Cognitive Textbook Generation

Project Summary

Abstract

Through history, humans have acquired an abundance of knowledge and information, salient in decision making and judgement. The spread of information is a primary factor in such knowledge persisting till today. However, the actual transfer of knowledge between people is challenging: the recipient cannot fully grasp the concepts immediately. To address this problem, textbooks have been written to store the overall information for a concept. Since textbooks are written for a target audience, any individual who misaligns with this audience will have difficulty understanding the material. Our system provides a solution by generating customized textbook content that is catered specifically to the individual. In doing so, the user does not have to expend as much time and effort to learning and thus, can easily pick up new concepts. By using a knowledge graph, the system can map out all the principles for a particular concept. The user can provide feedback to the system by highlighting segments or clicking the like/dislike action, highlighting their preferences of learning. The system will also administer small assessments to measure the user's knowledge. Using both these learning preferences and current knowledge metrics, the system can construct a model of the user's learning behavior. Finally, the system conjoins the knowledge mapping and the user model to output textbook content specific for the user.

Intellectual merits

Our system incorporates the knowledge graph, learner model and uses them both in the pedagogy model to generate the customized textbook material. Being able to construct a mapping of the knowledge components of a concept is salient, since this structure provides the basis of the information that a user will need to understand. The learner model provides valuable insight into the user's learning capacity and tendencies and an accurate model is essential for the system to comprehend the user. The pedagogy model use of the ties both the knowledge and learner models together and is responsible for computing the exact manner in which knowledge concepts need to be represented that best appeals to the user. The specific conjunction of all these models that our system makes use of is vital in understanding the process by which humans attain knowledge.

Broader impact

By outputting individualized textbook content, our system greatly simplifies the learning process. The user can read through the material that suits their natural learning style and in doing so can better gain the knowledge about the topic. Presently, with an increasing percentage of the population undertaking higher education, many more students can use their time more efficiently with tools like our system. Students will not face as much frustration in learning and can attain more value from their educational programs. In an alternate perspective, our system also empowers teachers to better cater to their students. With an average of 27 students for each teacher in public secondary schools, the teacher will face difficulty in attributing time for all the students. Our system can ease this pressure as the student can obtain the main information for a concept and the teacher can act more as a facilitator. Finally, with a multitude of children receiving little attention in their education program, personalized teaching can greatly assist their learning experience. Our system offers this as the student can grasp the knowledge in a manner best suited to them.

Project description

Objectives

Challenge Statement:
As the human race has developed over the years, a great deal of information has been acquired. Some of this information is considered useful for individuals to know so that they can better understand the world, make informed decisions, solve problems, and generally have an improved chance of achieving their goals. The challenge is that this knowledge is not easily transferred from one individual to another. A person cannot simply download information into their brains. Instead, people go to school to learn facts, principles, and techniques related to a variety of subjects such as mathematics, science, and history. While there is an education system in place to facilitate this information transfer, the general principles at work are still largely a mystery. What changes take place in the human brain while viewing educational material? Why do some students understand a concept after the first explanation while others require many repetitions? There are still a lot of challenges to be addressed in the domain of education, but this project specifically focuses on the following challenge: Suppose that there is a specific concept that a specific person desires to understand. How can a textual presentation of that material be custom generated for that person so that they can most efficiently reach an understanding of the concept? An ideal solution to that problem would allow any concept to be explained to any person interested in learning that topic. If a machine existed that could generate such individually optimized material, then the effort required to learn would be greatly reduced. People could spend less time learning, avoid the frustration often associated with learning, and more easily pick up new concepts as they are needed.

In order to address the stated problem, there are a few different areas that require investigation. *Intelligence Unleashed* [1] describes a framework for understanding the process of education using three models: domain, learner, and pedagogy. A teaching tool, such as the one proposed in this project, must combine the information from all three models in order to teach the learner. The domain model encodes an understanding of the subject matter and is the source of knowledge used to generate material to present to the learner. The domain model is a knowledge representation and may include specific facts and ideas as well as connections which organize those fact and ideas into a comprehensive understanding of the concept. The learner model encodes an understanding of the person seeking knowledge. This includes a breakdown of what the person already has learned and what preferences they have for receiving new information. Developing an accurate learner model requires collecting data from the learner. By assessing the learner's knowledge before and after presenting them with information, the tool can evaluate the effectiveness of various techniques to discover which ones the learner prefers. That assessment may be done by comparing the learner's responses to what is represented in the domain model. Additionally, data can be collected while the learner is viewing the educational material. This can provide information about how engaging a learner finds specific techniques. Finally, the pedagogy model describes the various approaches that can be taken to transfer knowledge. This can include how to present an individual segment of material and also how to organize a lesson plan for the concept as a whole. The teaching tool can combine these three models to decide how to best present information to the learner. The specific information from the domain model is presented using techniques from the pedagogy model which require information about the prior knowledge and preferences that are encoded in the learner model. Thus by updating these models as the learner interacts with the tool, the learner can receive increasingly customized content and learning efficiency will continue to improve.

