Adaptive Learning Content Recommendation using a Probabilistic Cluster Algorithm

Adson Marques Esteves, Aluizio Haendchen Filho, André Luiz Alice Raabe,
Angélica Karize Viecelli, Jeferson Miguel Thalheimer and Lucas Debatin
Laboratory of Technological Innovation in Education (LITE), University of the Itajai Valley (UNIVALI), Itajai, Brazil

Keywords: Adaptive Computer Learning, Educational Technology, Learning Technology, Recommendation System.

Abstract: Nowadays there are many research using the LDA (Latent Dirichlet Allocation) algorithm to find preferences and characteristics for recommendation systems. In some of the most relevant studies, the recommendation is based on the student's level of evolution within the discipline. This work presents a new recommendation approach with the LDA algorithm. The approach differs from previous LDA studies since the recommendation technique is based on the experiences and preferences from a group of students and not just an individual student. The main objective is to verify, through simulation, whether the methods used, and the algorithm can generate recommendations close to those considered ideal. The obtained results indicate that the application of the LDA for creating groups to generate recommendations provides a good result in delivering content and practices in accordance with the student's interests. It's empirical research, as the conclusions are drawn from concrete and verifiable evidence used in the simulations.

1 INTRODUCTION

The high dropout rate in computer science and related courses at universities is a negative phenomenon that occurs all over the world. In Brazil, the competition for admission to technology courses is quite high. Among the technology courses, science, mathematics, and computing courses were the three most sought after disciplines in 2017 (INEP, 2018). The offer of technological undergraduate courses in specific branches of information technology has expanded considerably, and there is growing demand for both on-site and distance learning courses.

Despite this high demand, the dropout rates are quite significant, ranging from 22% to 32%, one of the highest among undergraduate courses in Brazilian universities (Lobo, 2017). One of the reasons for this are the difficulties in the early discipline of computing course: Algorithms and Programming (Hoed, 2016).

One of the resources frequently used to reduce the difficulty associated with these courses are the Intelligent Tutoring Systems (ITS). These systems consider the student's peculiarities during the process of conducting the learning paths.

More recently, ITS have been called Adaptive Systems, because they adapt to the knowledge of the student who is learning. The strategies used in these systems generally include response adaptation, tips, recommendations, navigation along the learning path, and adaptive referential material. The content recommendation algorithm is one of the most important processes in the adaptive system and is one of the system's main focuses of intelligence. Different types of techniques can be applied in this process, such as decision models, reasoning rules, ontology, clustering, etc.

A systematic review of related literature shows that many researchers are using the LDA (Latent Dirichlet Allocation) algorithm to group students based on their preferences and characteristics (Apaza, et al. 2014; Erkens, Bodemer, & Hoppe, 2016; Lin, He, & Deng, 2021). In some of the most relevant studies, the recommendation is based on the student's level of evolution within the discipline. Attributes are extracted from texts written by the students on subjects such as their hobbies and interests.

The LDA recommendation technique is used in our approach, but with a different strategy. Instead of using it to identify interests and hobbies and use them to form a recommendation, students are grouped according to their previously acquired preferences

Esteves, A., Filho, A., Raabe, A., Viecelli, A., Thalheimer, J. and Debatin, L.
Adaptive Learning Content Recommendation using a Probabilistic Cluster Algorithm.
DOI: 10.5220/0011056200003179
In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS 2022) - Volume 1*, pages 724-731
ISBN: 978-989-758-569-2; ISSN: 2184-4992
Copyright © 2022 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

and profile characteristics. From this grouping, and considering their learning paths, the system can recommend contents that are in line with the contents already covered by the student's peers.

We conducted experiments as part of an effort to improve the adaptability of the Portugol Studio (PS) platform (Noschang, Pelz, & Eliezer, 2014). PS is a beginner-oriented programming IDE (Integrated Development Environment) that is widely used in Brazilian universities for teaching programming. SACIP (Esteves, 2021) is an adaptive system that uses AI techniques to recommend programming content for beginners as a Portugol Studio plugin. It was developed with the idea of using learning paths (Santos, Gomes, & Mendes, 2013) to teach programming. This type of approach enables the student to learn to program, preferably using topics of which he or she already has knowledge. Making effective connections between programming and the students' interests can make learning more attractive and less complex.

