Al-Azhar Engineering 16th International Conference

Vol. 19, No. 72, July 2024, 84 - 101

AN INTELLIGENT MODEL FOR AUTO GENERATING E-COURSES FROM TEXTBOOKS IN AN ADAPTIVE E-LEARNING ENVIRONMENT

Hager H. Fou'ad*, Shehab A. Gamalel-Din, Mohammed A. Elnabwy

Department of Computers and Systems Engineering, Faculty of Engineering, Al-Azhar University, Cairo, Egypt

***Correspondence:** hhfaams1996@gmail.com

Citation:

H.H. Fou'ad, S.A. Gamalel-Din and M.A. Elnabwy, "An intelligent model for auto generating e-courses from textbooks in an adaptive e-learning environment", Journal of Al-Azhar University Engineering Sector, vol. 19, pp. 84 - 101, 2024.

Received: 06 December 2023

Revised: 26 January 2024

Accepted: 01 February 2024

DoI:10.21608/auej.2024.252066.1498

ABSTRACT

Recent researches suggest the new model of adaptive learning that considers the interpersonal differences between learners when presenting a certain lesson for each individual student. However, teachers are resisting the new model under the pretext of not wanting to rebuild their lessons again to follow the recommendations of this new model. Therefore, the objective of this research is to overcome this resistance by automating the process of constructing adaptive lessons given a certain textbook. Therefore, this research follows a 2-step process: 1) identification and extraction of learning objects from the textbook, 2) composing the lessons using those extracted objects. Therefore, this research developed a YOLOv5 deep learning model, for book's elements identification, using a dataset of about 70 books, references and school textbooks, containing over 6000 pages in both English and Arabic languages of various specializations and knowledge domains. The developed model has an accuracy of about 92%. Those identified learning objects are then compiled into lessons with the aid of a concept ontology that is automatically constructed from the book's table of contents.

KEYWORDS: Adaptive eLearning, Concept Ontology, Yolo Models, Transfer Learning, Adaptive eBook, auto-generation of lessons, Learning Objects

نموذج ذكي للتوليد التلقائي للمقررات الدراسية من الكتب الدراسية من الكتب المدرسية في بيئة التعلم الإلكتروني التكيفية

هاجر حمدي فؤاد*، شهاب أحمد جمال الدين، محمد عاطف النبوي

قسم النظم والحاسبات، كلية الهندسة، جامعة الأزهر، مدينة نصر، القاهرة، مصر.

***البريد الاليكتروني للباحث الرئيسي :** hhfaams1996@gmail.com

الملخص

توصي الأبحاث الحديثة بنموذج تعليم جديد، "التعلم التكيفي"، الذي يأخذ في الاعتبار الاختلافات الشخصية بين المتعلمين عند تقديم درس معين لكل طالب على حدة. لكن المعلمين يقاومون النموذج الجديد بحجة عدم رغبتهم في إعادة بناء الدروس من جديد طبقا لتوصيات هذا النموذج. ولذلك، فإن الهدف من هذا البحث هو التغلب على هذه المقاومة من خلال أتمتة عملية بناء دروس

تكيفية من واقع الكتاب المدرسي. انتهج هذا البحث عملية ذات خطوتين: 1) تحديد واستخراج الكائنات التعلمية (learning objects)، من الكتاب المدرسي، 2) تكوين الدروس باستخدام تلك الوحدات المستخرجة. ولذلك، قام هذا البحث بتطوير نموذج التعلم العميق، باستخدام YOLOv5، لتحديد عناصر التعلم وعناصر الكتاب، باستخدام مجموعة بيانات (dataset) تضم حوالي 70 كتابًا ومرجعًا وكتابًا مدرسيًا، تحتوي على أكثر من 6000 صفحة باللغتين الإنجليزية والعربية في مختلف التخصصات والمجالات المعرفية. بلغت دقة النموذج الذي تم تطويره حوالي 92٪. يتم بعد ذلك تجميع وحدات التعلم التي تم استخراجها من الكتاب في دروس بمساعدة اونتولوجي المفاهيم (Concept Ontology) الذي يتم إنشاؤه تلقائيًا من جدول المحتويات الخاص بالكتاب.

الكلمات المفتاحية : التعليم الإلكتروني التكيفي، اونتولوجي المفاهيم، نماذج اليولو، الكتب التعليمية التكيفية، الكائنات التعليمية، أتمتة إنتاج الدروس.

INTRODUCTION

The epidemic of COVID-19 has proved the importance of e-Learning as an essential educational method. However, e-Learning, as it is today, inherits many of the problems of traditional education systems, though had solved many others. Among those inherited problems is the "one size fits all" teaching model. Recent researches suggest the new model of adaptive learning that considers the interpersonal differences between learners when presenting a certain lesson for each individual student.

In fact, one-to-one teaching (private tutoring) is the most effective learning approach [1] because a human teacher is smart enough to adapt his teaching approach in real time based on each individual student's psychology and skills, among other traits. Unfortunately, he cannot do that for a large group of students, as it requires unaffordable efforts [2].

In the adaptive model, a lesson is composed by bringing together a set of small-granularity Learning Objects (LOs) in a different way for the different students to meet their learning models. Noteworthy, in the traditional environment, instructors compose their lessons/lectures similarly by collecting several extracts (analogous to LOs) from existing large-granularity traditional contents, such as textbooks; a process that almost each instructor/teacher follows unintentionally without knowing that this is an essential step in the adaptive process. They, for example, use the elements of a textbook, such as examples, definitions, explanations, exercises, etc. to compose their lessons/lectures, though as one-fits-all, while an adaptive engine takes those same LOs and present them differently to the different students, provided they are put in the proper format.

In fact, researchers in the field of adaptive learning are concentrating more on personalized lessons, therefore one goal of our research is to widen this area to cover personalizing eBooks as well. To explain, instead of printing one single e-Book edition for all students, it is suggested to develop a different personalized e-book version for each student.

A personalized eBook is a version of the textbook where the book is presented in a style that is more suitable to the learning model of the specific student [2]. In other words, each eBook version would look differently for the different students, though contains all/most of the elements of the original textbook while maintaining its objectives. Please note that the student's learning model describes the learning characteristics of the student via a set of parameters, among them is the learning style that determines the student preferences while learning. This research adopts the Felder-Silverman Learning Style Model (FSLSM) [11].

To explain, as depicted by Figure 1, the textbook is split into its primitive elements that are then recomposed differently for each student version. Examples of such book elements are: section, text, image, diagram, graph, table, Table of Contents (TOC), reference, index, title, caption, header, box, etc. These book elements are then processed to create the LOs. Knowing the learning model of the student, the adaptive engine selects the appropriate LOs and order them according the learning style of the student. Although the details of the adaptive engine and the adaptive process are outside the scope of this article, it is apparent from the figure that in order to recompose the book into an adaptive one, it has first to be decomposed into small granularity LOs—a critical process of this research and is central in this article.

