A Recommendation System Framework for Educational Content Reinforcement in Virtual Learning Environments

Adson R. P. Damasceno, Lucas C. Carneiro, João Victor F. T. De Sampaio, Allberson B. O. Dantas, Eudenia Magalhães, Paulo Henrique M. Maia and Francisco C. M. B. Oliveira
UECE - State University of Ceará, Fortaleza, Brazil

Keywords: Recommendation of Educational Content, Educational Content Reinforcement, Recommendation Systems, Virtual Learning Environments.

Abstract: In Virtual Learning Environments, tutors play a vital role by supporting students and improving their learning through the courses. One important task is to identify content with which the students struggle and give them suggestions for educational resources to reinforce their learning and overcome difficulties. However, providing individualized suggestions for each student may be infeasible, especially for courses with many enrolled students. In this work, we propose and validate a framework for building recommendation systems of educational content for Virtual Learning Environments. Our proposed system identifies the content that a student needs to reinforce based on the results of his assessments and recommends resources that best relate to the questions that he answered incorrectly, using Information Retrieval, Machine Learning, and Natural Language Processing techniques. We validate our proposed solution by taking as a case study data collected from DAL - Dell Accessible Learning, an distance learning platform. We built a dataset with content from 8 courses to compare the performance of different methods and text representations in our framework. Our best result achieved an accuracy of 0.89 using a Nearest Neighbor method with TF-IDF representation.

1 INTRODUCTION

During the learning process, a critical learning opportunity is created when the student is asked in assessments to use the concepts presented in a context different from the one they were introduced to. It is by working with new ideas that the apprentice appropriates them. This is also where his incompetence surfaces (McLeod, 2010). At this point, the instructor's intervention is precious, providing the necessary information for the practical completion of the appropriation process. However, in online courses mediated through Virtual Learning Environments (VLE), the sheer volume of online activities can be too much for the tutor, and the workload on online tutors are often reported to be significantly greater than what it is in a face-to-face teaching context (Bernath and Rubin, 2001). It is challenging and time-consuming for tutors to keep track of all students' learning activities to provide individualized recommendations and feedback, especially when the number of enrolled students in a course grows. Some VLE may not even have tutors available to perform such activity. Therefore, a VLE should make available the greatest number of resources that favor the students' initiative, since the main premises of distance learning are the optimization of time, flexibility, and autonomy in the knowledge retention process.

One solution to mitigate this problem is the use of Machine Learning (ML) to build recommendation systems of educational content. Such systems aim to provide valuable educational resource suggestions to students to reinforce their knowledge in contents that they lack appropriation. Suggestions are given based on data patterns collected from the users and the educational resources, learned by ML algorithms. Recommendation systems of educational content allow the reduction of workload on online tutors and increase the autonomy of students during their learning process in VLE.

In light of the provided scenario, we offer a framework for developing recommendation systems for educational content reinforcement in VLE. Our framework consists of gathering textual data from the courses of a VLE to build a knowledge base, and employ Information Retrieval (IR), ML and Natural Language Processing (NLP) techniques to suggest educational content. The suggestions are related

Damasceno, A., Carneiro, L., T. De Sampaio, J., Dantas, A., Magalhães, E., Maia, P. and Oliveira, F.
A Recommendation System Framework for Educational Content Reinforcement in Virtual Learning Environments.
DOI: 10.5220/0011032000003182
In *Proceedings of the 14th International Conference on Computer Supported Education (CSEDU 2022) - Volume 1*, pages 228-235
ISBN: 978-989-758-562-3; ISSN: 2184-5026

to questions incorrectly answered by students during pedagogical exams, thus reinforcing the content with which they struggled the most. Our approach uses only textual data from pedagogical resources and incorrectly answered questions and can be implemented in any VLE, provided that texts of the education contents can be gathered. To validate the viability and performance of the proposed framework, we considered a case study where we evaluated the recommendation methods on data gathered from Dell Accessible Learning (DAL) platform, an accessible distance learning environment. The obtained results show the viability of creating and implementing recommendation systems in VLEs using our framework. Furthermore, we believe that, in addition to reducing the effort of online tutors, this application will allow students to gain more autonomy in the learning process.