Learning path is described as the chosen route taken by a learner through a range of (commonly) e-learning activities, which allows them to build knowledge progressively (Scott, 1992).

This work aims to answer the following research question: "Can the use of the LDA algorithm be effective to generate groups of students considering their preferences and past experiences, rather using those of an individual student to generate recommendations close to what is considered ideal?" We argue that a recommendation based on the preferences and learning paths of similar students in a group can reach values close to the optimal.

The remaining parts of the paper are organized as follows. Section 2 describes related researchers. Section 3 presents the methodology, explaining all involved procedures. Experiments are presented in section 4 and obtained results in section 5. Following, discussion and conclusions are presented.

2 RELATED RESEARCH

We analysed research papers that used the LDA technique to recommend educational content. These are briefly described below.

Apaza et al. worked on an online course recommendation system. The LDA algorithm was used to define the main topics of each course. The online courses are the MOOC courses (Massive Open Online Courses), which openly welcome many students. Due to the large number of existing courses, students are interested in working on courses that have topics of their choice. This preference was defined from the grades of students in the college and the recommendation is made by comparing this student preference with the topics of each course discovered by the LDA.

Erkens et al. developed the GRT (Grouping and Representing Tool). The project's objective is to seek to form heterogeneous groups of students to apply collaborative learning. This tool uses text mining with LDA to identify students with similar backgrounds. These contexts are defined from various texts written by students during their school career. The similarities are found and the differences between the students are analysed to define the heterogeneous groups in each course discovered by the LDA.

Lin et al. uses AI techniques to recommend educational resources for distance learning courses. These resources are recommended to students according to their needs, hobbies, and interests. Online courses are evaluated according to the students' grades and the length of use of each one. The student's preferences and needs are evaluated by an LDA using a three-layer Bayesian model. The input layer deals with static data (personal information, management, and security) and dynamic data (learning, interests, etc.). From this information, the hidden layer can infer the features, presenting in the output layer those most recommended to students.

3 METHODOLOGY

The recommendation procedures were developed for the SACIP (blind review) system. Fig. 1 shows its simplified architecture, composed by two modules: (i) User Agents and (ii) Recommendation system.

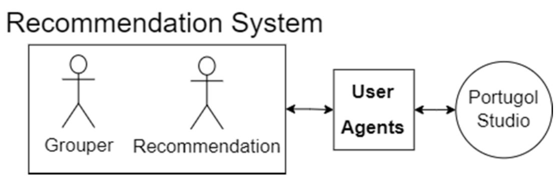

Figure 1: SACIP simplified architecture.

The methodological procedures include data collecting, data preparing, creating groups of students, and performing the recommendation procedures. These steps are presented in the following subsections.

3.1 Data Collection

User agents are responsible for handling communication with the recommendation module, monitoring the actions of each user in the system, their choices, their learning paths and recording student data in the database.

These collected data enable the system to obtain knowledge to deliver contents that are in accordance with the student's level of knowledge, so that they can learn with reference topics that are of personal interest to them. The data registered are as follows:

- Academic Grade: which grade the student is currently in. This attribute allows the system to define the level of logical-mathematical knowledge the student has and can affect the way the system poses easier or more difficult problems.
- Age: attribute that registers the student's current age. It can be useful since students of the same age group may have similar interests.
- Preferences: a list of different interests that the student has selected in the system. It is used in two stages: (i) to find students with similar interests and (ii) to find content that includes topics of interest to the student.
- Path: includes all contents that the student has used in the system, sorted by selection. It is used to define the student's current knowledge. The path also enables the system to discover the differences between the paths of students in the same group, allowing for new recommendations.

These data obtained by the interface agents are used by the system, enabling it to recommend individualized content for each student. These collected data enable the system to obtain knowledge to deliver contents that are in accordance with the student's level of knowledge, so that they can learn with topics that are of personal interest to them.