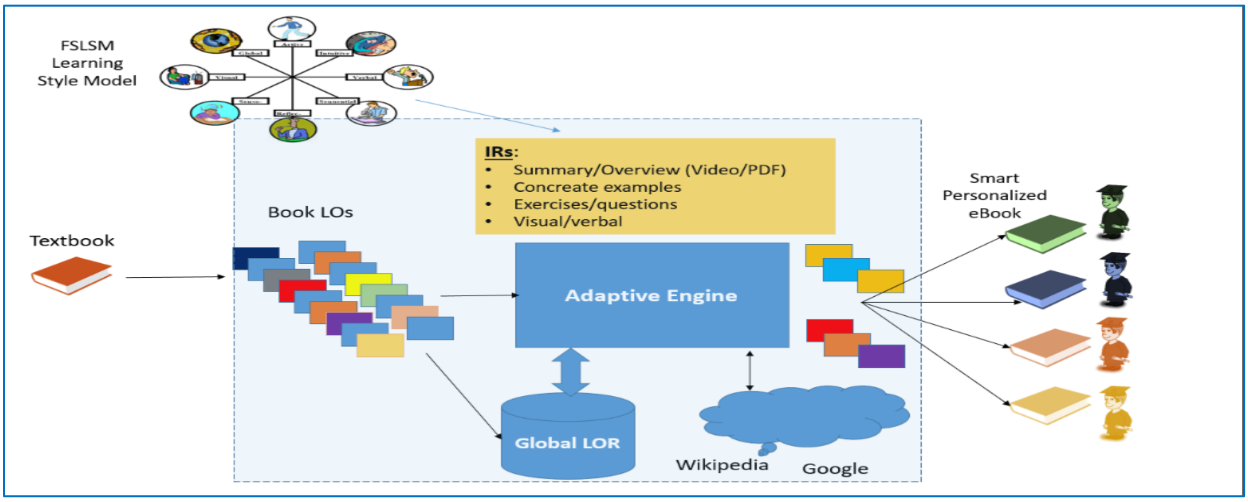


Figure 1. From a Textbook to Personalized eBooks.

However, the experience in deploying adaptive e-Learning systems at universities and schools (e.g., at Al-Azhar University, during the covid-19 pandemic) declared that instructors, especially schoolteachers, resist to redesign their lessons according to the new technology. Therefore, the objective of this research is to automate, as much as possible, the process of constructing e-Learning lessons given a specific textbook, as depicted by Figure 2. In fact, researches in this field focused on adaptive lessons and none had touched the problem of constructing adaptive e-Learning courses from textbooks.

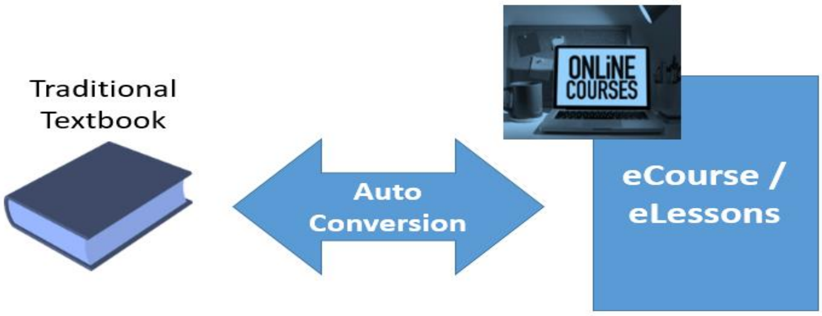


Figure 2. Auto Generation of eCourses/eLessons out of Textbooks.

Noteworthy, the Textbook-to-Lesson automation process requires two main constituents:

- The lesson structure, which can be inferred from the book's TOC,
- The small granularity LOs from which the lesson can be composed (e.g., explanation, examples, equations, experiments, questions/exercises, etc.), which should be extracted from the textbook manuscript.

Identifying those two constituents from a textbook, are two concerns of this research. For the first concern, a dataset was developed from about 70 books, references and school textbooks, containing over 6000 pages in both English and Arabic languages of various specializations. This dataset was used to train a model using YOLOv5 Algorithm for the detection of book elements. The developed model has an accuracy of about 92%.

For the second concern, a concept ontology was constructed from the extracted book's TOC, and was then used to shape the course and its lessons' structure, while those other extracted book's LOs are used to compose the lessons.

Fine-tuning the identification of the book's elements is still undergoing to improve the accuracy and to determine the LO's metadata attributes. Moreover, currently, the lessons are composed according to a predefined standard template; working on adaptive lessons is a future work.

In this article, Section 2 reviews few related work, Section 3 explains the proposed model of mapping a textbook into an adaptive lesson, as well as how the textbook constituents can be identified and used in the lesson construction process. Section 4 discusses the deep learning training process to obtain a model for identifying the different blocks of a textbook, while Section 5 highlights discusses the implementation and the algorithms and the rules used. Section 6 concludes.

2. Related Work

2.1. Learning Objects Extraction and Annotation

In e-Learning, learning materials are composed of multimedia objects referred to as Learning Objects (LOs) [3]. An LO is a self-contained learning contents, such as explanation, experiment, example, etc. An LO is composed of two important parts, namely the learning contents itself and the metadata describing the learning object. Metadata is used for the proper identification and selection of the appropriate object. Metadata is developed to enhance the learning object reuse and retrieval.

Recently, several researches have focused on the segmentation, extraction and annotation of smaller granularity Micro Learning Objects (MLOs), especially video lectures. These researches can be categorized according to their segmentation approach, e.g., text-based segmentation or motion-based segmentation.

For motion-based segmentation, Ma and Agam [4] segmented a video into various scenes by identifying frame transitions and analyzing color histograms, while Ma and Zhang et al [5] have later advanced this work to fit with video lectures and proposed an automatic video indexing framework that compares lecture slide images with candidate video frames using Boosted deep neural networks.

For text-based segmentation, Atef et al [3] proposed a framework that is designed to handle both PowerPoint Presentation lectures and PowerPoint-based video lectures. They used a hybrid approach in which a deep learning model is developed and used to segment new learning material into MLOs. The resulted segmentation is then tuned using a proprietary set of rules of their own research.

Mohammed and Dimitrova [6] presented a generic ontology-underpinned framework, called VISC-L, which uses video transcripts to segment, characterize and link videos to the domain knowledge covered in the segments. They use existing knowledge models, in the form of a domain ontology, to identify the domain concepts as well as use the ontology hierarchy and a language model based on BERT to identify focus topics and concepts for each video segment.

Noteworthy, none of these researches worked on textbooks, which is the main concern of this research.

2.2. eBooks

It is becoming evident that eBooks will be beneficial to the delivery of eLearning than traditional texts as they better suit the study style of a predominantly millennial student generation [7] who can study at their own pace and convenience [8]. However, although much attention has been given to the functional advantages of eBooks, there is still limited understanding as to how the information in them should be displayed to maximize learning effectiveness [9].

The major issue with current eBooks is that they are monotonous with long paragraphs of text and sizable images. More importantly, they do not employ the multimedia to its maximum capacity to enhance learnability. The newer generations of readers used to visual-based learning as these resources engage with them better. This led to the rise of interactive eBooks [10].

