A Recommender Model for the Personalized Adaptive CHUNK Learning System

Daniel O. Diaz
Department of Operations Research,
Naval Postgraduate School,
Monterey, CA

Ralucca Gera
Department of Applied Mathematics,
Associate Provost for Graduate Education,
Teaching and Learning Commons,
Naval Postgraduate School, Monterey, CA
Email: rgera@nps.edu

Paul C. Keeley
Department of Information Sciences,
Naval Postgraduate School,
Monterey, CA

Matthew T. Miller
Department of Operations Research,
Naval Postgraduate School,
Monterey, CA

Nickos Leondaridis-Mena
Department of Operations Research,
Naval Postgraduate School,
Monterey, CA

***Abstract*—Recommender systems attempt to influence one's behavior based on explicit and implicit information provided by the users of the system. Users who take part in e-commerce or watch cat videos online will be familiar with this concept. Different algorithms exist that determine what objects or concepts to recommend to users, but every one of them has the similar goal of providing a *good* recommendation. In this context, *good* means that the recommendation will be user relevant suggesting accurate topics, and will influence the user's behavior. Additionally, a good recommendation system is adaptive, consistently seeking feedback from the user. Feedback is then used to make the next recommendation better. In this work, we develop a recommendation methodology for an existing personalized learning system, where both content and teaching methodology options are presented to the user. Our methodology provides solutions to both the user and the network coldstart problems, where little up-front information is available in order to make good recommendations. Using real system data, we show how our method recommends the most relevant learning topics and styles and incorporates user feedback to improve future recommendations.**

***Index Terms*—Education; Chunk Learning; Adaptive Learning**

I. Introduction and Motivation

The Internet changed the world in many ways. Globally, it transformed the way people conduct banking, commerce, communication, and even warfare. The field of education is no exception, yet lagging behind other fields. Educational websites like Khan Academy [1], Chegg [2], and Coursera [3] allow people to personalize their educational path and interfaces at all levels of learning. These websites provide students a way to broaden their learning at their own pace, in a linear fashion similar to a text book, broadcasting the same information to everyone. Factors that once limited one's learning, such as the personality or the delivery method of an instructor, can be reduced by personalizing the learning experience using a network of knowledge, and we seek it in this work.

One such network is CHUNK Learning [4], a website that provides "a modular real-time and adaptive teaching-learning method for enhanced and personalized education which enables the student to heuristically discover and learn based on personal background and interests" [4]. The CHUNK acronym is from the Curated Heuristic Using a Network of Knowledge. The CHUNK Learningsystem allows users to explore topics via a Graphical User Interface (GUI), and choose topics and teaching methods matching their *personal* interests. Figure 1 shows the CHUNK's user interface as of March 2019.

Each red bubble, called a CHUNK, represents a topic, and within each topic are various learning modules called CHUNKlets. The content of CHUNKlets is uploaded by instructors, and users can graphically explore connected topics and view content as a network rather than linear fashion. This mimics more of a map of the world view for learning, rather than back to back linear chapters learning.

While the visualization is helpful to get a global view of the network of knowledge, the website only provides an initial set of topic recommendations based on keywords in the user's profile. In the current work, we explore to improve this selection of topics by adding an adaptive mechanism to generate future, more tailored recommendations, other than users being directed on what topic to learn. We seek to incorporate a recommendation system within CHUNK that recommends relevant topics to each learner based on profile information about the learner. This will better align with the goal of CHUNK Learning system to support life long learners whose new knowledge builds on and relates to the previous learner's skills and knowledge. Throughout the paper, "user" and "learner" are used interchangeably.

 ISBN: 978-1-61208-725-2

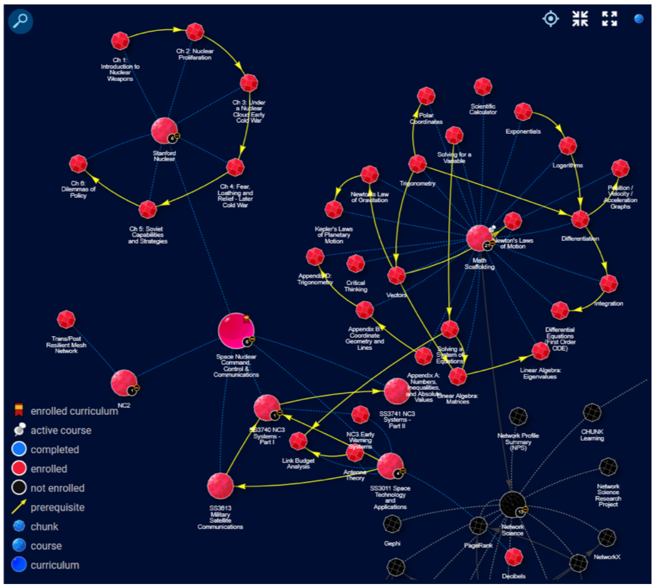

Fig. 1: CHUNK GUI. The main view that users see when initially logging on to the system.

