Automatic Generation of Learning Objects Using Text Summarizer Based on Deep Learning Models

Leandro Massetti Ribeiro Oliveira[1], Antonio José G. Busson[2], Carlos de Salles S. Neto[1], Gabriel N. P. dos Santos[2], Sérgio Colcher[2]

[1]Department of Informatics - Federal University of Maranhão

[2]TeleMídia Lab - Pontifical Catholic University of Rio de Janeiro Rio de Janeiro

Abstract. *A learning object (or LO) is an entity, digital or not, that can be used and reused or referenced during a technological support process for teaching and learning. Despite mainly being multimedia, with audio, video, text and images synchronized with each other, LOs can help disseminate knowledge even only in educational texts. However, creating these texts can be costly in time and effort, creating the need to seek new ways to generate this content. This article presents a solution for the generation of text-based LOs generated through summaries supported by Deep Learning models. The present work was evaluated in a supervised experiment in which volunteers rate computer educational texts generated by three types of summarizers. The results presented are positive and allow us to compare the performance of summaries as LO generators in text format. The findings also suggest that using post-processing in the output of models can improve the readability of generated content.*

1. Introduction

There are several ways to disseminate knowledge in education, from the most traditional, such as classroom learning with tutors in the classroom, even those that use technology to pass knowledge to the student. There are digital resources such as text, video, software, among others, that can be used and reused for education. These resources are called Learning Objects (LO) [Damasceno et al. 2020].

Creating these LOs is a manual process and requires an expert, which transforms the entire process onerous [Busson et al. 2017]. To produce a LO is required to have a minimum of knowledge in the subject area. With the advent of artificial intelligence methods, a viable solution would be to generate educational content using knowledge-ware text synthesizer models, facilitating the development of these educational resources.

Recent works focused on generating educational content are often limited in specific functionalities, such as: generation of questionnaires [Kurdi et al. 2020, Rocha et al. 2020], response generation [Li and Xing 2021], generation of math problems [Xu et al. 2021], level generation for educational games [Hooshyar et al. 2018]. Unlike these works, our proposal consists of a method that creates structured educational content. More specifically, given a subject, our algorithm extracts the subject topics and their respective contents from a knowledge base. Next, we use deep learning models to summarize the content of each topic. Finally, we structure the summarized text to compose the LO.

We validate our proposal through the evaluation with specialists, tutors, and students, to estimate the feasibility of using summarization models to generate texts

DOI: 10.5753/sbie.2021.217360

with acceptable content. In our experiments, we evaluated three different summarization models. In addition, we also tested the feasibility of using automatic translators for other languages since the selected summarization models are trained in the English language.

The remainder of this paper is structured as follows. We begin, in Section 2 by presenting the related works. In Section 3 we introduce our proposal for automatic LO generation, followed by Section 4 where we describe the experiments conducted to evaluate the effectiveness of our proposal. Finally, Section 5 is devoted to our final remarks and conclusions.

2. Related Works

According to Kurdi et al. [Kurdi et al. 2020], recent studies that focus on educational content generation use the assessment of expert professionals as a way to validate the generated content. This approach appears to be standard procedure and is often a good quality factor. However, they point out that the evaluation of professionals is only one factor. It is crucial to measure students' evaluation and demonstrate the practical usability, readability, and discernment of the generated content.

Rocha et al. [Rocha et al. 2020], for example, propose a method that explores structured educational bases in order to generate questions automatically. In experimentation, volunteer teachers validated the quality of 100 questions generated by their algorithm, achieving an average rating of 3.5 on a scale between 1 and 5.

Yang [Yang et al. 2013] argues that educational contents are more difficult to be absorbed when they are extensive. Thus, the author presents a solution for summarizing these educational contents to better adapt to a mobile environment. In the study, the author used questions to assess student learning through summaries generated by an automatic summarizer. As a result, the article shows that the methodology is efficient for the student to acquire knowledge and be better adapted in a mobile device environment.