Noteworthy, none had worked on adaptive eBooks, which makes this research novel.

3. Mapping Textbooks onto eCourses and vice versa—The Model

This section discusses the mapping of the different textbook's elements onto an eCourse's elements. The identification and extraction of the textbook's primitive elements are discussed in full details in Section 5, while Section 6 discusses the algorithms used to compose LOs out of those primitive elements. Please note that the details of the adaptive engine for either Adaptive eCourses or Adaptive eBooks is outside the scope of this article.

Figure 3 depicts the relationship between a textbook and an eCourse. The LOs that were extracted from the textbook are recompiled to construct an eCourse. However, in eCourses, not only explanation LOs (that are mostly the book LOs) but also many other components are as important for the study process, especially if the eCourse would follow Bloom's course design recommendations [1], e.g., terminology definitions and drills/quizzes.

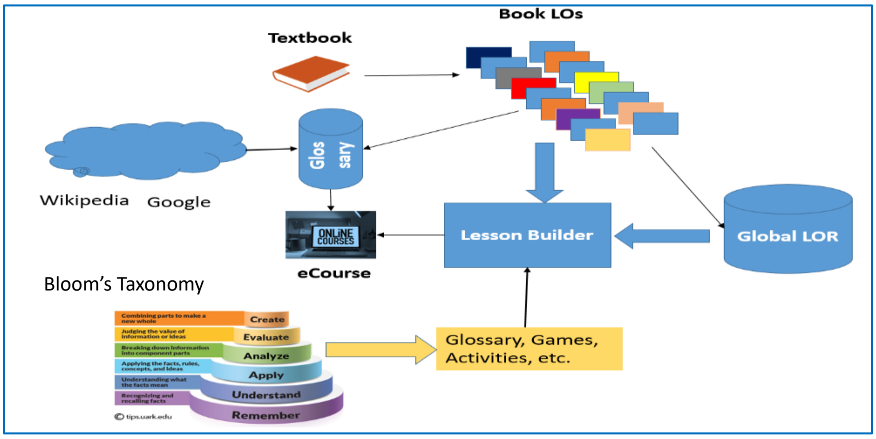


Figure 3. From Textbook to eCourse.

The two following subsections shed lights on the mapping of a textbook onto lessons from two different perspectives: mapping the structure and mapping the contents.

3.1. Mapping the Structure of a Textbook into the Structure of Lessons

A textbook is composed of chapters, each of which is divided into sections and subsections. In many cases, these chapters may be combined into a single title—a unit. In fact, the Table of Contents (TOC) of the textbook usually reflects the book and its chapters' structure. On the other hand, a course is usually composed of lessons and activities (e.g., experiments, exercises, quizzes, etc.), as shown in Figure 4.

A lesson is usually divided into topics and subtopics that explain or belong to a main topic/concept. A lesson may also contain other elements besides explanation, e.g., overview, questions, examples, etc. (IRs). In an adaptive environment, a lesson is composed of a set of LOs, each of which has its own IR. These LOs will be ordered differently for the different students according to their learning style. Figure 5 shows a sample lesson composed of a set of LOs corresponding to the elements of a book page, while Algorithm 1 highlights the important elements of the textbook that are involved in the process of auto generation of eCourses/eLessons. However, the algorithms of extracting LOs from textbook's elements are discussed in more details in Section 6.

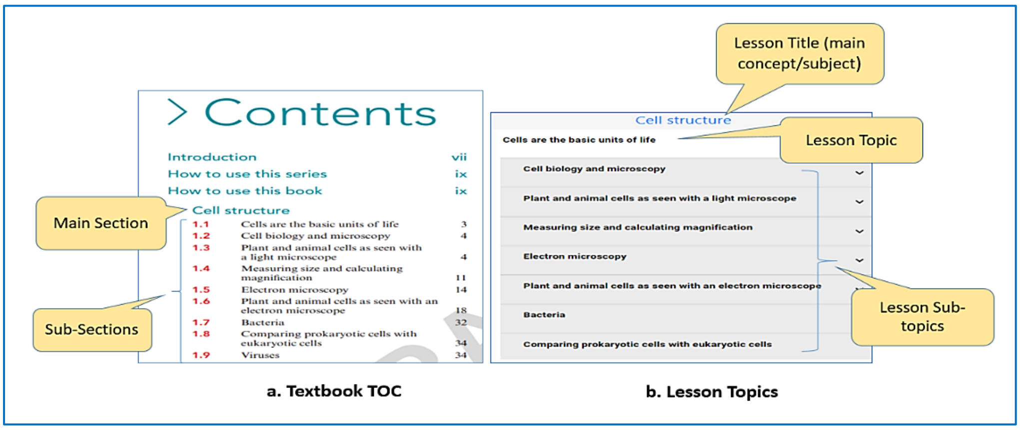
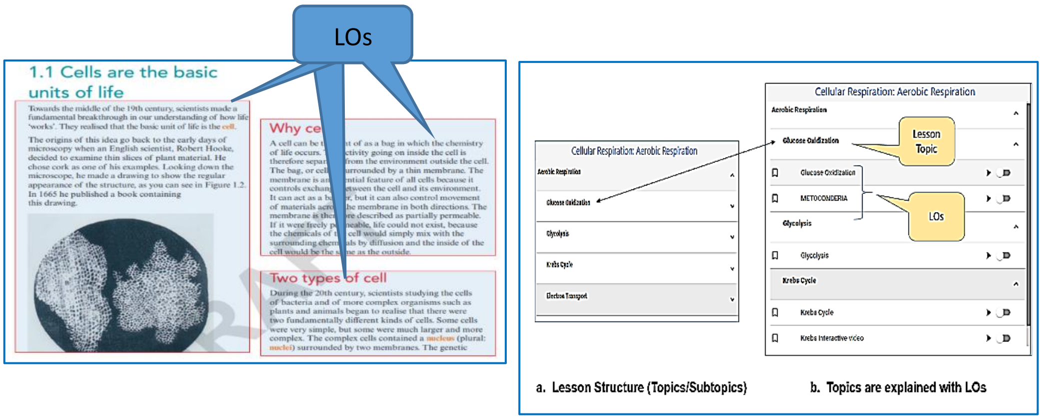


Figure 4. Textbook's TOC vs. Lesson's Structure.



a. Textbook's Sections/Sub-sections are Texts and Figures

b. Lesson Topics are taught in Terms of LOs.

Figure 5. Textbook Pages vs. Lesson Topic/Sub-Topics.

Algorithm1: Elements identification & Extraction

Input: textbook page images

Output: Book elements

Steps:

1. Identify and extract the book elements (see Section 5)
2. Create/update the Concept Ontology (CO):
   a. From the book TOC, create the draft CO (first PartsOf level)
   b. Analyzing the book structure (sections/subsections) to refine the CO by adding deeper PartsOf levels
3. Identify and extract LOs (see Sections 4 & 5)
4. Identify other important book elements:
   a. Identify keywords and extract their definitions:
      - Parse the Keywords sidebox (if exists)
      - Locate the keywords & definitions from the text itself
   b. Identify & extract Questions & exercises.