The structure of the paper is as follows. We begin with establishing the needed definitions and the problem statements in Section II followed by an overview of the related work in Section III. We then introduce the methodology for computing similarity between the users in this environment and the methodology for recommendations in Section IV. We present the experimental setup in Section V, followed by the results and interpretation in Section VI. We conclude and present further direction in Sections VII and VIII.

II. Definitions and Problem Statement

The CHUNK GUI learning system presents a network view of a variety of different courses, each of which has a collection of public-facing 30-60 minute modules called "CHUNKs" (shown in Figure 1). Each CHUNK represents a different topic, much like a section in a textbook. At a more granular level, each CHUNK is composed of a varied number of components called "CHUNKlets", with each CHUNKlet presenting a different explanation on the same topic. The goal of presenting these different views is to personalize a learner's experience in exploring a topic, based on learner's interests, learning styles and previous knowledge. Furthermore, a CHUNKlet may belong to one or more CHUNKs, depending on the CHUNKlet's applicability to topics of those CHUNKs.

The CHUNK Learning system also carries a user profile for each learner. This is populated with data on courses the user has explored, as well as information about the user's preferred learning method, existing skills, and topics of interest. The data captured in the user profile is used to present the user with a map of all the CHUNKs associated with that registered course. The edges of the network capture natural progressions through the topics based on prerequisites at the topic level, thus allowing users to move around the network with a global view of how the topics build on each other.

Once the user chooses a CHUNK to study, the user is presented with a selection of associated CHUNKlets. These CHUNKlets could be videos, slide shows, research papers, code, websites, or various other methods of facilitating information, with four purposes in mind: (1) a "Why" CHUNKlet motivating the topic, (2) a "How" CHUNKlet showing how subject matter experts use the topic in real life, (3) a "Methodology" CHUNKlet as lectures or activities teaching a skill, and finally (4) an "Assessment" CHUNK as a set of knowledge assessments for the topic of interest of the particular CHUNK.

The intended audience is made up primarily of two groups–what we term as the "exploratory learners" and the "directed learners." Directed learners are students who are directed to some course(s) or CHUNK(s) within the system–perhaps by an instructor to refresh or re-mediate them on a particular topic. Exploratory learners are students who are interacting with the system in a more open-ended fashion–perhaps to learn about a topic related to something they are studying or perhaps to familiarize themselves with an unrelated discipline. These two categories are not mutually exclusive–in fact, a desired outcome of our proposed recommendation system is that directed learners become exploratory learners as a result of meaningful, serendipitous recommendations. Both, directed and exploratory learners, have choices on how to progress through the CHUNKs associated with a course. Yet exploratory learners may choose a sequence of CHUNKs that are associated with many different courses.

In this paper, we examine the **user cold-start** problem for both types of users: How best to match a new user to material that fits his or her interests and learning style; particularly when we assume little to no knowledge of the user's actual preferences. We assume that the profile information provided by the average user is incomplete, and it will be updated as the learner progresses through the CHUNKs, making it easier to suggest CHUNKs at that point. In particular, we assume that the directed learners will provide the least amount of information, since we also assume that their motivation to provide information is the lowest.

The second aspect we research in this effort is the **network cold-start** problem: With little user data on-hand, how do we best acquire useful information over time to identify emergent connections and apply collaborative filter methods? Putting in another way, how does the network improve its recommendations and internal connections through implicit or explicit feedback?

III. Related Work

Technology Enhanced Learning (TEL) develops and tests technical innovations that will support and enhance learning practices based on technology. Such an introduction to TEL and recommender systems (RS) building on the educational information retrieval supporting life long learning can be

 ISBN: 978-1-61208-725-2

found in [5], [6]. There is a large group of TELs and papers [7] on TEL are regularly presented at Social Information Retrieval for Technology Enhanced Learning (SIRTEL) and expanded on topics presented at SIRTEL such as [8], [9] Through these means "interesting conclusions, the main ones being that:

- There is a large number of RS that have been deployed (or that are currently under deployment) in TEL settings;
- The information retrieval goals that TEL recommenders try to achieve are often different to the ones identified in other systems (e.g. product recommenders);
- There is a need to identify the particularities of TEL recommender systems, in order to elaborate on methods for their systematic design, development and evaluation" [5].

Further work have extended these topics to introduce a "paradigm for building intelligent systems that can better predict and anticipate the needs of users, and act more efficiently in response to their behavior" and summarized in several surveys [10], [11] as well as a summary of Mobile multimedia recommendation in smart communities [12].

A survey of educational RS that appear in the literature show various approaches to solving both of the cold-start problems identified above [13], [14], as well as summaries of requirements for such systems [15], [16]. These include the application of ontologies, rulesets and content-maps to filter, interactive tagging and sort responses to queries, with the goal of guiding a learner through a topic in a progressive fashion. User profiles are frequently generated in an explicit, manual fashion, similar to the existing CHUNK method; however, unlike the current CHUNK implementation, many of the systems described actually expand or augment user profile data following interactions with the system.