The remainder of this paper is organized as follows: Section 2 provides an overview of related works in the field of recommendation systems in educational contexts. Next, our proposed methodologies and their evaluation are described in Section 3. The results and inferences drawn from them are presented in Section 4. Finally, the conclusions we can make from our investigation are presented in Section 5.

2 RELATED WORK

Computer systems technologies have been widely used in the construction of educational software, from training systems, educational games to virtual learning environments capable of intelligently recommending educational content to students. A recommendation system of educational content is a functionality that provides students in VLE with pedagogical resources that are likely to be helpful to their learning process (Shani and Gunawardana, 2011). There are numerous research initiatives in educational resource recommendation systems (Marante et al., 2020; Rivera et al., 2018; Kulkarni et al., 2020; Klašnja-Milićević et al., 2015). In the following paragraphs, we present some promising directions.

In recent years, recommendation systems for educational content based on finding patterns of similarity among students have been widely investigated (Zaliane, 2006; Urdaneta-Ponte et al., 2021; Nurjanah, 2016). These patterns can occur in several ways, including performance measured in corresponding activities, patterns of access to the system, and socioeconomic issues. Such systems can use a variety of learning techniques, including data grouping, association rules, natural language processing, and collaborative filtering (Sicilia et al., 2010; Indrayadi and Nurjanah, 2015).

A relevant research question in recommendation systems for educational content is how to identify student learning patterns (Klašnja-Milićević et al., 2011; Truong, 2016; Yan et al., 2021). In fact, the study of individual student learning styles is a well-established research field, with several learning profiles identified, each proposing different descriptions and classifications of learning types (Coffield et al., 2004). Increasingly adopted methods consist in inferring students' learning patterns based on their interactions with the system, such as their access to pages and online content, the number of attempts they make in assessments, among others (Kelly and Tangney, 2004; Gope and Jain, 2017; Nafea et al., 2019; Jyothi et al., 2012).

In this work, the only data gathered from the students used as input are the texts of incorrectly answered questions in courses' assessments. Our framework uses NLP techniques to predict which content in a knowledge base is best related to these questions and recommend it to students in order to reinforce their learning process. We did not find in literature works that recommend content based purely on the text from questions that students answered incorrectly. In the next section, the details of our recommendation system are described thoroughly.

3 METHODOLOGY

This section comprises the proposed solution, as well as the experimental setup to validate our approach considering the specific case of DAL, an accessible distance learning environment focused mainly on technology courses.

Our approach consists of creating a knowledge base, or corpus, of educational content and recommending the most similar documents using the texts of questions answered incorrectly during evaluation activities as queries. We emphasize that our strategy applies to other educational platforms as long as it is possible to collect a corpus of educational content.

3.1 Corpus Construction and Analysis

We created our corpus with documents of educational content from the courses offered on the DAL platform. This platform provides a variety of online courses in various disciplines, including courses in the areas of data science, web development, mobile development, management, languages, *inter alia*. The lessons are displayed as videos or texts, the latter being split into topics. The original language of most of

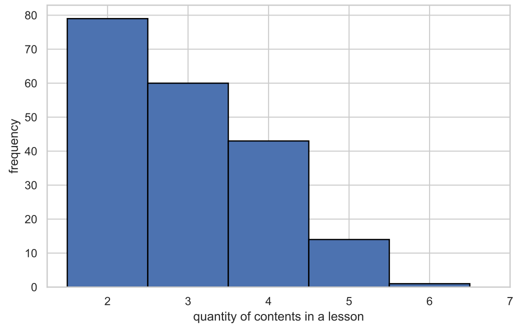

Figure 1: Frequency of the quantity of educational contents in lessons.