3.2 Data Preparation

To evaluate the system, content and students were randomly created. After creating these databases, it was possible to simulate the system's recommendations for interested students. Randomly generated students do not have purposeful resemblances to real students, but they do have some patterns in relation to the characteristics that were important for the simulation.

Table 1 presents the attributes of students, simulated content and how each one was generated. The tag values and preferences used comprise a list with thirty-one different selected themes.

Table 1: Variables used at dummy students' creation.

Users		**Contents**	
Name	Random String	**Name**	Random String
Password	Random String	**Description**	Random String
Avatar	Random String	**Topic**	Value between the 15 topics
Sex	Male, Female, Trans, Other	**Complexity**	Math, Cognitive, Algorithmic, Code
Scholar Grade	Middle School, High School, College.	**Exercise**	True or false
Age	A number between 12 and 42	**Taxonomy**	A number between 0 and 5
Preferences	A value from a list of themes	**Tags**	A value from a list of themes
Path	Content with tags in its preference	**Level**	A number between 1 and 5

These themes were chosen arbitrarily; their name has no relevance to the simulation, as it does not deal with the relationships between the themes. The list of themes is shown in Table 2.

Table 2: Theme tags used when creating dummy students.

cars	music	animes
geography	math	language
image	memes	myths
monsters	youtube	comedy
superheroes	history	sports
cartoons	animation	games
biology	animals	pets
marvel	books	international
culture	movies	technology
science	toys	food

To set student preferences and content tags, each content can have 1-3 tags, while each student can have 3-5 preferences. Both are set randomly. Fifteen topics were made available to be used by the contents. Topics do not have names; they only vary between t1 and t15. They are distributed in groups of 3, over 5 levels of difficulty. It has been established that topics 1 through 3 are level 1, 4 through 6 are level 2, 7 through 9 are level 3, 10 through 12 are level 4, and 13 through 15 are level 5.

3.3 Creation of Student's Groups

To group the students by their characteristics, the Grouping agent uses the Latent Dirichlet Allocation (LDA) clustering algorithm (Blei, Ng, & Jordan, 2002). LDA is a probabilistic model that allows any number of documents with multiple K words (user-decided value) to be grouped by topic. It uses

Dirichlet's distribution model to define the probability that a document will belongs to a predefined topic. Within a space of k topics, each document is approximated to a topic, based on the words it contains. Also, each word present in the documents is given a probability of belonging to a defined topic.

LDA uses information on preferences, age, and educational level of each student. These words are transformed into strings, so that each string of words represents a document. It was also established that k was defined seeking an average of 10 students per group, with at least 2 groups. Therefore, k is defined as: (Number of students/10) + 1. This value is rounded to an integer if the result is a real value.

The steps in the execution of the clustering algorithm are as follows:

1. A text document is created for each student, containing their characteristics and preferences.
2. The LDA is fed with all students' documents and the k value, and then executed.
3. LDA returns two objects: (i) list of topics contained in documents and (ii) list of documents belonging to each topic.
4. Groups of documents are generated for each topic. If there is more than one group in a document, the document is placed on the most likely to belong.
5. It is checked which student each document belongs to. Students who own the document are placed in the group of students with documents on the same topic. Similar student groups were created with this procedure.

After these steps, the recommendation procedures, presented below, can be performed.

3.4 Recommendation Procedures

The procedures are performed by the Recommender agent, in collaboration with the Grouper agent. The procedures can be summarized in the following steps:

1. Carry out the analysis of the student's path and check which topics he/she has already studied, and at which levels.
2. Perform a search in the database of all the contents belonging to the levels that the student has already completed.
3. Check which group the student belongs to, then analyse each path of each student in that group. Search for the most common content among them that has not yet been studied by the student and add to the recommendation. If the student has no peer group, this step is skipped.
4. Analyse the student's path, looking for the next recommended taxonomies by topic.
5. Search for content related to the recommended taxonomies in the content list of step 2.
6. Filter the contents of the taxonomy by the student's preferences, looking for contents that suit his or her interests, and add the contents to the list of recommendations.
7. Review the topics already covered and see which topics the student has not yet studied that will allow him or her to complete the lowest level not yet studied. If there are no topics to complete, the next level content is recommended. If there are topics to be completed, the contents of the list obtained in step 2 are filtered by each of the topics to be completed.
8. Filter the content resulting from step 7 by student preferences and add to the recommendations list.
9. Score the contents in the recommendation list. Scoring gives priority to taxonomy and then to student group and preferences. Content with recommended taxonomy earns 10 points, contents with tags equal to the student's preferences earn 1 point each, and contents belonging to the student group earn 1 point for each student who has the content on their track.
10. Sort and list recommended content in rank order.