Algorithm1. Textbook's Elements identification & Extraction.

3.2. Mapping the Elements of a Textbook into the eLesson Contents

Two important components in the Knowledgebase of an adaptive system, namely, the LOs Repository and the Concept Ontology. The details of identifying and extracting them from a textbook is left to Sections 5 and 6. However, this section explains how these two components are involved in the auto construction of a Lesson. Noteworthy, the Concept Ontology is necessary only if the system supports automatic construction of lessons, as it is the case in this research. The most important relationship, to support this feature, is the PartsOf relationship that relates a main concept with its detailing concepts that may be involved in constructing a lesson explaining this main concept.

Figure 6 demonstrates the mapping between the Concept Ontology and the TOC of a textbook, at least for the shallow levels of the ontology. Deeper levels may be extracted from the sub-headers of the manuscript of the textbook, as can be recognized by the sample book page of Figure 8 part a. Figure 7 depicts the steps of generating lessons by analyzing a textbook, while Algorithm 2 highlights the necessary steps of constructing an eCourse.

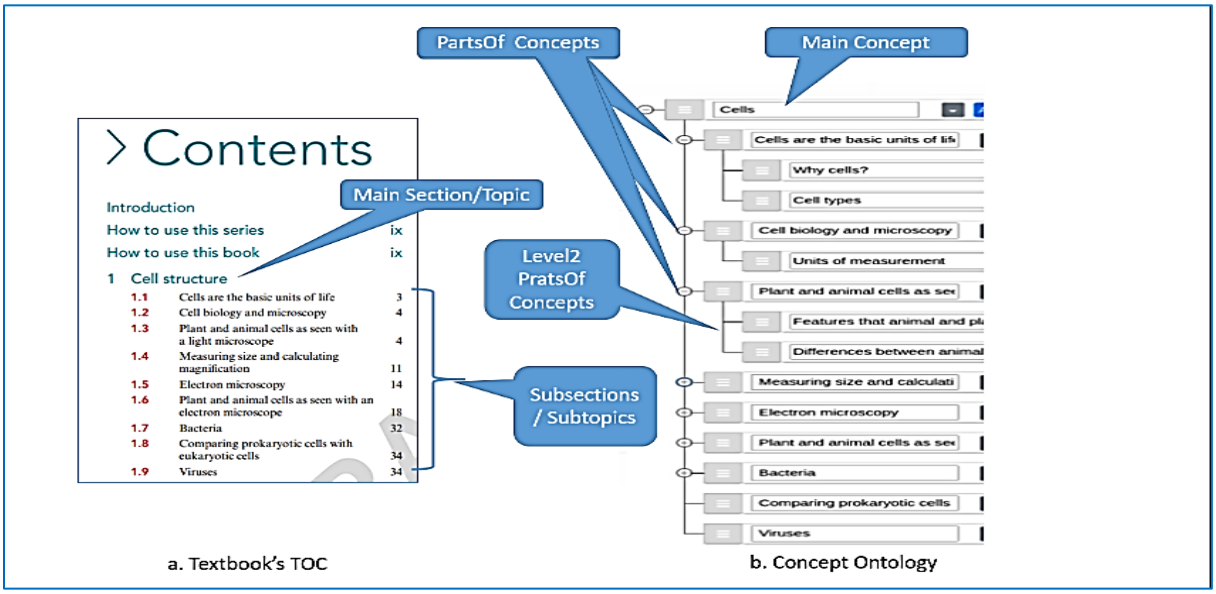


Figure 6. Textbook's TOC vs. Concept Ontology.

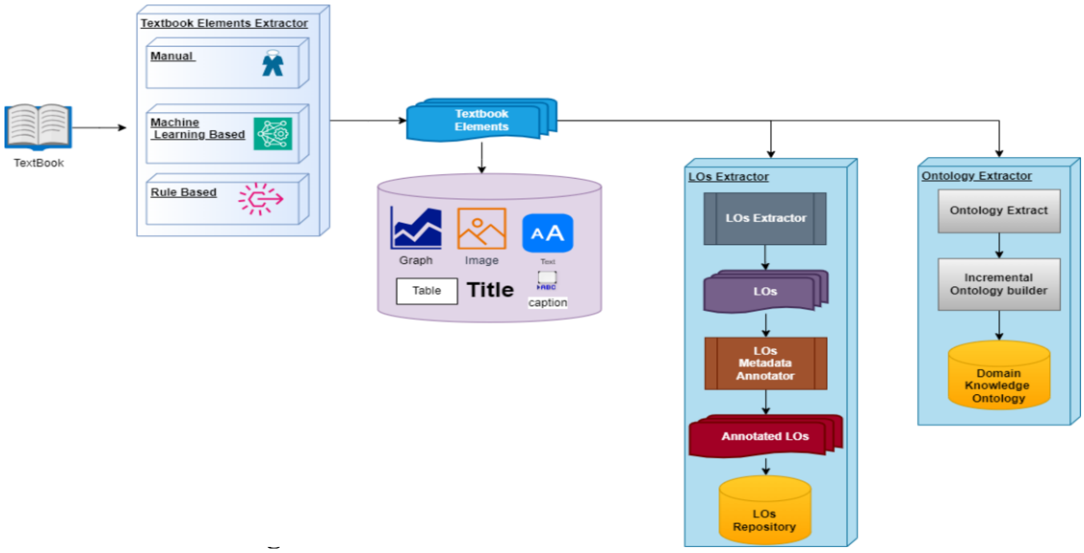


Figure 8. High-level architecture of the LO Extractor Framework.

Algorithm2: Auto eCourse Generation

Input: Textbook's LOR, TOC, Local CO, Other book elements (e.g., Keywords)

Output: eLesson

Steps:

1. Build the Lesson structure (hierarchy) from the appropriate CO
2. For each Topic & Subtopic use the appropriate LOs:
    a. Use Textbook's LOR as the main repository of LOs
    b. For adaptive books (needed may be LOs that are not in the textbook's LOR, Then, use the Global LOR
3. Build the lesson Glossary (synchronize with Global Glossary)
4. Build the lesson's tab of "questions/Activities" as in the textbook

Algorithm2. Auto eCourse Generation.

4. Identification of Book Elements

4.1. The Proposed Model

Figure 8 draws the framework of the LO extractors and their interconnections. The framework consists of three main components. The first is the "Textbook Elements Extractor" that is responsible for the extraction of the different elements of a given textbook. It is worth noting that the given textbook may be in one of many possible formats, e.g., scanned or editable formats. This section discusses the different methods (manual, deep learning-based model, and specially designed software programs) that were used in this research to identify the different elements of a textbook page, such as such as Image, text, title, graph, table, etc.

Those extracted book elements are then fed to the second component—the "LO Identifier"—that is responsible for tying those related elements to compose meaningful and atomic LOs.

The third component of framework is the "Ontology Extractor". More on this step is discussed in Section 6.

The following sections shed lights on the methods used in extracting the different textbook elements from a given textbook.