Typical of many examples are educational RS which use several of the techniques outlined above in a multi-round fashion to attack both the network and user cold-start problems. Albatayneh at al [17] incorporate semantic indexing and negative feedback to both determine a user's level of knowledge and recommend content in a logical, progressive manner. However, the use case (curating and recommending posts on an e-learning forum) has a much higher level of user interaction than we expect for CHUNK. This type of context-aware RS is akin to that described in [18], which seeks to identify users' strengths and weaknesses using the common Knowledge, Skills and Abilities paradigm. We see this as a well-suited approach to a highly structured learning environment (such as primary education), but we believe it would need constraint relaxation to provide a good fit for the exploratory learner model that we apply to CHUNK. Applying a more strictly ontological approach, another multi-round RS is developed by [19], which first matches content with user queries, then rates and selects offerings based on an ontological mapping of topical material paired with an adaptive knowledge of user mastery. In this case, an ontological map provides the structure needed to overcome the network cold-start problem, while the feedback gained through assessments is used to develop and update user profiles.

In the current work, we present a hybrid networked approach to overcome the cold-start problems identified above. We view users and content as nodes on a network, and we combine elements of content-mapping with syntactic sorting to determine a user's initial location on this network. We incorporate feedback and learning objective completion to update the user's location in the network of knowledge and then provide the user with recommendations to help guide him/her through the network.

IV. METHODOLOGY

In our modelling, we strip the CHUNK network of any ontological structure. Therefore, when we save the data, CHUNKs are not connected to each other via a topic umbrella or any prerequisite relationship. This allows the user to be unfettered in his or her path through the network. In contrast to the existing network, we create a strongly connected network of CHUNKs, where each CHUNK can be reached from every other CHUNK. For visualization and comparison purposes, we compute a similarity value between each pair of CHUNKs and display our new network in Figure 2. Note the logical grouping of CHUNKs into communities based on CHUNK title. While the ontological structure of the network is absent, a natural structure occurs based on similarity values. The methodology for computing this similarity value will be described in this section.

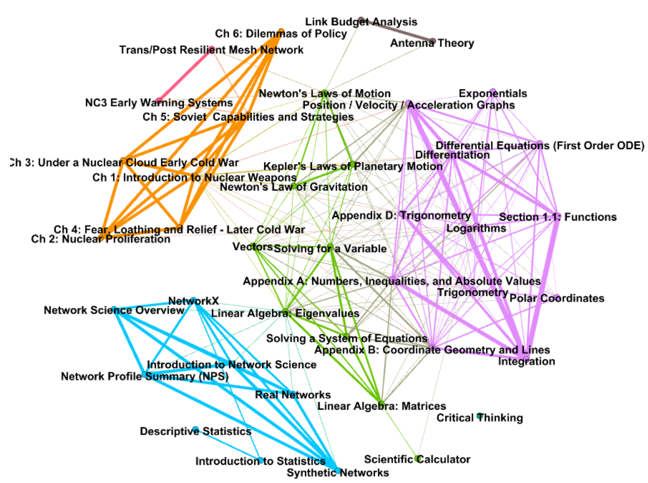

Fig. 2: CHUNK network using syntactical closeness and arranging by modularity class.

In order to make relevant recommendations, our recommendation system relies on computing similarity values between pairwise CHUNKlets, the user and each CHUNK, and subsequently between the user and each CHUNKlet. To compute the similarity value, we use the cosine distance

 ISBN: 978-1-61208-725-2

between two vectors in a $1 \times k$-dimensional space or a $1 \times \ell$-dimensional space, where k and l are the cardinalities of the network's CHUNK or CHUNKlet keyword sets, respectively. The CHUNKs and/or CHUNKlets (across all CHUNKlet types) with the highest similarity value relative to the user are recommended first. Before providing a methodology for computing this similarity value, we outline system information and structure requirements:

1) Initial System Inputs. The system resides in an information database, where each entity (CHUNK, CHUNKlet, and user) is identified with a profile(s). This profile has a unique identifier, a set of keywords, and, in the case of a CHUNK-CHUNKlet, a parent-child relationship. System administrators decide on CHUNK titles, and instructors upload CHUNKlets. When CHUNKlet upload occurs, the instructor must do four things: define the parent-child relationship between the CHUNKlet being uploaded and the CHUNK that it is assigned, categorize the CHUNKlet with one of the four categories "Why", "What, "Methodology", or "Assessment", assign to the CHUNKlet content keywords, and assign to the CHUNKlet learning method keywords (Video, PowerPoint, etc.).
2) User Profile Vectors. Two profile vectors will be built for each user: one based on content keywords that will be used for computing similarity values between the user and each CHUNK, and one based on learning method keywords that will be used for computing similarity values between the user and CHUNKlet. The first will be a $1 \times k$-dimensional vector, where k is the cardinality of the network's content keyword set, and the second will be a $1 \times \ell$-dimensional vector, ℓ being the cardinality of the set comprising learning methods keywords. The system populates the user's vectors when the user initially creates his or her profile. It is a binary vector, where a one represents the user's interest in that keyword, and a zero represents no feedback or negative feedback in that keyword. The manner in which the system obtains these keywords from the user during initial profile build is left to the current system administrators.
3) CHUNKlet Profile Vectors. CHUNKlets have two profile vectors: a $1 \times k$-dimensional content keyword vector and a $1 \times \ell$-dimensional learning method keyword vector. They are populated when the instructor uploads the CHUNKlet into the CHUNK Learning system based on that instructor's input.
4) CHUNK Profile Vector. Similar to the user's content keyword vector, the CHUNK's keyword vector is $1 \times k$-dimensional, but it is not a binary vector, rather it is the sum of the vectors of its CHUNKlets. That is, the value associated with each keyword position in the vector will be based on the parent-child relationship between each CHUNK and CHUNKlet. The keywords associated with the CHUNKlet that the instructor tagged during upload will aggregate within the CHUNK, and this aggregated number will be the value for the keyword's position within the vector. Therefore, unlike the user's initial content keyword vector of ones or zeros, the CHUNK's keyword vector is not limited to a binary value.

Figure 3 shows a possible data structure representation of these vectors. The top row is the user, and the rows beneath the user represent CHUNKs. The column titles are the keywords.

	absolut	aerospac	aircraft	algebra	analysi	angl	architectur	area	astronaut	aviat	...
Nickos	0	0	0	0	0	0	0	0	0	0	...
f13deca2-27f8-4c59-9499-64a73b81b6e8	0	0	0	0	1	2	1	0	0	1	...
79dc2521-3b1f-4062-8675-1823f73a7bdd	0	0	0	5	0	0	0	0	0	0	...
11fa0110-d428-4eea-bccf-6d81d6d43c35	0	3	1	0	0	0	0	0	1	0	...

Fig. 3: CHUNK data. Possible keyword data structure representation.

Now that the system has its requisite information and appropriate vector lengths, we can compute the cosine distance between vectors and provide as recommendations the CHUNKles with the highest cosine distance value. We do this in a two-round process.

Recommendation Round. Using the standard linear algebra cosine distance formula, we compute the distance between the user's keyword vector and all CHUNK keyword vectors. CHUNKs are then ranked from highest to lowest similarity value, and the first ranked CHUNK is recommended first. The user can accept or reject the CHUNK that is recommended, but we focus here on users that will always accept the first recommendation. Once the user accesses the CHUNK, another cosine distance is calculated between the user's learning method vector and all CHUNKlets associated with the current CHUNK. The closest m CHUNKlets for each CHUNKlet type are recommended in decreasing order, where m represents the desired number of CHUNKlets shown based on system administrators' input.

User Feedback Round. During this round, the user completes CHUNKlets within the current CHUNK. Implicit feedback, such as the length of videos watched, may be captured during this phase, but we do not focus on those possibilities here, rather capture it in the future work section. Our focus is on explicit feedback, which will be captured at the completion of each CHUNKlet and CHUNK.

In the CHUNKlet case, the user will be presented with a choice of rating the CHUNKlet as either a "like" or a "dislike". The user's learning method profile vector will then be adjusted by multiplying a scalar value to the vector entry associated with the CHUNKlet type, expanded upon later in this section.

In the CHUNK case, the user will be presented with the same "dislike" or "like" question regarding the CHUNK

 ISBN: 978-1-61208-725-2

as a whole, but if the user indicates positive feedback, a second feedback question will be asked. To support an adaptive CHUNK Learning system, this feedback round presents the user with the top three keywords (based on frequency) associated with the CHUNK and asks the user for either positive or negative feedback for each of the three keywords. The feedback collected will then impact the keywords attached to the CHUNK.

Lastly, to make the profiles adaptive, the user's profile vector will then be adjusted by multiplying a scalar value to the keyword(s) position in his or her content keyword vector. Additionally, if the user indicates positive feedback on any of the three keywords shown at the end of the CHUNK, and that keyword is not already represented in the user's keyword vector, a "1" value will be added to the user's keyword vector before the scalar is applied. This enables the user to prolong his or her exploration in the CHUNK Learning network by making it possible for related CHUNKs to be suggested to the user.

We set the "like" scalar value to 1.05 and the "dislike" scalar value to 0.01. These values can be adjusted depending on system administrator preference. Because of these updates, we refer to the CHUNK Learning system as having "dynamic profiles", since each user's profile adjusts according to explicit feedback.

Upon completion of a CHUNK, that CHUNK's similarity value to the user profile will be assigned the value zero. This is to prevent the user from being recommended a CHUNK that has already been completed.