Any format of interacting with the learner can fit into the three model system just described, but this project focuses on presenting textual information. Most people have had the experience of reading a textbook in order to learn about a particular topic. The content to be produced by the learning tool would be similar to what is typically seen in textbooks. Paragraphs of prose would be the primary content. Tables, lists, and possibly figures can be interspersed where appropriate. While the output will be similar to textbook content, the goal is to improve upon textbook learning by addressing key issues that limit the efficiency of textbook based learning. When a textbook is written, the author must have a target audience in mind and the content is presented in a manner that is understandable to that audience. A person with a background that differs from that target audience may find the material difficult to understand. Even learners within the target audience may encounter issues when reading the textbook. This is because the contents are written once and targeted to a group of people rather than an individual. A related issue with existing textbooks is that they are written to cover a large topic in detail. The individual sections typically reference content from other sections and fully understanding the contents may require a cover-to-cover reading. This project seeks to avoid these limitations by generating content that is tailored for a specific learner. Thus all the material for a topic does not need to be presented in a general way; instead the exact information sought by the learner can be presented in a format that they can easily comprehend. Additionally, the progression from one sub-topic to another is in the control of the system, and can be another source of individualization.

By providing individualized textbook style content to explain a topic, the efficiency of learning is improved. The learner receives material presented with their preferred techniques and in a way that takes into account their existing knowledge. There may also be an efficiency gain on the part of the textbook authors since they are no longer tasked with the layout and presentation of the information. The authors can focus on encoding their expertise in the domain model and rely on the tool to handle the details of presentation. As a result, learners can obtain a more complete understanding of the topics that they are interested in with less effort.

Desired Outcomes:
Our system incorporates the three models in generating the customized textbook content for the individual. To revisit, these models are the domain, learner and the pedagogy models. Each model has its own respective outcomes that will need to be met so that the overall system function can be satisfied. For instance, within the domain model, all the vital components of a concept will need to be enumerated as entities. Moreover, all the appropriate relationships will need to be formed so that a complete knowledge ontology can be constructed for the concept. This will avoid any information gaps in the model and thus, the system can generate a comprehensive excerpt for the given subject. Likewise, the learner model will also need to properly comprehend the user. The model should quickly and accurately determine the user's knowledge for the topic at any time. This way, the system can ascertain whether the user is actually grasping the content and can immediately adjust the content depending on the level of understanding.

A vital aspect of the system is to determine the user's learning preferences. Without wasting too much time on surveys or other convoluted questionnaires, the learner model needs to rely on more efficient means of feedback that also provide valuable insight into the user's learning style. Since no individual has a stagnant learning behavior, the learner model should be able to predict and adjust to the user's gradual shift in preferences. Finally, the pedagogy model, being the core component of the system, needs to compute the best possible mapping between the domain model and the learner model. All the entities and relationships need to be structured that best suits the user's learning preferences and current proficiency. With this mapping formed, the system can accordingly generate the content that both covers the concept and is catered to the user. As all these three models maintain these outcomes, the system can effectively maintain its goal of best helping the learner in understanding topics.

Measures of Success:
As our system is primarily focused on generating textbook content specialized for an individual, the main measure of success is whether the user has understood the underlying topic. The learner model has a component to measure the user's knowledge and the system can use this to assess how effective the content is. If the user scores well on a test after reading the customized content, then we can determine that the system performed successfully. An alternate way to check whether the user actually progressed their understanding is to administer two tests before and after reading the content. The difference in scores highlights the gain in knowledge as a result of using the system, assuming that the user did not review any relevant material outside the textbook contents. Aside of just attributing the system's success with the user's understanding, being able to adapt to the user is just as salient. The system could generate material that poorly fits the user's learning tendencies but they ended up scoring well because they received tutoring from an external source. In this case, the system clearly fails in achieving the goal. It is vital to ensure that the generated textbook content is best suited to the individual and that the user improves their understanding of this topic because of reading the material.

While our system is mainly beneficial to students, it can also assist teachers and authors. Teachers often find themselves burdened by having to instruct a lot of students. The system can reduce their workload as the students can learn more about the topics by reading through the specialized textbook content. Freeing up the teacher's time should be considered a success for the system since it allows the teacher to facilitate the student's education in more effective ways. Similarly, the system can reduce an author's efforts in writing textbooks. Instead of writing a new version of the textbook every year, the author can simply annotate the domain model and the system can dynamically structure this content for any individual.