4.2. The Manual Method

Figure 9 illustrates the pipeline of the required steps for extracting Textbook elements from a given textbook.

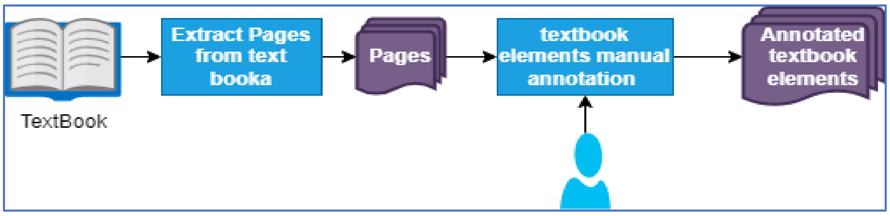


Figure 9. The Manual Labeling of the Textbook Elements.

The first step is extracting the pages of the given textbook. The extracted pages are then annotated manually in the second step resulting in annotated textbook elements. The annotated book elements, for at least a full section, are those passed to the second step of the framework—the "LO Extractor". Figure 10 shows examples of different annotated textbook elements that were manually annotated. To do this task manually for each page of a given book is a tedious and a time consuming task; therefore, a machine learning-based method is thought as described in the next section.

4.3. The Machine Learning-based Method

The YOLO deep learning algorithm is used to identify and annotate different textbook elements, as shown in Figure 11.

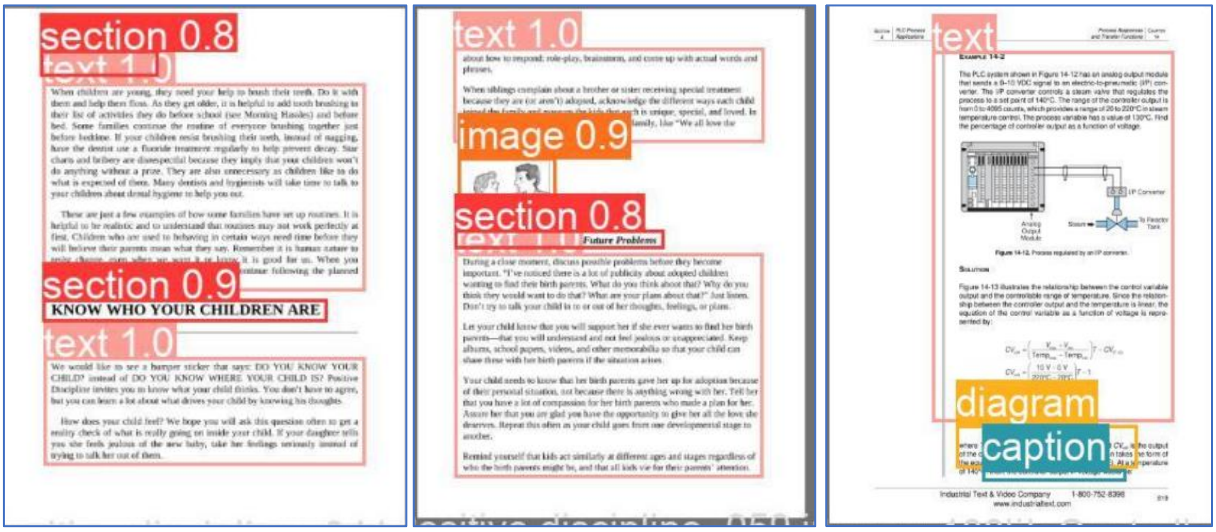


Figure 10. Examples of Annotated Textbook Elements.

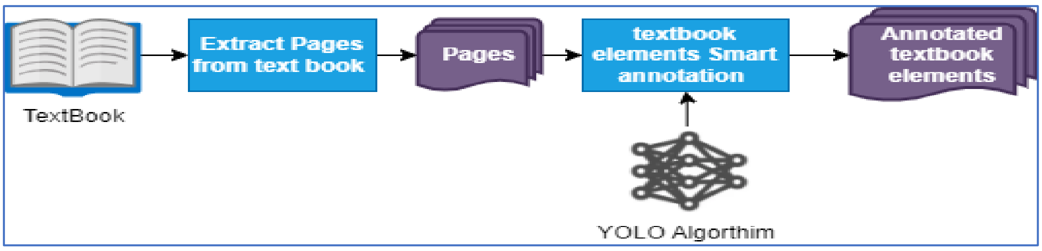


Figure 11. The Machine Learning-based method for labeling the book Elements

The process begins with creating a reasonable size dataset of annotated book pages as depicted in Figure 10, i.e., by using the same approach followed by the manual annotation process. In this research project, the created dataset was of size 17583 pages collected from 68 textbooks of various specializations in both languages of English and Arabic. This dataset was first preprocessed and prepared for the deep training process. This preprocessing step focusses on data normalization, shuffling, and resizing of those input images of the sample textbook pages. The preprocessed dataset was then split into two portions, one for transfer training the YOLOv5 pre-trained Deep Neural Network model [13], while the second was for testing the obtained model. This process is described in Figure 12.

The implementation used Google Colab [14] with a K80 GPU and 12 GB memory. The proposed framework is implemented using Python on a Linux platform with TensorFlow and Keras.

Model Evaluation

Figure 13 shows an example of different annotated book pages annotated using the resulted YOLO Model.

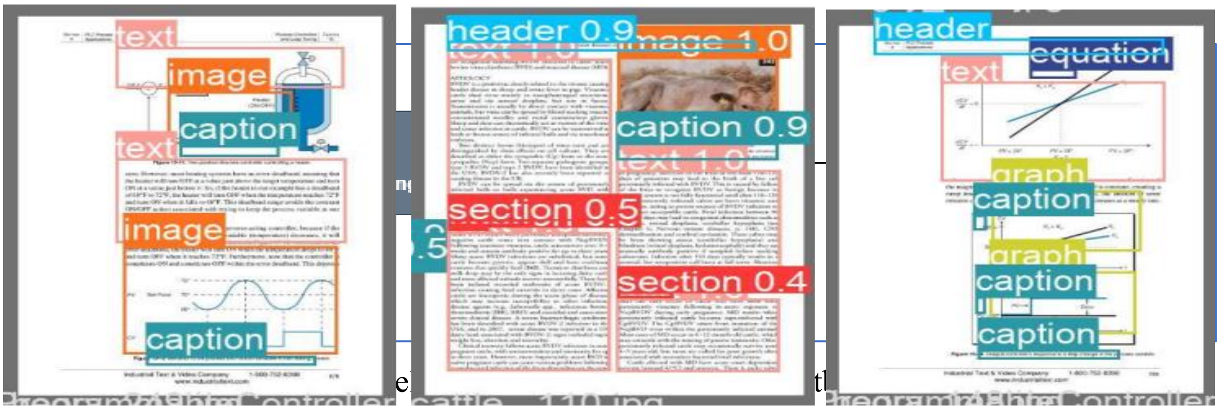


Figure 13: Sample Book Pages with the Elements Annotated using the Resulted Model.