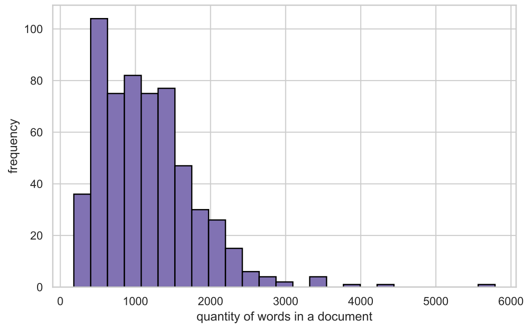

Figure 2: Frequency of the quantity of words in documents.

the courses is Brazilian Portuguese.

We developed a script that automatically downloads the files from the DAL platform and builds the corpus. The textual content from web page lessons is collected by parsing HTML files, while video content is collected by parsing its subtitles. The corpus is stored as a table in which each row represents a document. We describe the table's columns in Table 1.

Table 1: Description of variables in the corpus in table format.

Variable	Description
Course	Name of the course from where the content was collected. E.g.: Data Visualization.
Lesson ID	Number that uniquely identifies the lesson from where the content was collected. It will be used to filter out the possible recommendations. E.g.: 102192363916253.
Content name	The title of the content. This information is what will be recommended to the student. It can be a topic from a web lesson or the number of a video lesson. E.g.: Topic 1 - Correlation and scatter plots.
Text	Text of the document. This will be used to compare with a text of a question to find the best content to recommend. E.g. Understand the concept of correlation between two variables; Learn to create two-variable scatter plots using seaborn; Learn how to create scatter plots with a regression line using seaborn...

Each lesson has between 2 and 6 educational contents that can be recommended. Figure 1 illustrates the frequency of quantity of contents in lessons. Most lessons have less than four contents, with two contents in a class being the most frequent. The higher the quantity of contents in one class, the more a recommendation system is helpful for a student, since there is a higher space of search to reinforce the contents, and the system can direct its efforts to achieve better results.

We also analyse the distribution of the quantity of words in the documents, illustrated in Figure 2. In general, the documents are long texts. The smaller document have 180 words, and the longer have near 6000 words. Half of the documents have more than 1000 words.

3.2 Recommendation Methods

As discussed previously, we aim to recommend educational content related to incorrectly answered questions during assessment activities. In this sense, the challenge relies in identify which content is more relevant for a given question and lesson. We highlight that, in this scenario, every question is associated with a unique lesson that is known beforehand. To achieve this goal we took two approaches well know in literature: one based on finding the document with the highest cosine similarity in a vector representation, and the other based on supervised learning, using the Naive-Bayes method for text classification. These methods were chosen due its simplicity and good performance in scenarios where there is not a large amount of data available. Both approaches are discussed in detail below.

3.2.1 Nearest Neighbor

The Nearest Neighbor (NN) approach is a typical IR method, which is related with content-based recommendation systems (Pazzani and Billsus, 2007). This method consists of, for a given question and lesson, find the most similar document in the subset of documents related to that lesson in our corpus, and then recommend the name of the related content. In the context of artificial intelligence applied in education, this approach was previously used by (Montesuma et al., 2021) for retrieving documents that answer students' conceptual questions. We formalize ours as follows.

Let $q \in \mathbb{R}^M$ refers to a question in a vector representation, related to a lesson ℓ, and $D = \{d_1, d_2, ..., d_N\}$ refers to a corpus formed by N documents gathered from lessons of online courses. Each document d_i is defined as a tuple $d_i = \{v_i, \ell_i, c_i\}$, corresponding to the document's vector representation $v_i \in \mathbb{R}^M$, the associated lesson id of the document ℓ_i, and the name of the content c_i, respectively. Each lesson ℓ is associated to a subset $S = \{d_{k_1}, d_{k_2}, \cdots, d_{k_P}\} \subset D$ of P documents. For a given input question q from a lesson ℓ and a corpus D, the content name $c_{i^\star} \in S$ that best relates to q is given by $\operatorname{argmax} \operatorname{sim}(q, v_i)$ where the function $\operatorname{sim}(u, v)$ represents the cosine similarity between two vectors.