Rudian et al. [Rüdian et al. 2020] present in their article a problem regarding the automatic generation of questions, since most of them work only on single input sentences. This limits the generation of good questions on a large scale. They then propose to use summarization models of texts in German for this task, comparing some used in the literature through the evaluation of 30 professors in the area. Thus, the LexRank algorithm showed the best performance results regarding the readability of the content, thus being able to be used as a parameter in the generation of questions.

Li and Xing [Li and Xing 2021] propose a method for natural language generation to support MOOC learners. They used the GPT-2 (Generative Pretrained Transformer 2) [Radford et al. 2019] model to provide students with informational, emotional, and community support with natural language generation on discussion forums. In experiments, they showed that GPT-2 model could provide supportive and contextual replies to a similar extent compared to humans.

Hooshyar et al. [Hooshyar et al. 2018] propose a data-driven PCG approach benefiting from a genetic algorithm and SVM (Support Vector Machines) to generate educational-game contents tailored to individuals' abilities automatically. In experiments, they showed that users realized more significant performance gains from playing generated content tailored to their abilities than playing uncustomized game content.

Xu et al. [Xu et al. 2021] present a generic approach for procedural generation of mathematical problems. Their generation process consists of two phases: abstract math problem generation and text generation. For the generation of abstract math problems, they propose a generic template-based method that operates across various difficulty levels and domains. Moreover, for text generation, they propose a multi-language adaptive textual content generation pipeline to realize the generated abstract math problems into semantically coherent text questions in natural language. In experiments with experts, they showed that the math problems generated by their approach are sensitive and resolvable for primary school students.

3. Method

This work aims to verify the feasibility of using summarizers for the generation of learning objects. As a method, it is intended to generate summaries of texts from Wikipedia database[1]. However, some limiting factors for using this approach, such as the lack of summarizers for languages other than English, create the need to use automatic translators. A factor is the word limit in the summarization models dictionary, which requires the text division into several parts to generate short summaries. Another limiting factor is the need for initial formatting of text from Wikipedia, such as image descriptions, references, and out-of-context links to the content.

The proposed method for LOs generation is illustrated in Figure 1, and its steps are described on the following:

1. Search, through API (Application Programming Interface), educational content from the Wikipedia database on a specific subject.
2. The content then goes through a cleansing process to keep only the textual part, removing special characters from Wikipedia, references to other articles, descriptions of images, and transforming lists into a large text.
3. The cleaned text goes through a structuring step, partitioning it into small texts of a maximum of 1100 characters, thus preventing the summarizer from extrapolating its modeling capacity.
4. Each partition is automatically translated into the English language (in which the summarizers operate) and finally goes through the summarizer, generating a template and summarizing each partition.
5. The last step translates them back to the target language and joins the partitions.

We used the googletrans[2] library to perform the translations, which implements an API for the Google translator. Summarizers come from the HuggingFace platform[3] which is an Open-Source platform with several text generation models and natural language processing.

From the HuggingFace platform, the three deep learning-based summarizer models most used by users were selected to summarize texts for comparison purposes.

- **Model 1 - sshleifer/distilbart-cnn-12-6**: Summarization model developed by Sam Shleifer that uses the distilled version of the BART model, the DistilBART [Lewis et al. 2019], with 174 thousand downloads.