Figure 14 shows the values of the common evaluation metrics [15]: the Mean Average Precision (MAP), the Precision, the Recall, and the F1 Score. Figure 15 shows the Confusion Matrix presenting the values of True Positives (TP), False Positives (FP), and False Negatives (FN) for each class by comparing the predicted bounding boxes with the ground truth annotations.

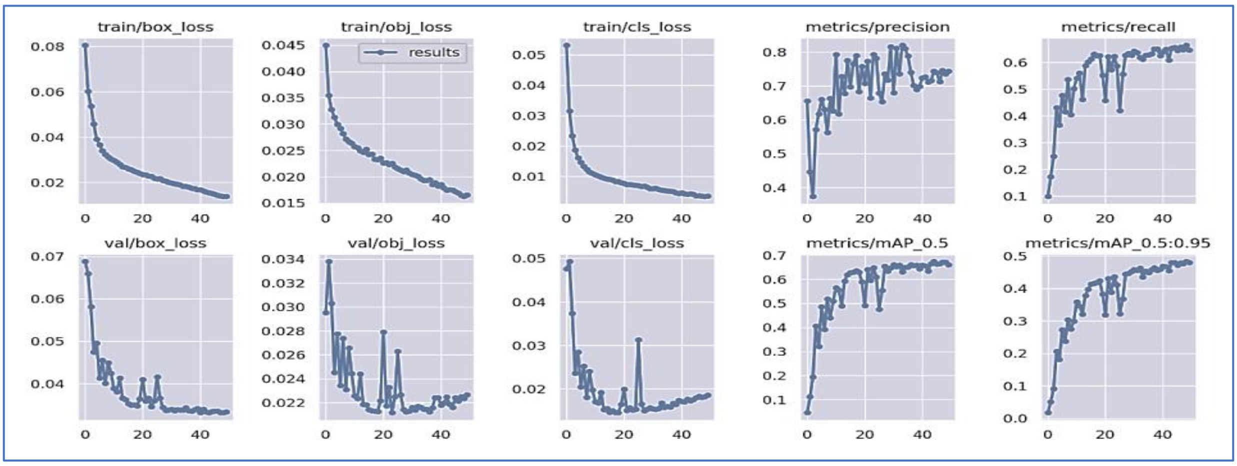


Figure 14. Textbook Elements Identification YOLO Model MAP, Loss, F1, Precision and

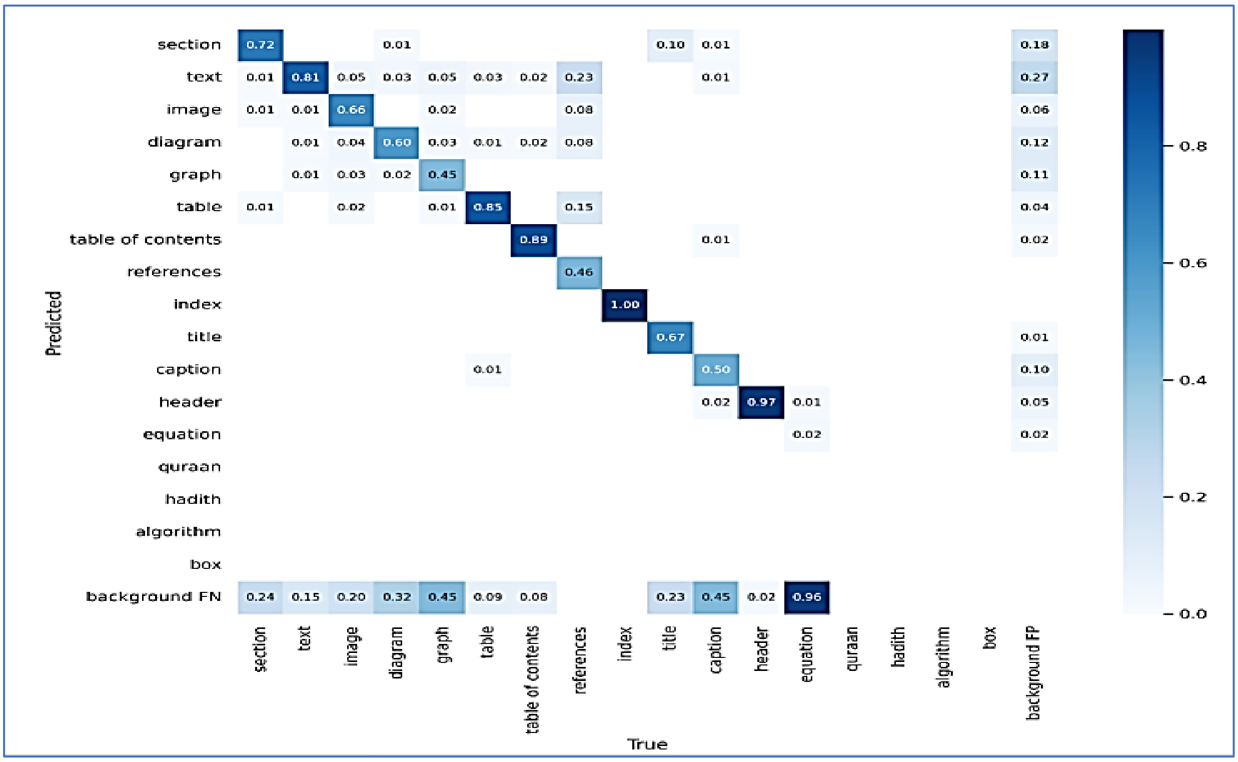


Figure 15. The Confusion Matrix for Textbook Elements Identification YOLO Model.

4.4. Identification of the Book Elements using Programming with Python

The first two element identification methods (as described above) are applied when the textbook is in a scanned (image-like) format. However, in many cases the PDF textbook is originated from a rich text format, in which case, the elements can be identified by analyzing the book pages using specially designed software programs.

PyMuPDF [16] is a high-performance Python library of functions to reliably and accurately extract text paragraphs, images/figures, text blocks/text boxes, etc. In addition, it offers functions to identify font information for each single text character in a PDF document. Figure 16 sheds lights on the multi-layered dictionary as an output of this library. Figure 17 interprets such an output on some textbook page as 4 text blocks and an image block, each of which contains information like writing direction, writing mode, color, font name, font properties and position information.