Any vector representation of text can be used in this approach. In our experiments, we used two well-known vector representations for documents: Term Frequency-Inverse Document Frequency (TF-IDF) (Jones, 1972) and centroids of Word Embeddings (Galke et al., 2017), more specifically the Word2Vec embeddings (Mikolov et al., 2013). TF-IDF representation models the importance of each word in a document so that documents that have a higher intersection of keywords will have greater similarity. Word2Vec represents each word in a dense vector space, where semantically similar words are spatially close. The centroid of a text is calculated as the mean vector of all words in that text. Thus, it is intended that texts with similar semantics have greater similarity in this vector space, which is desired in many scenarios. Figure 3 illustrates the NN approach. An advantage of the NN approach is that it does not requires model training, since it is a instance-based learning method. Updating the model to make recommendations for new lessons only requires updating the corpus.

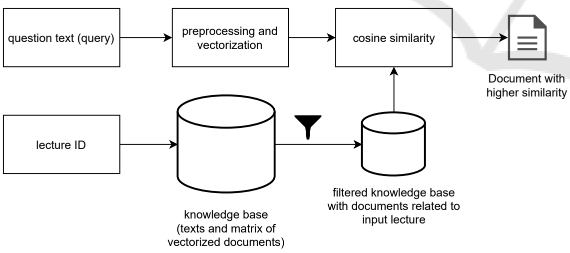

Figure 3: Diagram of the NN approach.

3.2.2 Naive Bayes

The Naive Bayes (NB) is a well-known supervised learning ML method used for classification, based on the Bayes Rule. In this strategy, the problem is modeled as a text classification task, where from a given question and a lesson, we aim to predict the name of the content related to that question. The NB classifier assumes that the features of a data point are independent. More specifically, we use the Multinomial Naive Bayes classifier (Kibriya et al., 2004), which assumes that the likelihood of TF-IDF features follows a multinomial distribution. Although the multinomial distribution presumes that variables are discrete counts, in practice, TF-IDF fractional counts are known to perform better.

Since we known beforehand that a given question is related to a specific lesson, we also know that the prediction should be one of the contents related to that lesson. Therefore, instead of training one model with all the corpus D, we train one independent model for each lesson. In this way, a lesson ℓ is associated to a subset $S_\ell = \{d_{k_1}, d_{k_2}, \cdots, d_{k_P}\} \subset D$ and with a model m_ℓ. Each model m_ℓ is trained with the documents $d_{k_i} \in S$ and can predict only the contents $c_{k_i} \in d_{k_i}$. As so, before predict a content for a new question q, one must first choose the correct model trained with data from the lesson related to q.

Since the target classes correspond to the content name c_i of each document, we only have one example for each class. This implies that the prior probability for each class is uniform, so only the likelihood term from the Bayes rule is used to determine the prediction. We highlight that any other supervised model for classification could be used, following the strategy of training one model for each lesson, but we choose NB for its relative low complexity, low computational cost to train and fair performance in many text classification problems, in particular when there are a low amount of examples available for each class. Figure 4 illustrates the NB approach.

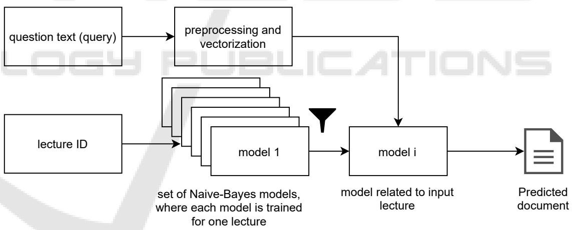

Figure 4: Diagram of the NB approach.

3.3 Validation Data Set

To validate and compare the performance of our approaches, we manually built a data set with a sample of actual questions from DAL platform courses. We collected multiple-choice questions from assessments in each class in the 8 most popular courses in DAL. The labels, i.e., the contents names, were then assigned by tutors with expertise in the fields by checking in the lesson the content each question refers.