The first 10 contents of this list are sent to the Pedagogical agent (Esteves, 2021), as possible recommendations for the student. Attributes such as education level, preferences, sex, and age group, among others, can be used by the agent to decide on the best ones to recommend. Contents are also recommended by taxonomies and levels already completed. This prevents the student from receiving content recommendations for levels that he or she has not yet completed and may have difficulty understanding.

4 EVALUATION

The evaluation consists of verifying the recommend ability of the approach using LDA clustering techniques. To fulfil this objective, the entry of students into the system via the SACIP plugin of Portugol Studio was simulated. For the validation and testing procedures the following steps were performed:

1. Dummy content data deployment in the knowledge domain.
2. Development and execution of a procedure to simulate the creation of students in SACIP, as well as content requests.

3. Development of a procedure executed by the Grouper agent that uses the LDA algorithm.
4. Execution of the recommendation algorithm
5. Accuracy assessment of recommendations performed by SACIP.

Step 1 implements 3 content instances with 100/200/500 content units for 3 different test environments. These contents must have random data ranging from 5 difficulty levels, 15 learning topics and 31 themed tags. Of the themes, each content can have 1 to 3 randomly generated tags.

Step 2 creates the student and content requests until it has completed its path. This system runs 1k times, creating 1k students during the simulation. Students must have different personal interests that are in accordance with the available content tags. These interests were chosen randomly from the list of 31 content tags generated in step 1. Each student can have 3-5 interest tags.

Step 3 performs a procedure within the SACIP that obtains the data of each one of the recommendations that a student receives. The data obtained by this procedure are: (i) student's name; (ii) student group; (iii) content recommendations; (iv) topics to be covered by the student; and (v) the level of recommendation level.

Step 4 stores the attributes referring to the requests made by students in tables, using the following information: (i) relevance of the group to the student; (ii) relevance of the content recommended for the student; (iii) adherence to the best content recommended to the student; and (iv) adherence to the best student content obtained in the database. This information will be used to assess the accuracy of the recommendations made by SACIP.

Step 5 analyses the attributes of each content and the characteristics of the students to find out if the contents correspond to the students' interests as described by their tags. This was done as follows: (i) comparing the content themes with the students' preferences; and (ii) comparing the best content in preferences with what was recommended.

In the measurement, the themes of each column were scored by tags and verified if the number of tags not relevant to the student are greater than the number relevant to the student. Next, a percentage was attributed, denoting how much the recommendation matched the student's interests. At the end, an average recommendation score was generated and recorded in the columns of the tables.

The obtained results are stored in tables with 1k lines, where the lines represent the student who entered the system, and the columns are those described in step 4.

During testing, each student: *(i)* is created; *(ii)* logged into the system; *(iii)* asked for a recommendation for each topic; and *(iv)* logged out of the system. At each student creation, the LDA algorithm is reorganized to include the new student among the students already created before it, if any. When running the system, there are no students at first. Each new student created logs into the system and asks for 15 recommendations, one for each topic. At the end, the student logs out and the data of the recommendations given to the student are recorded.

5 RESULTS

Each experiment with 100/200/500 contents generated 5 data tables, which are described below:

I. Group relevance: for each student, the group the student belongs to was checked, and the ten most common preferences of that group were obtained, along with that student's preferences.
II. Recommendation relevance: for each content request, the system's recommendations, their tags and how many student tags this recommendation has were registered.
III. Comparison of the best content: the best content from the system recommendation and the best content directly searched in the bank are obtained. The best content evaluation is based on the number of tags the content has, which is related to the tags the student has.
IV. Group Relevance Average, Recommendation and Best Content Comparison: for each of the data defined in (i), (ii) and (iii), a table was made with the average of these values for each student.
V. Average of all registered students: based on the results presented in 4th step table, a final table was drawn up, showing the average scores obtained by each student. In this way, it is possible to find out if, on average, the system can make a good recommendation.