The process then repeats. It should be noted that our methodology is applicable to both directed and exploratory learners. For the directed learner case, users may take a different path through the network than a purely exploratory learner might, but they can still use and benefit from the feedback mechanisms built into the system particularly in respect to the learning methods presented over time.

V. Experimental Set Up

Using the CHUNK data available as of January 2019, we conduct two distinct experiments in order to compare the exploratory performance of our RS against the performance of the current CHUNK Learning system.

The first experiment looks at network discovery using both systems. For this simulation, three users are created with unique profiles containing distinct sets of keywords. As a base case, one more profile is created that did not have any keywords. Each user starts the exploratory process by taking the CHUNK named "Vectors". To model our RS, we apply the similarity methodology to the user's profile in order to model the shift in paths from user to user. To differentiate the impact of user feedback, each user provides positive feedback on the CHUNK taken at each iteration. The user's profile is then updated with the top three keywords in each CHUNK profile studied, if the keywords are not already present in the profile. To model the existing CHUNK Learning system, because the user's profile does not change, feedback is not applied, and the user profile gains no new keywords. The model runs through twenty iterations and records the CHUNK taken as well as the similarity value between the user and that CHUNK during each iteration. We then visualize and analyze the simulation output using Gephi.

The second experiment is designed to demonstrate our system's ability to adjust CHUNKlet type recommendations based on user feedback. Because the current CHUNK Learning system does not apply a profile updating step, this experiment is only applicable to our new RS. Due to the infancy of the network, there was a lack of data when each CHUNKlet breaks down into each of the four categories for recommendations. It would be difficult to assess the true performance of the feedback mechanism on sparse data. Therefore, for this simulation we build only one user whose initial profile shows affinity for only one learning style, represented by its CHUNKlet type. We then simulate the user going through fourteen CHUNKlet recommendations, where the user's feedback to each CHUNKlet varies. At each recommendation, we record the CHUNKlet type as well as the "likeability" that the user has with each CHUNKlet type. We analyze the output data using Python.

VI. Results and Analysis

We begin by analyzing the network discovery data based on the different exploratory paths taken by a user with a dynamic profile as compared to a user with a static profile. Figure 4 shows the path that a user with a dynamic profile. The width of the edges is proportional to the similarity of the user to that CHUNKlet, represented by the red lines forming the path the user takes.

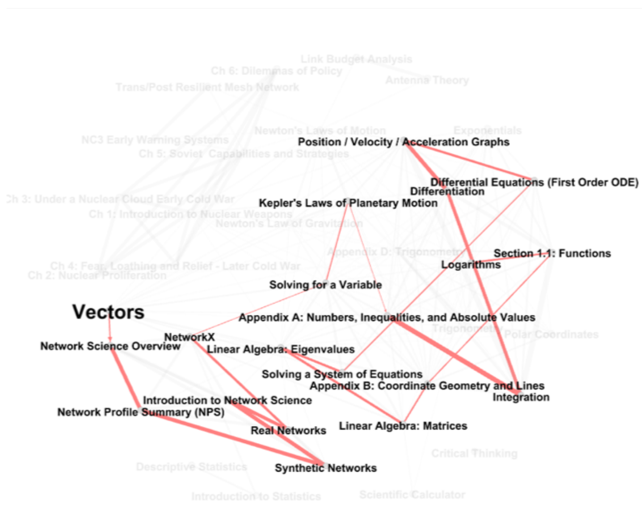

Fig. 4: Exploratory path based on a dynamic profile.

Figure 5 shows a user's path with a static profile. In this case, the red lines end when the user's similarity values relative to the remaining CHUNKs drop to zero. Because the user's profile is not updated at the end of each CHUNK, the user cannot acquire new keywords. At this point, the

 ISBN: 978-1-61208-725-2

user's next CHUNK is chosen at random, since no relevant recommendation can be given, and the remainder of his or her path is denoted by green lines.

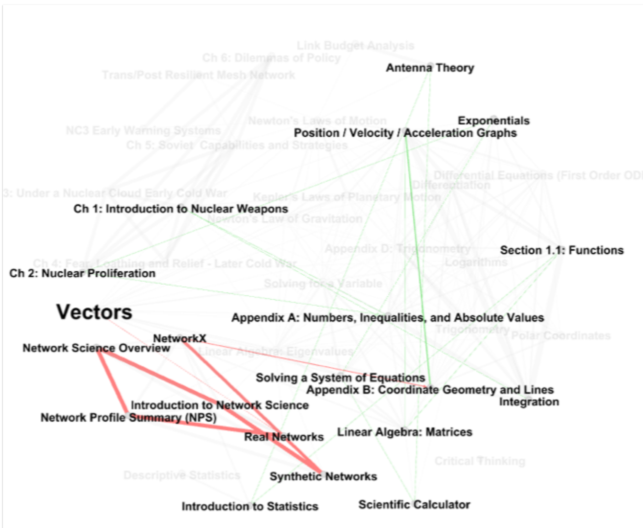

Fig. 5: Exploratory path based on a static profile.