Figure 5 illustrates the courses and the number of samples collected. The data set has 122 questions from 8 courses. Each question is related to only one lesson, which is known in advance. Some courses have a larger number of samples than others, due to the availability of tutors with the expertise to label the data.

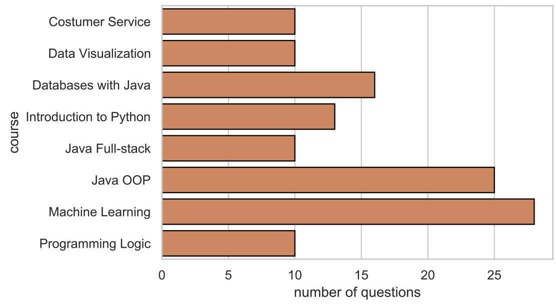

Figure 5: Quantity of questions by course in the validation dataset.

Besides the text of the question itself and the associated content name, it was also collected the course name, lesson id, multiple-choice items, and correct answer. Table 2 describes each variable in this data set.

We use this data set by taking the text and the lesson ID from each question as input to our methods and comparing the predicted content name with the original content name manually assigned to the question. We also aim to investigate how the addition of extra information as the multiple-choice items or the correct item impacts the performance of our models. Therefore, we compare the predictions using the text of the question alone, text and multiple-choice items, and text and only the correct answer. We believe that the addition of the correct answer may provide extra information that may help our models improve performance, due to the existence of keywords in many of the answers.

For comparing between methods, we use as a metric the recall for each course, i.e., the proportion of correct predictions in each course. For further comparison, we use the balanced accuracy, which corresponds simply to the average recall of each course.

Table 2: Description of columns in the validation data set.

Variable	Description
Course	Name of the course.
Lesson	Numeric ID that uniquely identifies the lesson from which the question was collected.
Text	Text of the question.
Multiple-choice items	Items with possible answers to the question.
Correct answer	The item that correctly responds the question.
Content	The target that we want to predict. The name of the content related to the question.

4 RESULTS AND DISCUSSION

In this section, we describe the experimental setup and results of the proposed experiments. The main goal of these experiments was to validate the viability of our framework, but also to compare the performance of each proposed method in the particular scenario described in our validation data set. We also highlight that after the experiments, we implemented a recommendation system in DAL platform, which send recommendations to student thorough a chatbot described in (Damasceno et al., 2020). The integration of the system with the chatbot occurs by using a message broker.

4.1 Experimental Setup

The only preprocessing applied to the texts is converting them to lowercase. This choice is due to the high variability of expressions and terms present in technology courses, such as words in other languages and programming language keywords and symbols. We stated empirically that more sophisticated preprocessing steps, such as lemmatization, resulted in a worse general result. As pointed out previously, we use TF-IDF and Word2Vec centroids to represent text in a vector space. The TF-IDF representation considers unigrams and bigrams as tokens, separated by spaces or punctuation, and the resulting vector space has a dimension of 227,917. The Word2Vec model was trained using all documents from our corpus, with a vector dimension of 100.

4.2 Results

We first evaluate the effect of adding multiple-choice items and the correct answer to the text input for our methods. Table 3 describes the balanced accuracy for each method, when different inputs are taken into account. We can observe that for each variation of inputs, the NN approach with Word2Vec features performed the worst. The NN and NB approaches using TF-IDF features have similar performances for every input, being somewhat equivalent.

This indicates that using the centroid of word vectors is not a good representation for this task. This may be due to the loss of information that occurs when representing a text composed of many word vectors as an average vector. The information loss is more significant for longer texts, which is the case for most of the documents in the corpus. In fact, Word2Vec focuses on modeling the semantic similarity of words, not their importance in a document, unlike TF-IDF. The experiments revealed the latter to be more suitable for the task of finding a document related to a question by measuring the cosine similarity between them.