The data in the tables from steps 4 and 5 were graphed and displayed in Fig. 2, 3 and 4. Each x-axis value in the graphs represents the number of students registered in the system at the time the new student is logged in and entered a group. The y axis represents the percentage of adherence of the attribute to the student's interests or best-case content. The first graph (Fig. 2) is related to the relevance of the groups to which each student was allocated during the tests.

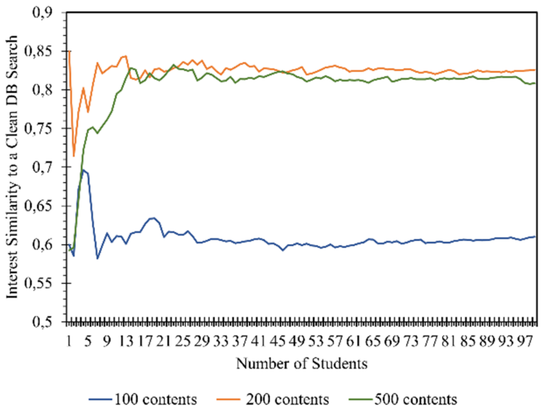

Figure 2: Cluster group relevance by the algorithm to the student.

To check relevance, the ten most common tags in the student's group were obtained. Each student's preference was analysed, to see how many of his or her interests were contained in these ten common tags. The final value, as a percentage, showed how many of the student interests matched those of the group. On average, adding the scores for all the students, the relevance match was around 27%.

Fig. 3 shows the recommendations relevance, with tests carried out for the three amounts of content, i.e., 100/200/500. In this case, it is the tags generated randomly for each student that varies. For each student, all the relevance of their recommendations were summed, and an average score calculated. The mean represents the total relevance value of the recommendations the students received.

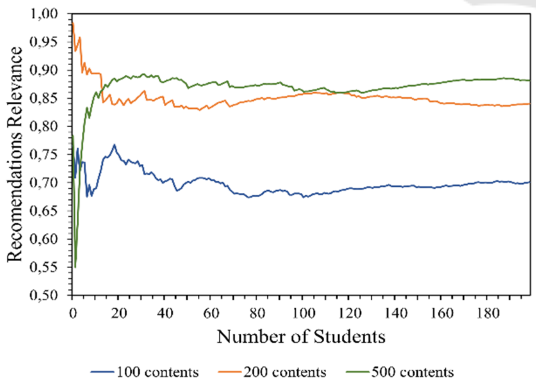

Figure 3: Relevance of recommendations to the student.

To carry out the experiments with the recommendations, all the contents recommended by the system were captured, and all the tags were registered. These tags were compared with the student's preferences and the number of student interests that matched these recommendations was counted. The generated value was transformed into a percentage of relevance of the content recommended for the student.

Tags were not sorted by number of occurrences as the system will always recommend according to the student's preferences rather than the highest occurrence of tags. Therefore, the final score for the average relevance of the recommendations in each test was 67% for 100 contents, 84% for 200, and 88% for 500.

The next experiment (Fig. 4) is carried out in two steps: (i) analysis of the student's path; and (ii) search on the database for the content with the most student interests, without using the algorithm.

In the first step, the student's path is analysed, and the next topics to be covered, at the student’s level, are determined. The contents are then filtered by these topics and arranged in order of the number of tags that relate to the student's interests. The first item listed is selected as the content of most interest.

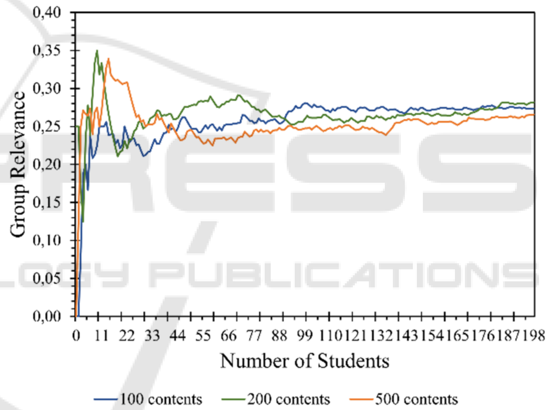

Figure 4: Adherence of the best recommended content.