From Figure 4 and Figure 5, we see that updating a user's profile at the end of each CHUNK prolongs the user's relevant exploratory path through the network. Since our network's construct does not incorporate prerequisites or any ontological structure, it is important that the user's profile be updated in order to provide both a logical and meaningful progression of CHUNKs.

We next demonstrate network discovery by showing different paths taken by unique users. Figure 6 and Figure 7 outline paths taken by users whose initial set of keywords showed an interest in Physics and Space, respectively. Since no randomness was used in any steps, the different paths taken by each user demonstrate that our recommendation system provides unique recommendations based on user input. The accuracy and relevancy of these recommendations is, of course, dependent on the system's data. However, by noting the different CHUNKs that are recommended to each of the users, we see that the system points users in directions that appear appropriate and relevant.

We have graphically demonstrated that our recommendation system positively impacts network discovery based on profile updates as well as differences in initial user profile input. Next we present results on how the similarity value between the user and the CHUNKs, as well as the user and the CHUNKlet learning methods, changes over time based on user feedback, given a dynamically updating profile.

Figure 8 shows the change in similarity values between a user with a Nuclear and Space centered profile, and four of the nine most similar initial CHUNKs to the user. Five of the nine CHUNKs were removed in order to keep the chart readable. From Figure 8, we see that the CHUNK with the highest similarity value relative to the other CHUNKs is chosen before the others. This is a simple observation that

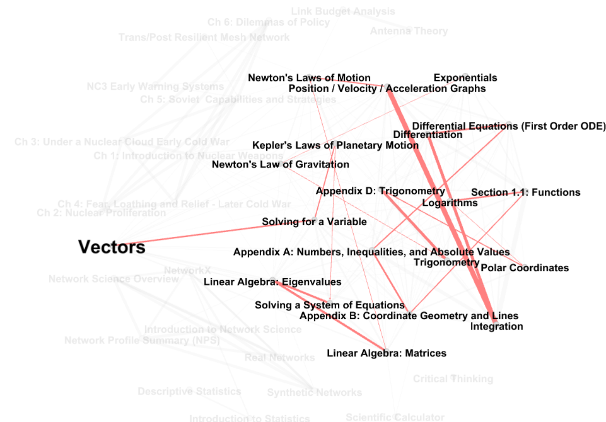

Fig. 6: Path for a student interested in Physics.

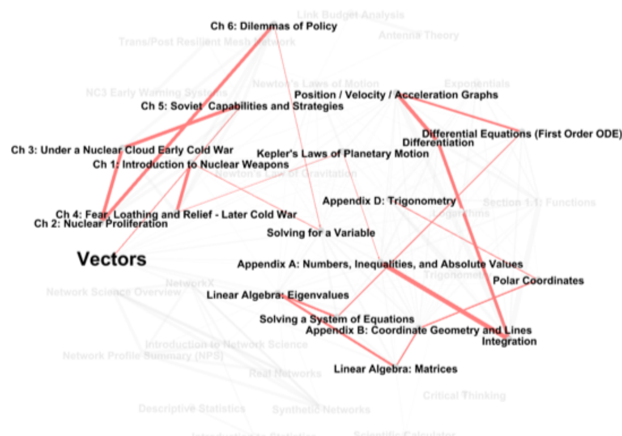

Fig. 7: Path for a student interested in Space.

should not come as a surprise based on our methodology. Once the CHUNK with the highest similarity value has been "completed', its similarity value decreases to zero and that CHUNK is no longer considered for recommendation. Thus the next CHUNK chosen has the next highest similarity, and lower than the one of the just completed CHUNK.

Two noteworthy observations are: (1) The similarity value between the user and a group of similar CHUNKs decreases as that user completes each CHUNK in that group. While we did not explore the underlying reason behind this behavior, our hypothesis is that as the user gains keywords, he or she is becoming more of an "expert' and less of a "generalist', bringing the similarity value down as the user progresses through the network. (2) Based on the dynamic nature of the user's profile, our system recommended the CHUNK "Kepler's Law", which started with a similarity value of zero, after nine iterations of CHUNK completions. This further demonstrates that the use of a dynamically updating profile enables the user to prolong his or her exploratory learning experience.