[1]https://en.wikipedia.org/wiki/Wikipedia:Database_download
[2]https://pypi.org/project/googletrans/
[3]https://huggingface.co/

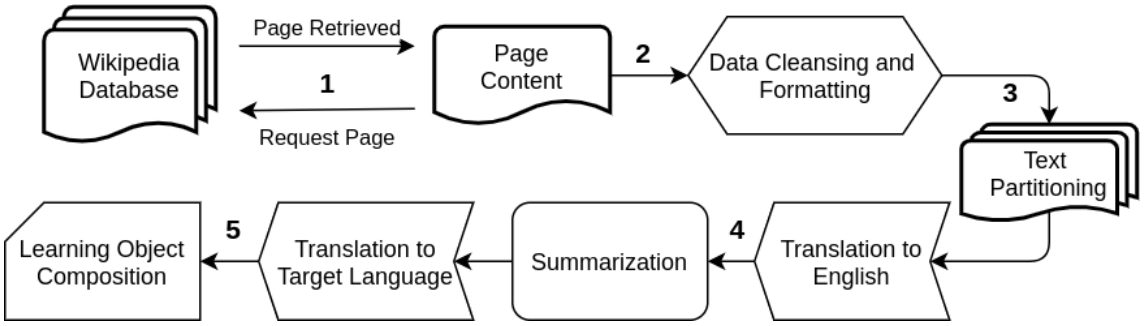

Figure 1. Steps of the proposed method for LO generation.

- **Model 2 - facebook/bart-large-cnn**: Summarization model developed by the Facebook team that uses the BART model [Lewis et al. 2019], with 139 thousand downloads.
- **Model 3 - google/pegasus-cnn_dailymail**: Summarization model developed by the Google AI team using the PEGASUS model [Zhang et al. 2020], with 55 thousand downloads.

4. Experiment

To verify the feasibility of using summarizers as learning objects, a subject from the computer science field was selected in order to assess the quality of its educational content and the quality of the grammar of the summarized text. The selected subject was from the Software page in Wikipedia[4], where the original text is divided into several topics and each topic has some lists of items, making it a good case study to carry out the summary. Our experiment was conducted with two groups of volunteers, the first composed only of students, while the other is composed of graduates and teachers.

In the form, volunteers evaluated the quality of the text generated for each of the five topics of the Software subject, resulting in 15 texts (5 for each model). The form initially consists of the user authorizing the free and informed consent term and filling in some personal information, especially their background. Then it continues with the evaluation of each of the 15 texts summarized, analyzing both its educational content and its grammar. These evaluations use a score from 1 to 5, with 1 being an imperfect text and 5 being a good text. The form did not show the label of each text to make it impossible to bias the participants. In addition, it was presented in random order.

4.1. Results

As the first result from the form, Figure 2 shows the evaluation chart of students in computing courses and related areas. Answers were collected from 12 students who are not yet graduated in the field. The Figure shows the average of each one's score, together with the confidence interval for the variance.

In terms of content, the summaries of Model 1 and Model 2 had similar scores and equal variance, having a better performance than Model 3 according to the participating students. They had an average of 3.35. As for the text's grammar quality, the students

