, E., Sicilia, M.A., Sanchez-Alonso, S. (2015). Interlinking educational resources to Web of Data through IEEE LOM. Computer Science and Information Systems, 12(1): 233-255. https://doi.org/10.2298/CSIS140330088R
[24] Currier, S. (2008). Metadata for Learning Resources: An Update on Standards Activity for 2008. Ariadne.
[25] Hernandez, N., Mothe, J., Ramamonjisoa, A.B.O., Ralalason, B., Stolf, P. (2009). Indexation multi-facettes des ressources pédagogiques pour faciliter leur réutilisation. Institut de Recherche en Informatique de Toulouse. ftp://ftp.irit.fr/IRIT/SIG/2008_RNTI_HMRRS.pdf.
[26] Marne, B., Wisdom, J., Huynh-Kim-Bang, B., Labat, J.M. (2012). The six facets of serious game design: A methodology enhanced by our design pattern library. In European Conference on Technology Enhanced Learning, pp. 208-221.
[27] Pernin, J.P. (2004). LOM, SCORM et IMS-Learning Design: Ressources, activités et scénarios. In: actes du colloque "L'indexation des ressources pédagogiques numériques", Lyon, France.
[28] Crescenzi, V., Mecca, G., Merialdo, P. (2001). Roadrunner: Towards automatic data extraction from large web sites. In Proceeding of the 27th VLDB conference, Roma, Italy, pp. 109-118.
[29] Bharti, S.K., Babu, K.S. (2017). Automatic keyword extraction for text summarization: A survey. arXiv:1704.03242 [cs].
[30] Arasu, A., Garcia-Molina, H. (2003). Extracting structured data from web pages. In Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, ACM, New York, NY, USA, pp. 337-348. https://doi.org/10.1145/872757.872799
[31] Roldán, J.C., Jiménez, P., Corchuelo, R. (2017). Extracting web information using representation patterns. Presented at the (2017). https://doi.org/10.1145/3132465.3133840.
[32] van Deursen, A., Mesbah, A., Nederlof, A. (2015). Crawl-based analysis of web applications: Prospects and challenges. Science of Computer Programming, 97(Part 1): 173-180. https://doi.org/10.1016/j.scico.2014.09.005
[33] Velloso, R.P., Dorneles, C.F. (2017). Extracting records from the web using a signal processing approach. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, ACM, New York, NY, USA, pp. 197-206. https://doi.org/10.1145/3132847.3132875
[34] Lytvyn, V., Vysotska, V., Chyrun, L., Smolarz, A., Naum, O. (2017). Intelligent system structure for Web resources processing and analysis. Presented at the Computational Linguistics Andintelligent Systems (COLINS 2017).
[35] Wu, Y.C. (2016). Language independent web news extraction system based on text detection framework. Information Sciences, 342: 132-149. https://doi.org/10.1016/j.ins.2015.12.025
[36] Keith, J. (2010). HTML5 for web designers. A Book Apart, New York, NY.
[37] Berners-Lee, T., Hendler, J., Lassila, O. (2001). The semantic web. Scientific American, 284: 34-43.
[38] Bast, H., Björn, B., Haussmann, E. (2016). Semantic search on text and knowledge bases. Foundations and Trends® in Information Retrieval, 10(2-3): 119-271. https://doi.org/10.1561/1500000032
[39] Lubbers, P., Albers, B., Salim, F. (2011). Pro HTML5 programming. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4302-3865-2
[40] Hoy, M.B. (2011). HTML5: A new standard for the web. Medical Reference Services Quarterly, 30(1): 50-55. https://doi.org/10.1080/02763869.2011.540212
[41] Lui, M., Lau, J.H., Baldwin, T. (2014). Automatic detection and language identification of multilingual documents. Transactions of the Association for Computational Linguistics, 2: 27-40. https://doi.org/10.1162/tacl_a_00163
[42] Massart, D. (2009). Towards a Pan-European Learning Resource Exchange Infrastructure. In Feldman, Y.A., Kraft, D., Kuflik, T. (eds.) Next Generation Information Technologies and Systems, Springer, Berlin, Heidelberg, pp. 121-132. https://doi.org/10.1007/978-3-642-04941-5_14
[43] Zheng, C.M., He, G.M., Peng, Z.J. (2015). A study of web information extraction technology based on beautiful soup. JCP, 10(6): 381-387. https://doi.org/10.17706/jcp.10.6.381-387

APPENDIX

The Table 5 and Table 6 are respectively Table 1 and Table 3 data with their URLs.

Table 5. List of learning games catalogues

Catalogue	URL
GameClassification	http://serious.gameclassification.com/FR/games/index.html
Moby Games	https://www.mobygames.com/browse/games/list-games/
Serious Games Fr	https://www.serious-game.fr/category/serious-games/
MIT Education Arcade	https://education.mit.edu/project-type/games/
Vocabulary Spelling City	https://www.learninggamesforkids.com/

Table 6. List of learning games

LEARNING GAMES	URL
Lure of the Labyrinth	https://education.mit.edu/project/lure-of-the-labyrinth/
Supercharged	https://web.mit.edu/mitstep/projects/supercharged.html
StarLogo	https://www.slnova.org/
MecheM	http://serious.gameclassification.com/EN/games/16870-MeCHeM/index.html
Foldit	http://Fold.it
Robot Tueur	http://lesecsper.mines-douai.fr/
Fuite Fatal	http://lesecsper.mines-douai.fr/
Estimation bien-être en entreprise	http://lesecsper.mines-douai.fr/
Mission a Emosson	http://lesecsper.mines-douai.fr/
Learn Japanese To Survive!	https://igg-games.com/learn-japanese-175115241-to-survive-hiragana-battle-free-download.html
codecombat	https://codecombat.com/
Prog & Play	http://programminggames.org/Prog-Play.ashx
taxman	https://www.mobygames.com/game/browser/taxman-game
Algo bot	http://www.algo-bot.com/
Typing of dead	http://www.jeuxvideo.com/jeux/pc/00004646-the-typing-of-the-dead.htm
Cranky	https://www.mobygames.com/game/cranky/
10 minutes Solution	https://www.mobygames.com/game/wii/10-minute-solution
Zombie division	http://serious.gameclassification.com/FR/games/3152-Zombie-Division/index.html
Poubelle ecologique	http://www.gameclassification.com/FR/games/66-La-poubelle-ecologique/index.html
Algebots	http://www.socialimpactgames.com/modules.php?op=modload&name=News&file=article&sid=249
English taxi	http://www.desq.co.uk/sections/portfolio/index_search.aspx?clientID=20#
Reconnaître les déterminants	http://serious.gameclassification.com/FR/games/46068-Reconnaitre-les-determinants/index.html
check.io	https://checkio.org/
Color memory games	https://www.learninggamesforkids.com/memory_games/color-memory-game.html