In terms of which input works best for the task, both adding multiple-choice items and adding correct answers improved accuracy when compared to using only the text of the question. However, adding all multiple-choice items had a slightly better result than adding only the right answer. Despite the correct answer being a more precise information, adding all items, including the incorrect ones, was revealed to be a better choice, since words in wrong answers may still be present in the target documents because they can be related to the same topics.

Table 3: Balanced accuracy for different inputs. The best results were obtained with the methods NN and NB in TF-IDF representation with questions and all of the multiple-choice items. Word2Vec features did not perform well for all the inputs.

	NN TF-IDF	NN Word2Vec	NB TF-IDF
Question only	0.82	0.63	0.79
Question and multiple-choice items	**0.89**	0.60	0.86
Question and correct answer	0.87	0.63	0.84

We further investigate the best input option by analyzing the performance in each course for the different methods. These results are presented in Table 4.

Table 4: Recall by course for each method, using as input the question and all the multiple-choice items.

Course	NN - TF-IDF	NN Word2Vec	NB - TF-IDF
Customer Service	1.00	0.30	1.00
Databases with Java	0.81	0.81	0.81
Java OOP	0.76	0.56	0.76
Introduction to Python	0.85	0.62	0.69
Programming Logic	1.00	0.70	1.00
Java Full Stack	0.90	0.70	0.80
Machine Learning	0.79	0.64	0.79
Data Visualization	1.00	0.40	1.00
Balanced Accuracy	**0.89**	0.59	0.86

As expected, NN with Word2Vec features performed the worst for all courses, with the notable exception of the course Database Fundamentals in Java, for which all methods had the same recall. Both NN and NB with TF-IDF features performed similarly in each course. In most courses, they achieved the same recall, possibly due to the relatively low quantity of samples. In fact, those methods are somewhat similar, as they take into consideration the importance of words in documents, weighted by TF-IDF.

NN method performed better than NB in courses like Introduction to Python and Java Full Stack, therefore having a slightly better balanced accuracy. In practical terms, the NN method is also less expensive to maintain and scale, since it does not require the training of new models when new courses are created, only that the corpus be updated. On the other hand, because one NB model is trained for each class in a course independently, it would require new models to be trained every time a course is updated or a new course is added to the corpus.

Our best result, obtained with NN and TF-IDF features achieved a balanced accuracy of 0.89, which is considered a satisfactory result for our pedagogical purposes. We also believe that even when the desired educational content is not recommended, the recommended document still have a significant relation with the question, since both come from the same lesson. Thus, even in case of errors, the target and the predicted documents have closely related topics, which would still be beneficial to the students. To confirm this statement, we next analyze some examples of incorrect recommendations.

4.3 Error Analysis

Out of 122 questions, our best method predicted incorrectly 18 contents. To recognize the limitations of this method, we performed a qualitative analysis, comparing the text of the incorrect predictions with the text of the desired ones. More specifically, we analyzed the errors in the Machine Learning course, which had 6 incorrect predictions out of a total of 28 questions. Table 5 shows this comparison. The first column in the table is the name of the content predicted by the model and the second one (Sim Q/P) is the cosine similarity between the text of the question and the text of the predicted content in TF-IDF representation. The third column is the name of the target content and the fourth one is the cosine similarity between the text of the question and the text of the target content (Sim Q/T).

Table 5: Comparison between predicted and target contents in the incorrect predictions from Machine Learning course. Most of predicted contents are introductory topics that precedes the targets in the lesson.

Predicted content	Sim Q/P	Correct content	Sim Q/T
Introduction to data analysis with Python	0.033	Statistical analysis of a data set	0.025
Deleting columns	0.074	Allocating missing values	0.056
The artificial neuron	0.049	The "exclusive-or" example	0.027
Introduction to Tensorflow	0.049	Construction and evaluation of the Tensorflow model	0.043
Introduction to SVM	0.102	Implementing SVM with Kernels	0.076
Confusion, Recall and Confusion Matrix	0.049	Introduction to Model Evaluation	0.031

From Table 5, we notice that most of the errors in the Machine Learning course occurred by recommending an introduction content instead of the target, a most advanced content. That is, the recommended contents usually precede the targets. In this course, this occurred 4 out of 6 times.