[4]https://pt.wikipedia.org/wiki/Software

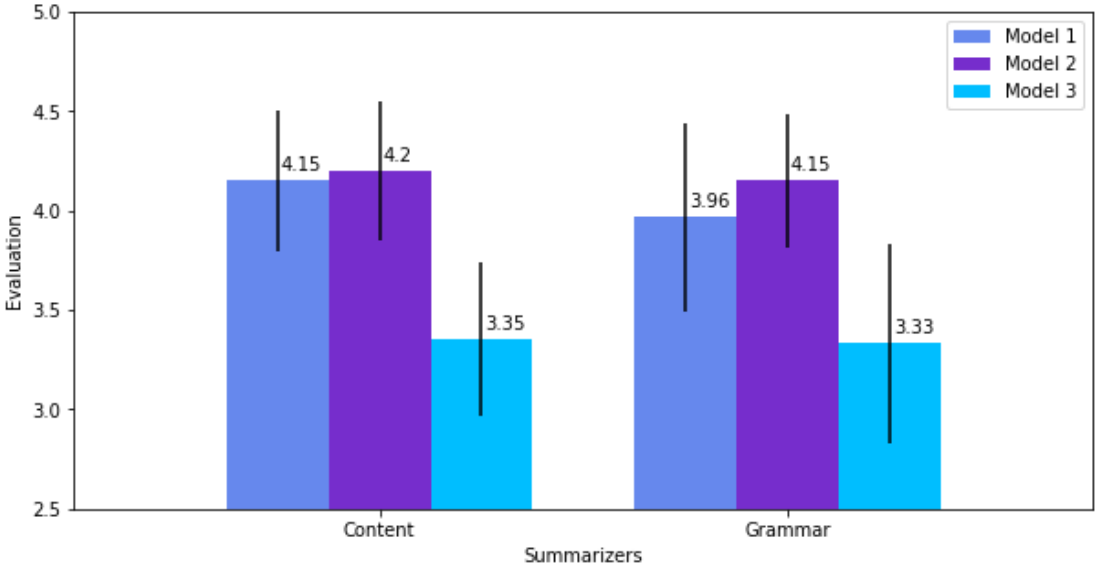

Figure 2. Student evaluation regarding the quality of the generated texts. On the left, the content evaluation and on the right, the grammar evaluation.

elected Model 2 as the best summarizer of educational content, standing out in terms of performance compared to the other two.

The students' justification regarding the evaluation of each summarizer, it is noted that the Model 3 summarizer had a lower performance because some parts of the text are disconnected and meaningless, in addition to having strings "<N>" dropped by the text, which could be suppressed. While the texts of Model 1 and Model 2 had fewer complaints in the justification field than Model 3. Table 1 and 2 present each student's vote on which would be the best and the worst Content Summarizer. It is possible to notice that students elected Model 2 as the best text in almost all cases. Only in text 3, Model 1 performed better than the others.

Table 1. Voting for the best text by students.

	Text 1	Text 2	Text 3	Text 4	Text 5	Total
Model 1	3	3	**8**	5	3	22
Model 2	**6**	**7**	4	**6**	**8**	**31**
Model 3	3	2	0	1	1	7

Table 2. Voting for the worst text by students.

	Text 1	Text 2	Text 3	Text 4	Text 5	Total
Model 1	2	3	2	2	5	14
Model 2	3	0	0	0	1	4
Model 3	**7**	**9**	**10**	**10**	**6**	**42**

As for the lower text, there was unanimity to elect the texts summarized by Model 3. However, the justifications given by the students were few, primarily referring to the text being confused. It is interesting to note that Table 2 also elects the texts from Model

2 as the best ones. Considering the students' observations, the reason seems to have been the least grammatically mistaken and more fluid reading for computer students.

The second audience used in the research is people who graduated in computing courses, post-graduate students, and professors in areas related to computer science. Totaling 13 answers, the results of the average of the evaluation are shown in Figure 3.