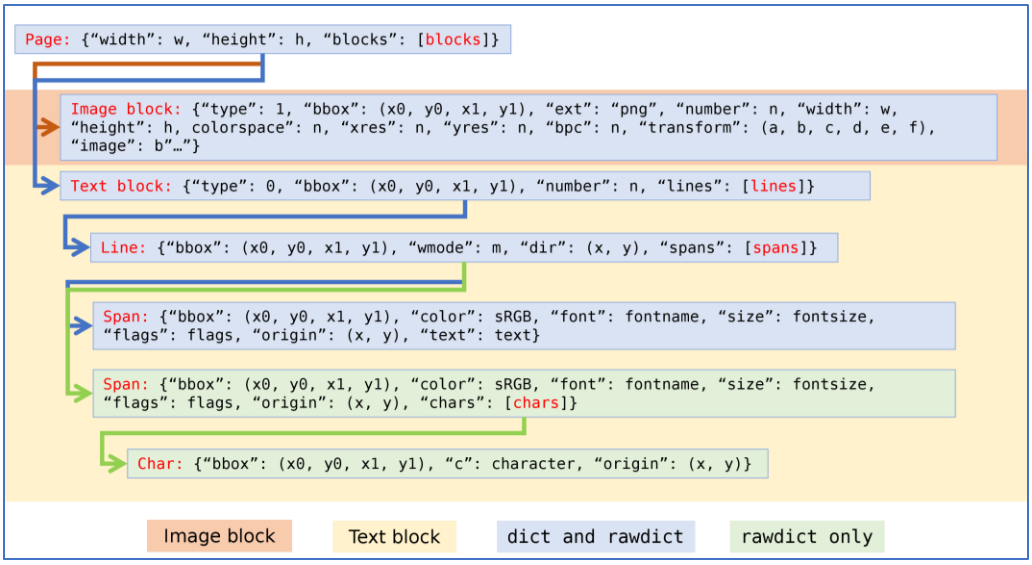


Figure 16. The TextPage Dictionary Structure.

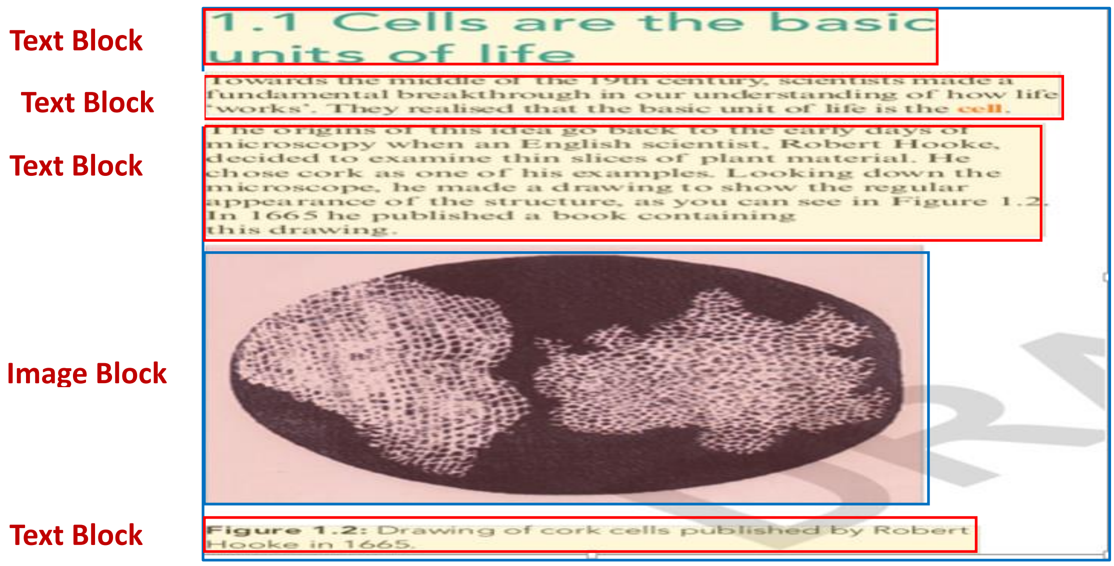


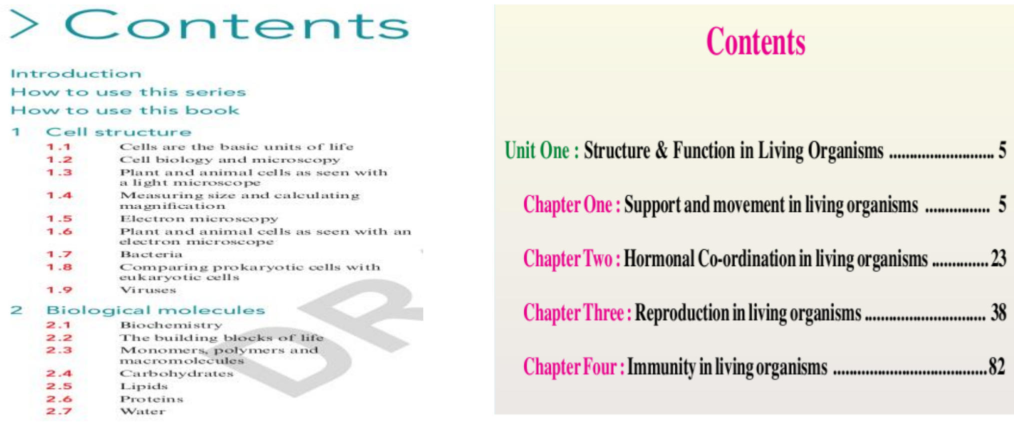
Figure 17. A Sample Textbook Page as split into its Constituent Blocks.

5. Implementation Algorithms

This section sheds lights on some critical implementation algorithms for building the Table of Contents (TOC), and hence the Concept Ontology, and the identification of some types of LOs, e.g., LOs centered on figures.

5.1. Extracting the Table of Contents (TOC)

Revising several textbooks revealed that there are different types of TOCs, e.g., giving either one or two levels of the chapter structure in terms of its sections and subsections, as shown in Figure 18.


> Contents

Introduction
How to use this series
How to use this book
1 Cell structure
1.1 Cells are the basic units of life
1.2 Cell biology and microscopy
1.3 Plant and animal cells as seen with a light microscope
1.4 Measuring size and calculating magnification
1.5 Electron microscopy
1.6 Plant and animal cells as seen with an electron microscope
1.7 Bacteria
1.8 Comparing prokaryotic cells with eukaryotic cells
1.9 Viruses
2 Biological molecules
2.1 Biochemistry
2.2 The building blocks of life
2.3 Monomers, polymers and macromolecules
2.4 Carbohydrates
2.5 Lipids
2.6 Proteins
2.7 Water

Contents

Unit One : Structure & Function in Living Organisms 5
Chapter One : Support and movement in living organisms 5
Chapter Two : Hormonal Co-ordination in living organisms 23
Chapter Three : Reproduction in living organisms 38
Chapter Four : Immunity in living organisms 82

a. Two-Level TOC b. Single-Level TOC

Figure 18. Samples of Different Types of Textbook TOCs

A TOC is usually found either at the beginning of the textbook before all chapters or at the beginning of each chapter. It usually includes the titles of the first-level headings, and sometimes the second-level headings (sections or A-heads), as shown in Figure 18. In most cases, third-level headings (subsections or B-heads) are implicitly included within the section text, and hence requires special analysis to extract. Algorithm 3 highlights the high level outlines of the algorithm used to identify the 3-level TOC.

```
Algorithm: Identifying B-heads

Inputs:
    -   eTextbook blocks (text_blocks)
    -   ToC n-level blocks (toc_blocks)
Output: n+1-level blocks
Steps:

// Identify common font size of text blocks.
Declare array variable all_font_sizes
Declare variable common_font_size
for counter = 0 to text_blocks.length - 1:
    all_font_sizes.push(text_blocks[counter].type)

set common_font_size = common value of all_font_sizes

// Search for text blocks containing n-level headings listed in the ToC.
for counter = 0 to toc_blocks.length - 1:
  Declare first_block = toc_blocks[counter]
  Declare second_block = toc_blocks[counter + 1]
  for counter2 = 0 to text_blocks.length - 1:
    if (text_blocks[counter2].text == first_block.text):
      // Identify n-level headings font size.
        Declare variable heading_font_size = text_blocks[counter2].font_size
        for each coming block repeat
          if (current_block.font_size > common_font_size && current_block.font_size <
        header_font_size):
             output.push(current_block)
      until text_blocks[counter2].text == second_block.text)
```