 ISBN: 978-1-61208-725-2

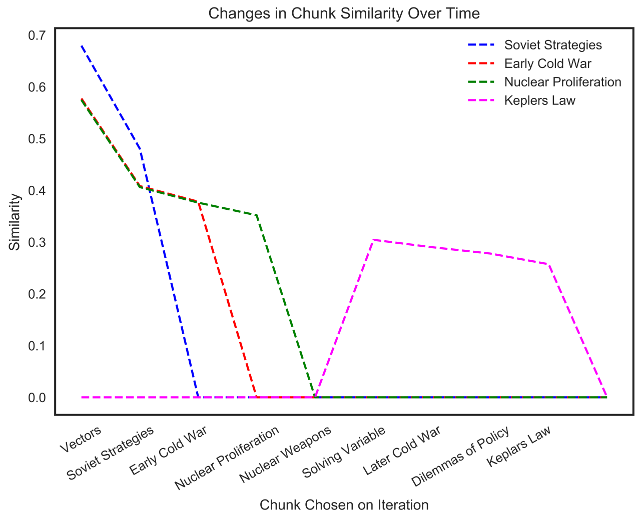

Fig. 8: Changes to CHUNK similarity values as a user completes CHUNKs.

In addition to observing changes in CHUNK similarity values over time, seeing the result of CHUNKlet feedback is also important in order to demonstrate to the reader the recommendation system's ability to adapt to each user in a unique way, as well as how adjusting system parameters can influence system behavior. Figure 9 shows CHUNKlet recommendations over the course of fourteen CHUNKlet completions.

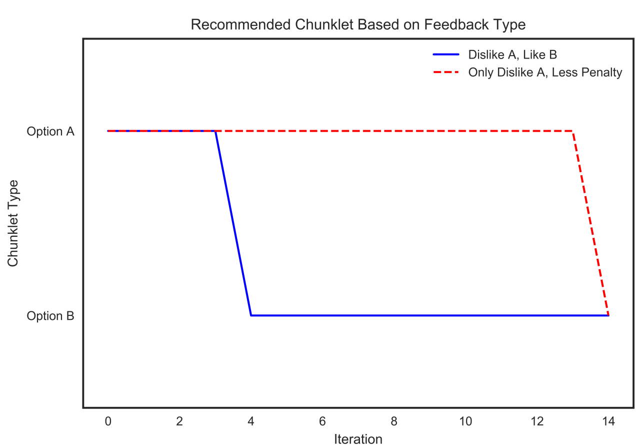

Fig. 9: CHUNKlet recommendations over time based on different user feedback.

To demonstrate the adaptability of our feedback mechanism, in our second experiment we repeatedly show the user the same two CHUNKlet options- Option A and Option B. Each option represents a different learning style. Option A represents a CHUNKlet labeled as a video while CHUNKlet B represents a CHUNKlet labeled as a PowerPoint. In both cases, our user initially indicates that they prefer videos on their user profile. However, as soon as they are shown a video CHUNKlet, they realize they dislike videos. When the user signals that they dislike videos, we apply the 'dislike' penalty scalar to their user profile. To demonstrate the effect of the penalty value, we display two lines. The red line represents a penalty of .5 while the blue line represents a penalty of .01. As shown in Figure 9, the user repeatedly dislikes videos until the system randomly selects them a CHUNKlet containing a PowerPoint. Once they see a PowerPoint, the user signals that they do in fact prefer this style of learning.

While not groundbreaking, the second experiment demonstrates how the severity of the penalty correlates to how quick the system adapts and responds to user behavior. A 'dislike' penalty scalar of 0 immediately removes keywords from the user profile while higher penalty values introduce a lag. The penalty offers the system administrators another layer of flexibility in how they choose to control the feedback loop.

VII. Conclusions

In this work, we describe a method for providing relevant and personalized topic recommendations as applied to the CHUNK Learning system. By storing topic and method keyword counts in vectors, we are able to compute a simple similarity value between the user of the CHUNK Learning system and each CHUNK as well as each CHUNKlet. Those CHUNKlets with the highest similarity values are then recommended first, to include the CHUNKs they are part of. Secondly, user feedback provides a method for dynamically updating the similarity calculation in order to promote the most relevant information to the user throughout his or her use of the CHUNK Learning system. Through multiple simulations, we have demonstrated that this methodology provides unique and accurate recommendations to the user based on his or her profile and feedback.

While our simulations show proof of concept, time and data limitations only allowed for simple feedback behaviors and a limited number of user profile builds. In future, one can perform sensitivity analysis on our methodology.

VIII. Further Directions

Our team barely scraped the surface on providing a comprehensive recommendation system for CHUNK, rather we looked to test the possibility of an adaptive system. Below are numerous suggestions for follow-on research:

1) Feedback Method. Our feedback mechanism only allows the user to either "like" or "dislike" a CHUNK, CHUNKlet, or keyword. However, various other feedback methods exist that are not necessarily binary in nature, which may provide more relevant or accurate recommendations than our binary response. For example, user interactions with CHUNKlets could be tracked using metrics such as video/module completion to get implicit feedback on content or learning method relevancy. Alternatively, the exercises and knowledge checks already included as CHUNK assessments could

 ISBN: 978-1-61208-725-2

tagged down to the individual question level with supporting content keywords. If a user is struggling with questions associated with one of these supporting areas, those keywords could be added to his or her profile to generate future recommendations.