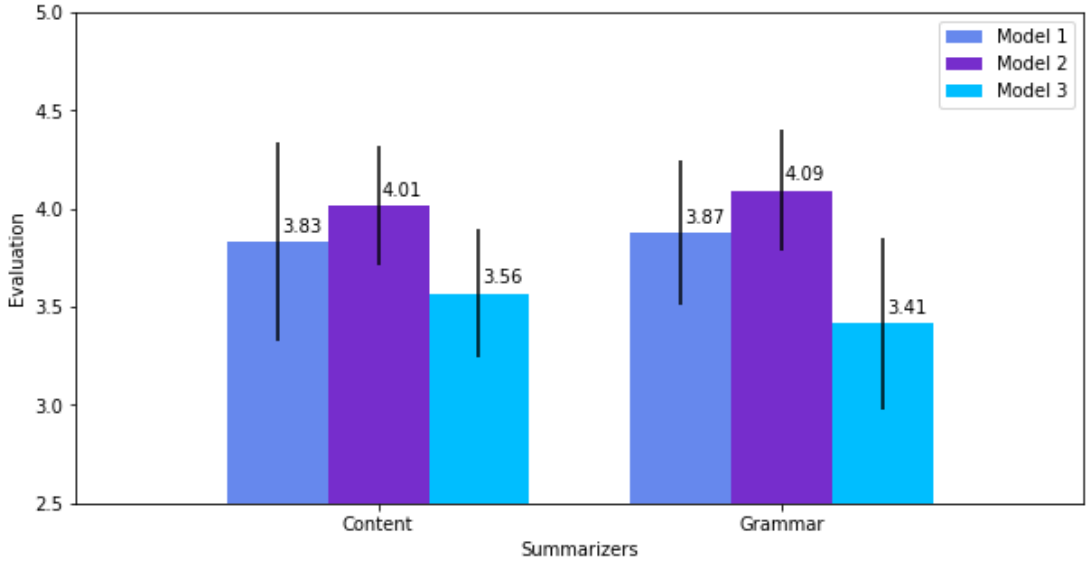

Figure 3. Evaluation of graduates regarding the quality of the generated texts. On the left, the content evaluation, and the right, the grammar evaluation.

Initially, according to Figure 3, it is noted that the assessments of Model 1 and Model 2 decreased compared to the grades given by students in computing courses. In contrast, the grade, both for content and grammar, of Model 3 grew little, although it remained, according to the graduates, with an inferior performance compared to the other summarizers.

The same pattern of Figure 2 was maintained in the evaluation of Figure 3, the summarizer of Model 2 was the best evaluated and, unlike the previous one, it had lower values of variance, indicating an agreement of the evaluators regarding content and grammar. Thus, it remained with an average close to 4. Unlike the observations made by the students, those given by the graduates had more praise regarding the content quality of the text and grammar. Table 3 presents some of the justifications for the assessments given by computer graduates and students.

Tables 4 and 5 present the vote of each graduate and computer professor of what would be the best and worst content summarizer, according to their personal opinions. It is easy to check again in this Table the evaluators' preference regarding Model 2, remembering that the texts were randomized and without a description of which was the summarizer.

Interestingly, again Model 2 was preferred as the best summarizer. However, for teachers and graduates, there was a tie and a preference for Model 1 in texts 1 and 2.

It was expected that Model 2 was superior to Model 1, given that Model 1 uses the

Table 3. Justification of the candidate's evaluations on each summarizer.

	Students	Graduates and teachers
Model 1	Minor mismatches and the use of terms not commonly used such as "Idiom Languages"	Very good grammatical quality
	Duplicates are unnecessary.	For someone who is not a student in the field, the text is confused and has pieces of ideas that do not communicate.
Model 2	Incorrect use of the mid-sentence period: "analysis, requirements analysis. Specification(...)"	In grammatical terms the text is very good
	The first paragraph restricts a definition that could be more comprehensive and complete.	The first part of the text is unclear and doesn't seem to make much sense
Model 3	It is not as clear as the previous texts, but it informs the necessary	Presence of expressions without syntax (). Refers to a string "by this string [...]" in the first paragraph, but does not specify which.
	<N>is confusing	The text demonstrates clarity in the transfer of information

Table 4. Voting for the best text by graduates and teachers.

	Text 1	Text 2	Text 3	Text 4	Text 5	Total
Model 1	**6**	**7**	4	2	4	23
Model 2	**6**	6	**7**	**9**	**8**	**36**
Model 3	1	0	2	2	1	6

Table 5. Voting for the worst text by graduates and teachers.

	Text 1	Text 2	Text 3	Text 4	Text 5	Total
Model 1	3	2	2	4	5	16
Model 2	4	1	4	1	1	11
Model 3	**6**	**10**	**7**	**8**	**7**	**38**

DistilBART. This distilled version is lighter and with fewer parameters than the BART model used in Model 2. Thus, because Model 2's learning network is more robust, it can generate more succinct texts, although it requires more processing time. Model 3 uses PEGASUS, which manages to obtain good performances even if trained with few examples. However, it is also not as robust as the BART model, which may be one reason. Model 3 was inferior in the assessment on all form requirements.