A possible reason behind that is that documents of introductory contents in this corpus are usually longer, having more words. They also usually introduce some of the concepts that will be studied in the next content, thus having a lot of words in common. So it is likely that a question has words in common with introductory content. Although the recommendation of this introductory content is not misleading to the student, it lacks the precision that is expected from a recommendation system and may not be helpful enough to improve the student learning process. This gap shows room for improvement by exploring other methods that don't solely rely on the frequency of words in texts.

5 CONCLUSIONS

In this work, we proposed an approach to build a recommendation system of educational content in VLE. The approach consists of create a corpus of documents from online courses and recommend to a student the contents best related to the incorrectly answered questions, aiming to reinforce the students learning process. We explored two methods to find the best document for a given question and lesson, the Nearest Neighbor method, based in cosine similarity from documents represented in a vector space (TF-IDF or Word2Vec), and the Naive Bayes method.

We evaluated this approach in a study case using data collected from DAL platform. Our best result achieved a balanced accuracy of 0.89 using NN with TF-IDF, which was considered satisfactory but still shows room for improvement. In general, the results showed the feasibility of implementing our framework, which can directly impact the students' learning process by improving their autonomy. A practical consequence was the implementation of a recommendation system in DAL platform with the settings that achieved better results in these experiments.

As future work, we intend to evaluate the impact of this application in the learning process of students in DAL platform, collecting quantitative and qualitative data of the experience and performance of the students while taking lessons. We also want to experiment with more sophisticated and state of the art methods or different similarity metrics or text representations to achieve better accuracy in recommendations, e.g. attention-based models as BERT (Vaswani et al., 2017; Akkalyoncu Yilmaz et al., 2019), pre-trained deep learning models, and transfer learning (Do and Ng, 2005; Yan and Zhang, 2009; Deb, 2019). Finally, we intend to validate our framework on a larger dataset, with data with greater variation, collected from courses in different areas.

REFERENCES

Akkalyoncu Yilmaz, Z., Wang, S., Yang, W., Zhang, H., and Lin, J. (2019). Applying BERT to document retrieval with birch. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 19–24, Hong Kong, China. Association for Computational Linguistics.

Bernath, U. and Rubin, E. (2001). Professional development in distance education – a successful experiment and future directions. *Innovations in Open & Distance Learning, Successful Development of Online and Web-Based Learning*, pages 213–223.

Coffield, F., Moseley, D., Hall, E., and Ecclestone, K. (2004). Should we be using learning styles? what research has to say to practice.

Damasceno, A. R., Martins, A. R., Chagas, M. L., Barros, E. M., Maia, P. H. M., and Oliveira, F. C. (2020). Stuart: an intelligent tutoring system for increasing scalability of distance education courses. In *Proceedings of the 19th Brazilian Symposium on Human Factors in Computing Systems*, pages 1–10.

Deb, S. (2019). Driving content recommendations by building a knowledge base using weak supervision and transfer learning. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 531–531.

Do, C. B. and Ng, A. Y. (2005). Transfer learning for text classification. *Advances in neural information processing systems*, 18:299–306.

Galke, L., Saleh, A., and Scherp, A. (2017). Word embeddings for practical information retrieval. *INFORMATIK 2017*.

Gope, J. and Jain, S. K. (2017). A learning styles based recommender system prototype for edx courses. In *2017 International Conference on Smart Technologies for Smart Nation (SmartTechCon)*, pages 414–419. IEEE.

Indrayadi, F. and Nurjanah, D. (2015). Combining learner's preference and similar peers' experience in adaptive learning. In *CSEDU (1)*, pages 486–493.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

Jyothi, N., Bhan, K., Mothukuri, U., Jain, S., and Jain, D. (2012). A recommender system assisting instructor in building learning path for personalized learning system. In *2012 IEEE Fourth International Conference on Technology for Education*, pages 228–230. IEEE.