2) Keyword Updates. We implement a way for the user to explore the CHUNK network by dynamically updating the user's profile based on keyword feedback at CHUNK completion. These keywords are chosen based on overall count in the CHUNK. Choosing these keywords in this manner, however, may not be the best way of ultimately providing the user with the most opportunities to explore the various topics in the network. It may indeed be a limiting factor depending on the sparsity of the data or the current similarity of the CHUNK to the user. Instead of choosing the top three keywords by count, the system could calculate all of the possible similarity values based on all $\binom{n}{3}$ combinations of keywords and then providing the combination that supports the best exploratory option. Many other possibilities can be devised. On the topic of keyword collection, manual methods would soon prove cumbersome, especially when videos are concerned. We recommend an automatic keyword scraper instead.
3) Collaborative Filtering/Recommendations. At the time of this work, the CHUNK Learning system was in its infancy. Once the system has time to incorporate many users and CHUNKlets, it will be possible to incorporate collaborative filtering and recommendations.
4) Exploiting social tagging in the TELs much like it has been done for the Web 2.0 recommender system [20].

ACKNOWLEDGMENTS

The authors would like to thank the DoD for partially funding this work.

REFERENCES

[1] "Khan academy," https://www.khanacademy.org/, Mar. 2019.
[2] "Chegg," https://www.chegg.com/, Mar. 2019.
[3] "Coursera," https://www.coursera.org/, Mar. 2019.
[4] R. Gera, M. Isenhour, D. Bartolf, and S. Tick, "CHUNK:Curated Heuristic Using a Network of Knowledge," in The Fifth International Conference on Human and Social Analytics. HUSO, July 2019.
[5] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, and R. Koper, "Recommender systems in technology enhanced learning," in Recommender systems handbook. Springer, 2011, pp. 387–415.
[6] O. C. Santos, J. G. Boticario, S. Baldiris, G. Moreno, D. Huerva, and R. Fabregat, "Recommender systems for lifelong learning inclusive scenarios," in Workshop on recommender systems, 2008, p. 45.
[7] H. Drachsler, K. Verbert, O. C. Santos, and N. Manouselis, "Panorama of recommender systems to support learning," in Recommender systems handbook. Springer, 2015, pp. 421–451.
[8] H. Drachsler, H. Hummel, and R. Koper, "Identifying the goal, user model and conditions of recommender systems for formal and informal learning," 2008.
[9] R. Vuorikari and R. Koper, "Ecology of social search for learning resources," Campus-Wide Information Systems, vol. 26, no. 4, pp. 272–286, 2009.
[10] K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachsler, I. Bosnic, and E. Duval, "Context-aware recommender systems for learning: a survey and future challenges," IEEE Transactions on Learning Technologies, vol. 5, no. 4, pp. 318–335, 2012.
[11] M.-Á. Sicilia, E. García-Barriocanal, S. Sánchez-Alonso, and C. Cechinel, "Exploring user-based recommender results in large learning object repositories: the case of merlot," Procedia Computer Science, vol. 1, no. 2, pp. 2859–2864, 2010.
[12] D. Gallego, E. Barra, A. Gordillo, and G. Huecas, "Enhanced recommendations for e-learning authoring tools based on a proactive context-aware recommender," in 2013 IEEE Frontiers in Education Conference (FIE). IEEE, 2013, pp. 1393–1395.
[13] Ricci, Recommender systems handbook, 2nd ed. Springer, 2015.
[14] J. K. Tarus, Z. Niu, and A. Yousif, "A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining," Future Generation Computer Systems, vol. 72, pp. 37–48, 2017.
[15] H. Drachsler, D. Pecceu, T. Arts, E. Hutten, L. Rutledge, P. Van Rosmalen, H. Hummel, and R. Koper, "Remashed–recommendations for mash-up personal learning environments," in European conference on technology enhanced learning. Springer, 2009, pp. 788–793.
[16] H. Drachsler, H. Hummel, and R. Koper, "Personal recommender systems for learners in lifelong learning: requirements, techniques and model," 2007.
[17] N. A. Albatayneh, K. I. Ghauth, and F.-F. Chua, "Utilizing learners' negative ratings in semantic content-based recommender system for e-learning forum," Educational Technology & Society, vol. 21, no. 1, pp. 112,125, 2018.
[18] J. Underwood, "Metis," in Educational Recommender Systems and Technologies, 2012, pp. 24,42.
[19] A. Ruiz-Iniesta, G. Jiménez-Díaz, and M. Gómez-Albarrán, "A semantically enriched context-aware oer recommendation strategy and its application to a computer science oer repository," IEEE Transactions on Education, vol. 57, no. 4, pp. 255,260, 2014-11.
[20] A. B. B. Martinez, M. R. Lopez, E. C. Montenegro, F. A. M. Fonte, J. C. Burguillo, and A. Peleteiro, "Exploiting social tagging in a web 2.0 recommender system," IEEE Internet Computing, vol. 14, no. 6, pp. 23–30, 2010.