Algorithm 3. Identifying B-Heads TOC Elements

5.2. Extracting Learning Objects (LOs)

As mentioned in Section 4, LOs are at the center of the process of generating adaptive lessons; hence, the extraction of LOs given a textbook is crucial. One key determinant of an LO, is the detection of a figure block as one of the book elements (see Section 5), which spots an LO candidate to focus on. The task turns into identify compiling this figure (and in more complex cases, several related figures) and its corresponding textual explanation in the textbook as one LO. Algorithm 4 highlights the high-level outlines of extracting those LO centered on figures. Noteworthy, several other types of LOs are treated similarly, e.g., those centered on tables, equations, graphs, formulas, etc.

```
Algorithm: Extracting Figure-Centered LOs

Inputs:
        // Text blocks start with the prefix Fig, FIG or figure followed by one or more digits.
    -   caption_blocks
    -   image_blocks = [ ]
    -   other_blocks = [ ]
Output: Learning objects of type figure
Steps:
for counter = 0 to caption_blocks.length:
  // Identify figure related to each caption.
  for counter2 = 0 to image_blocks.length:
    if (image_blocks[countrt2].coordinates.y1 > caption_blocks[countrt].coordinates.y1):
      Declare variable figure_explanation_text = ""
      // Identify figure explanation text.
      for counter3 = 0 to other_blocks.length:
        // Search for paragraph that contains figure number (paragraph n).
        if (other_blocks[counter3].text.includes(caption_blocks[counter].caption_number)):
          figure_explanation_text += other_blocks[counter3].text
          // Measure sentence similarity (using sBERT model [17]) between:
          Declare similarity_with_p1 = calc_similarity(other_blocks[counter3],
other_blocks[counter3 - 1])
          Declare similarity_with_p2 = calc_similarity(other_blocks[counter3],
other_blocks[counter3 + 1])
          if (similarity_with_p1 > similarity_threshold):
            figure_explanation_text += other_blocks[counter3 - 1].text
          if (similarity_with_p2 > similarity_threshold):
            figure_explanation_text += other_blocks[counter3 + 1].text
          From figure, caption and explanation text, create learning object. Put LO in proper format
          Identify LO title, concept, and instructional role using SBERT and summerization algo.
```

Algorithm 4. Extracting Figure-Centered LOs.

This article described the novel idea and its implementation of automatically generating e-Lessons directly from textbook chapters. The employed algorithms use deep transfer learning using Yolov5 for the identification of the different elements of the textbook, from which the learning objects as well as the concept ontology are then identified and constructed with the aid of transformer deep learning sBERT.

In fact, this part as described in this article is a part of a bigger research project that targets the automatic conversion of the textbook into an adaptive eBook, and an adaptive e-Course as well.

1. 6. Conclusion & Future Work

This article described the novel idea and its implementation of automatically generating e-Lessons directly from textbook chapters. The employed algorithms use deep transfer learning using Yolov5 for the identification of the different elements of the textbook, from which the learning objects as well as the concept ontology are then identified and constructed with the aid of transformer deep learning sBERT.

In fact, this part as described in this article is a part of a bigger research project that targets the automatic conversion of the textbook into an adaptive eBook, and an adaptive e-Course as well.

References

[1] LW et al., A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. 2001.

[2] Shehab Gamalel-Din, "Smart E-Learning: A Greater Perspective, From the 4th to the 5th Generation e-Learning" In the Proceedings of the First International Conference on Operations Research and Applied Informatics (CORAI 09), Cairo, Egypt, 15 – 19 February, 2009

[3] Mohammed Atef Elnabwy, "Text Analysis for Learning Objects Identification," Faculty of Engineering, Al-Azhar University, Cairo, 2022.

[4] D. Ma and G. Agam, "Lecture video segmentation and indexing," C. Viard-Gaudin and R. Zanibbi, Eds., Jan. 2012, p. 82970V. doi: 10.1117/12.910594.

[5] D. Ma, X. Zhang, X. Ouyang, and G. Agam, "Lecture Vdeo Indexing Using Boosted Margin Maximizing Neural Networks," in 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, Dec. 2017, pp. 221–227. doi: 10.1109/ICMLA.2017.0-155.

[6] Mohammed and V. Dimitrova, "Video Segmentation and Characterisation to Support Learning," 2022, pp. 229–242. doi: 10.1007/978-3-031-16290-9_17.

[7] Casselden and R. Pears, "Higher education student pathways to ebook usage and engagement, and understanding: Highways and cul de sacs," Journal of Librarianship and Information Science, vol. 52, no. 2, pp. 601–619, Apr. 2019, doi: 10.1177/0961000619841429.

[8] J. Akeroyd, "EBOOKS AND ELEARNING: A NEW CHAPTER?," SCIRES-IT, vol. 2, p. 1, Jan. 2012.

[9] H. F. Wang and M. C. Chiu, "Ebook presentation styles and their impact on the learning of children," in Proceedings of the Interaction Design and Children Conference, IDC 2020, Association for Computing Machinery, Inc, Jun. 2020, pp. 438–443. doi: 10.1145/3392063.3394425.

[10] Sarasa-Cabezuelo, "A Model for Creating Interactive eBooks for eLearning," Future Internet, vol. 12, no. 12, p. 223, Dec. 2020, doi: 10.3390/fi12120223.

[11] E. M. Watson, "A comparative study of medical ebook and print book prices," Health Info Libr J, vol. 38, no. 1, pp. 39–48, Mar. 2021, doi: 10.1111/hir.12310.

[12] Ivan Krechetov, and Vladimir Romanenko. "Implementing the Adaptive Learning Techniques", 2020.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[14] E. Bisong, Google Colaboratory BT - Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners. 2019.

[15] L. LIU and M. T. ÖZSU, Eds., "Mean Average Precision," in Encyclopedia of Database Systems, Boston, MA: Springer US, 2009, p. 1703.

[16] McKie, M., Liu, R.: Pymupdf - python binding for mupdf (2020).

[17] Halawa, A., Gamalel-Din, S. ., & Nasr, A. (2023). EXPLOITING BERT FOR MALFORMED SEGMENTATION DETECTION TO IMPROVE SCIENTIFIC WRITINGS. Applied Computer Science, 19(2), 126–141. https://doi.org/10.35784/acs-2023-20