Kelly, D. and Tangney, B. (2004). Predicting learning characteristics in a multiple intelligence based tutoring system. In *International Conference on Intelligent Tutoring Systems*, pages 678–688. Springer.

Kibriya, A. M., Frank, E., Pfahringer, B., and Holmes, G. (2004). Multinomial naive bayes for text categorization revisited. In *Australasian Joint Conference on Artificial Intelligence*, pages 488–499. Springer.

Klašnja-Milićević, A., Ivanović, M., and Nanopoulos, A. (2015). Recommender systems in e-learning environments: a survey of the state-of-the-art and possible extensions. *Artificial Intelligence Review*, 44(4):571–604.

Klašnja-Milićević, A., Vesin, B., Ivanović, M., and Budimac, Z. (2011). E-learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & education*, 56(3):885–899.

Kulkarni, P. V., Rai, S., and Kale, R. (2020). Recommender system in elearning: A survey. In Bhalla, S., Kwan, P., Bedekar, M., Phalnikar, R., and Sirsikar, S., editors, *Proceeding of International Conference on Computational Science and Applications*, pages 119–126, Singapore. Springer Singapore.

Marante, Y., Silva, V., Jr., J. G., Vitor, M., Martins, A., and Souza, J. D. (2020). Evaluating educational recommendation systems: a systematic mapping. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pages 912–921, Porto Alegre, RS, Brasil. SBC.

McLeod, S. (2010). Zone of proximal development. *Simply psychology*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.

Montesuma, E. F., Carneiro, L. C., Damasceno, A. R., de Sampaio, J. V. F., Férrer Filho, R. F., Maia, P. H. M., and Oliveira, F. C. (2021). An empirical study of information retrieval and machine reading comprehension algorithms for an online education platform. In *Anais do XIII Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pages 217–226. SBC.

Nafea, S. M., Siewe, F., and He, Y. (2019). On recommendation of learning objects using felder-silverman learning style model. *IEEE Access*, 7:163034–163048.

Nurjanah, D. (2016). Good and similar learners' recommendation in adaptive learning systems. In *CSEDU (1)*, pages 434–440.

Pazzani, M. J. and Billsus, D. (2007). Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer.

Rivera, A. C., Tapia-Leon, M., and Lujan-Mora, S. (2018). Recommendation systems in education: A systematic mapping study. In Rocha, Á. and Guarda, T., editors, *Proceedings of the International Conference on Information Technology & Systems (ICITS 2018)*, pages 937–947, Cham. Springer International Publishing.

Shani, G. and Gunawardana, A. (2011). Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer.

Sicilia, M.-Á., García-Barriocanal, E., Sánchez-Alonso, S., and Cechinel, C. (2010). Exploring user-based recommender results in large learning object repositories: the case of merlot. *Procedia Computer Science*, 1(2):2859–2864.

Truong, H. M. (2016). Integrating learning styles and adaptive e-learning system: Current developments, problems and opportunities. *Computers in human behavior*, 55:1185–1193.

Urdaneta-Ponte, M. C., Mendez-Zorrilla, A., and Oleagordia-Ruiz, I. (2021). Recommendation systems for education: Systematic review. *Electronics*, 10(14):1611.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yan, L., Yin, C., Chen, H., Rong, W., Xiong, Z., and David, B. (2021). Learning resource recommendation in e-learning systems based on online learning style. In *International Conference on Knowledge Science, Engineering and Management*, pages 373–385. Springer.

Yan, R. and Zhang, J. (2009). Transfer learning using task-level features with application to information retrieval. In *Twenty-First International Joint Conference on Artificial Intelligence*.

Zaliane, O. (2006). Recommender systems for e-learning: towards non-intrusive web mining. *Data Mining in E-Learning(Advances in Management Information)*, 4:79–93.