In the second step, for each recommendation, a search is performed in the database, looking for the content that has the tags of interest that most resemble the student's interests, and that is content at the current level. Once the recommended content is known, the best case is retrieved for this content existing in the database without using the algorithm. Taking the best case as a basis, an adherence calculation (A) is made in which each content tag has a value of $A = 1/T$, where T is the total interests that the student has. Next, the average of these scores is calculated, and the average adherence to the contents recommended by the best-case contents obtained in the database without the algorithm is calculated. The closer to 1 this final value is, nearer the best possible case is to the recommendation.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

During the experiments, it was found that there are cases of repeated recommendations among the topics. This means that when receiving a recommendation for a content, the student preferred to select a different content instead of the recommended one. This action causes the system to re-recommend the previous content because it remains the most appropriate.

6 DISCUSSION

The systematic literature review was useful because it enabled us to find out the main existing approaches, and which of them use similar techniques. The approaches found and selected apply the LDA technique using student texts or individual characteristics to find topics of interest to them.

Our approach uses the LDA technique with data previously obtained about the student's profile, their preferences, and the learning paths they went through. Based on this information, the LDA algorithm generates groups by similarity, and the contents are recommended, considering, in addition to the profile attributes, the knowledge of the paths taken by the student and by other students in the group. Thus, the knowledge gained from the paths of other similar students in the group can be used to benefit the recommendation.

When analysing the results, the student's adherence to the group to which he or she was allocated by the LDA algorithm was first verified. On average, the group obtained about 27% relevance for the student considering the 10 most common interests. This means that on average, at least 1 of the students' interests is common in the group.

A direct search in the database for students with the highest number of tags similar the required student may be better than using LDA for clustering. However, for the definition of groups, the algorithm also considers data such as age and education, among others. For large-scale use, a direct search considering these values would be much more complex and laborious, and less effective than using LDA.

Next, the relevance of the recommendations was verified, determining the average adherence of the best recommended content to the student's interests (Fig. 3). The high relevance ratings obtained from the recommendations are intuitive; it is not difficult to recommend content within the topics of interest to the group. The main information that this graph presents is the difference between the tests. From 100 to 200 contents there was a significant increase in adherence. However, from 200 to 500, adherence did not significantly increase. This shows that at around 200 contents, the algorithm reaches a good limit, but more contents do not make a significant difference in adherence to the student's interests.

Finally, a comparison was performed between the best recommended content and the best content searched directly in the database (Fig. 4). In all cases (100/200/500 contents), the recommended content obtained from the database manually had the most interests of the student. However, there was a pattern of about 80% similarity where there were 200 or more contents. This similarity is very high, which means that the contents recommended by the system are close to the best possible. Also in this case, it is possible to see how the increase in the number of data influenced the increase in content adherence. From 100 to 200 contents, the adherence of recommended contents improved significantly, with almost 20%. From 200 to 500 there was an improvement in both, reaching close to 90%. With more than 200 contents, there is a smaller, but gradual improvement. In a way, this confirms what had already been seen in another indicator, that for more than 200 contents there is no significant improvement in the recommendation.

Through the experiments carried out, it was noticed that 100 contents did not manage to reach the students' interests very well. At between 100 and 200 contents, there was a progressive improvement, and from 200 contents onwards, less significant improvements were observed in the recommendation. This shows that around 200 contents are needed for the algorithm to be able to generate recommendations that are close to those considered ideal.

The information obtained from the experiments also demonstrates how the recommendation by LDA can be very similar to the ideal search. It is coherent to assume that in a real scenario, with several students, some preferences may tend to appear together in groups of students, making the groupings more strongly related.

There is a strong tendency, in real situations, for adherence rates to improve still further. For example, a student who enjoys Marvel is likely to also enjoy superheroes, so many students may appear with these two preferences on their profiles. This correlation cannot occur with randomly generated students.