Although the texts needed to go through two stages of automatic translation, the summarizers were able to disseminate the educational content satisfactorily in some cases. Thus, this work begins to follow a path towards the generation of more sophisticated learning objects through a structuring by topics or even working with post-processing in the output of these summaries.

5. Conclusion

This work presents an alternative in the generation of learning objects through the use of summarizers to help students and teachers in the teaching process in the digital environment. An algorithm was developed in combination with knowledge bases, translators, and summarizers based on Deep Learning models to generate text summaries.

To evaluate the result, we experiment with experts through a form where they had to assess the quality of content and grammar of the generated learning object. Computer students and graduates were selected to evaluate and compare the quality of texts generated by three summarizer models.

Two models stood out in terms of content and grammar evaluation, obtaining good results with an average of around 4 on a scale of 1 to 5. Much is since both use one of the most robust summarization models, especially Model 2, which uses the BART model, which expected better performance than Model 1 with its DistilBART architecture. It is interesting to note that even with the translation process into another language, the models still stood out in the field of education. Model 3 was the one with the lowest evaluation and showed a need for post-processing to remove possible errors that hinder the readability of the content.

Thus, it is possible to note the feasibility of using summarizers for the generation of text-based learning objects, contributing to the demonstration of the feasibility of using summarizers from English to Portuguese through automatic translators.

As future work, further tests with experts from other domains of knowledge are still needed. Comparative blind tests can also be performed on summarized texts between knowledge professionals and summarizers. Another possible future work is related to the generation of multimedia learning objtects in textual hypermedia authoring languages like the model seen in [Lima et al. 2010].

6. Acknowledgment

The authors are grateful for the financial support of FAPEMA (*Fundação de Amparo à Pesquisa e ao Desenvolvimento Científico e Tecnológico do Maranhão*).

References

Busson, A. J. G., Damasceno, A. L. d. B., Azevedo, R. G. d. A., Neto, C. d. S. S., Lima, T. d. S., and Colcher, S. (2017). A hypervideo model for learning objects. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media*, HT '17, page 245–253, New York, NY, USA. Association for Computing Machinery.

Damasceno, A. L. B., Busson, A. J. G., Lima, T. S., and Neto, C. S. S. (2020). Authoring hypervideos learning objects. *Special Topics in Multimedia, IoT and Web Technologies*, page 149.

Hooshyar, D., Yousefi, M., Wang, M., and Lim, H. (2018). A data-driven procedural-content-generation approach for educational games. *Journal of Computer Assisted Learning*, 34(6):731–739.

Kurdi, G., Leo, J., Parsia, B., Sattler, U., and Al-Emari, S. (2020). A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30(1):121–204.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Li, C. and Xing, W. (2021). Natural language generation using deep learning to support mooc learners. *International Journal of Artificial Intelligence in Education*, pages 1–29.

Lima, G., Soares, L. F. G., Neto, C. d. S. S., Moreno, M. F., Costa, R. R., and Moreno, M. F. (2010). Towards the ncl raw profile.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rocha, O. R., Zucker, C. F., Giboin, A., and Lagarrigue, A. (2020). Automatic generation of questions from dbpedia. *International Journal of Continuing Engineering Education and Life Long Learning*, 30(3):276–294.

Rüdian, S., Heuts, A., and Pinkwart, N. (2020). Educational text summarizer: Which sentences are worth asking for? *DELFI 2020–Die 18. Fachtagung Bildungstechnologien der Gesellschaft für Informatik eV*.

Xu, Y., Smeets, R., and Bidarra, R. (2021). Procedural generation of problems for elementary math education. *International Journal of Serious Games*, 8(2):49–66.

Yang, G., Chen, N.-S., Sutinen, E., Anderson, T., Wen, D., et al. (2013). The effectiveness of automatic text summarization in mobile learning contexts. *Computers & Education*, 68:233–243.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. (2020). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.