7 CONCLUSIONS

This research analyses the recommendations made by the LDA algorithm with different volumes of content, for a growing number of students. The experiments were carried out using randomly generated content and students. The goal was to find the best possible recommendation technique for the SACIP system.

The obtained results indicates that the application of the LDA algorithm to create groups and generate recommendations provides a good result in delivering content and practices that are in accordance with the student's interests. The applied technique reaches values very close to the recommendation considered optimal, confirming the research hypothesis.

The approach differs from those of previous studies in the literature that use LDA techniques, since recommendation techniques are based on the experiences of a group, not just an individual student. The purpose is to acquire knowledge about preferences from a group of students and use them to present content that might make sense to a learning student.

Furthermore, it is not always possible to obtain accurate information about the student's preferences. This occurs on the first access to the system, for example, or when the student is not sure what he wants. The knowledge acquired by the paths taken by students with similar characteristics can help or even encourage it to enjoy the recommendation.

New experiments with students and real content are part of the next stage of the work. It is possible that many correlations among interests will be lost when students and content are randomly generated. The hypothesis that the real results will be better than those obtained in these experiments may be proven. Also, other clustering techniques can be applied and compared to achieve the best possible type of clustering to be used in adaptive systems.

REFERENCES

Apaza, R. G., Cervantes, E. V., Quispe, L. C., & Luna, J. O. (2014). Online courses recommendation based on LDA. In *CEUR Workshop Proceedings* (Vol. 1318, pp. 42–48).

Blei, D. M., Ng, A. Y., & Jordan, M. T. (2002). Latent dirichlet allocation. *Advances in Neural Information Processing Systems*, 3(null), 993–1022. Retrieved from https://doi.org/10.5555/944919.944937

Erkens, M., Bodemer, D., & Hoppe, H. U. (2016). Improving collaborative learning in the classroom: Text mining based grouping and representing. *International Journal of Computer-Supported Collaborative Learning*, 11(4), 387–415. Retrieved from https://doi.org/10.1007/s11412-016-9243-5

Esteves, A. M. da S. (2021). *Sacip: Sistema Adaptativo Construcionista Para Iniciantes Em Programação No Portugol Studio Utilizando Trilhas De Aprendizagem*. Universidade do Vale do Itajaí - Univali.

Hoed, R. M. (2016). *Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de Computação. xvi, 164, [8] f., il. Dissertação (Mestrado Profissional em Computação Aplicada) — Universidade de Brasília, Brasília, 2016.* Universidade de Brasília. Retrieved from https://repositorio.unb.br/bitstream/10482/22575/1/2016_RaphaelMagalhãesHoed.pdf%0Ahttp://repositorio.unb.br/bitstream/10482/22575/1/2016_RaphaelMagalhãesHoed.pdf

INEP. (2018). Censo da educação superior 2017: divulgação dos principais resultados. Retrieved from https://bit.ly/359LMlV

Lin, Q., He, S., & Deng, Y. (2021). Method of personalized educational resource recommendation based on LDA and learner's behavior. *International Journal of Electrical Engineering Education*, 002072092098351. Retrieved from https://doi.org/10.1177/0020720920983511

Lobo, R. (2017). A Evasão No Ensino Superior Brasileiro – Novos Dados. Retrieved 24 July 2021, from https://educacao.estadao.com.br/blogs/roberto-lobo/497-2/

Noschang, L. F., Pelz, F., & Eliezer, A. (2014). Portugol Studio: Uma IDE para Iniciantes em Programação. In *Anais do XXII Workshop sobre Educação em Computação* (pp. 535–545).

Santos, Á., Gomes, A., & Mendes, A. (2013). A taxonomy of exercises to support individual learning paths in initial programming learning. In *Proceedings - Frontiers in Education Conference, FIE* (pp. 87–93). IEEE. Retrieved from https://doi.org/10.1109/FIE.2013.6684794

Scott, P. H. (1992). Pathways in learning science: A case study of the development of one student's ideas relating to the structure of matter. *Research in Physics Learning: Theoretical Issues and Empirical Studies*, 203–224.