Prior work

With our project specifically centralized on generating textbook contents adapted to an individual's learning tendencies, we have only uncovered related projects that are loosely similar to both these aspects. Brusilovsky et. al. [2] elaborate an approach to constructing web-based textbooks that are adapted to different groups of users. Firstly, this method develops a domain model that is a mapping of all the concepts of a given topic. It also constructs a user model that keeps track of the user's knowledge of these concepts, which is then overlaid upon the concept model. Doing so enables the framework to compute the ideal way to structure the concepts in the webpage to exemplify the user's learning experience. However, this approach has a shortcoming where the content stored for each node in the domain model is static making it poorly generalizable to students. Our approach will need to circumvent this problem and ensure that the information in the domain model is dynamic and best suited towards the individual.

From an author's perspective, it would tremendously help if they do not have to write any new versions of a textbook for different target audiences. Stewart et. al. illustrate that with the MOT authoring system and the WHURLE delivery system, one can create a single adaptive version of educational material that can be catered to many different contexts [3]. In a similar implementation to the Brusilovsky et. al.'s, the author constructs a domain concept mapping to piece together how the different parts of content should relate to one another. Goals and constraints can also be embedded onto this mapping such that the author can fine-tune how the content should be conveyed, whether through an image, text or sound. WHURLE can determine the user's characteristics through a small questionnaire and accordingly deliver the material in the best representation for the user. A problem is that WHURLE only considers a limited number of factors that are all strictly related to learning style. One's learning style can vary through more intricate parameters such as health, psychological, economic and social factors. The model proposed by El-Bakry and Saleh targets this problem [4]. Just as in WHURLE, a student takes a brief questionnaire to evaluate their state along ten different dimensions, including the ones mentioned above. The model constructs a mapping that allows it to identify the best teaching style for the given results. Given the teaching style, the appropriate lesson plan can be developed that best caters to the student.

In both these implementations, using a basic survey to understand the user's learning behavior is quite incomplete. Determining complex characteristics with simple questions is not ideal. Furthermore, a user's attitude is constantly changing and relying on the fixed results of a survey will not be useful. To better account for the user's complex behavior, our system will need to detect the user's emotions, actions, and past history and create a mapping of these variables. This allows for our system to output the textbook contents to precisely match the individual's preferred way of learning at any point in time. Another way to improve upon these systems is to utilize a continuous observation process and constantly acquire data from the user. With this data, the system can further improve its model of the user's learning style and generate even more applicable textbook material. Therefore, our system should take advantage of this reinforcement procedure in order to constantly improve its effectiveness.

Domain Model:

A core component of this system is the domain model which encodes the knowledge of the subject matter that will eventually be presented to the learner. Representing knowledge to be used algorithmically is a problem that other systems face, and there is some prior work that has been done. One existing approach to knowledge representation is to construct a graph where the nodes are entities and the edges are relationships. This is referred to as a "knowledge graph" or "knowledge base". Some example (entity, relationship, entity) triples that can make up a "knowledge graph" are: (Earth, is part of, the solar system), and (carbon, has atomic number, 6). This format of knowledge representation has been used by Google to provide the summary boxes that appear with search results in addition to the traditional list of links. A search for "carbon" will return a summary box with key value pairs such as: Chemical symbol: C, and atomic number: 6. By using this representation, existing systems are able to provide a summary of an entity, or answer simple questions such as "what is the atomic number of carbon?". By constructing a complex graph of such triples, the systems are able to handle a wide range of queries. Since the queries handled by systems using these "knowledge graphs" are fairly simple, recent research has been focusing on how to automatically predict new relationships and fill in gaps in the knowledge base [5]. The system being proposed here seeks to provide more detailed explanations of the subject matter, and so new research is needed to enhance the knowledge representation to support more detailed explanations.

The domain model of this system needs to support richer content generation, but it also needs to guide a learner through a topic. Prior work has been done in existing intelligent tutoring systems (ITS) to organize the material within a course. Existing research [6] has been done based on the common approach of organizing course sub-modules into a directed graph. For example, the graph for a data structures course would have a root node that represents introductory material on abstract data types. From that root node there is a choice of which sub-topics to cover next. Before covering any sub-topic, all of the nodes which represent prerequisites must have already been covered. In this way, structure is provided to the user of the ITS, and material is only presented once the learner has the necessary prior knowledge. Depending on the learner's goal and current knowledge, the system can decide what sequence of topics is best to cover. The directed graph of sub-topics is also useful for the system being proposed, but the content at each node will need to be different than what is used by current systems. Current systems have "pre-canned" text, or other content that is covered for each sub-topic. Instead, this system's domain model needs to provide enough information to generate content customized for the learner. The “knowledge graph” described above is a starting point for encoding the content for an individual sub-topic, but more research is needed in order to support richer content generation. In addition, there is an opportunity to perform research on how to better utilize the directed graph of sub-topics to provide content across sub-topics. For example, the system can utilize general information from the “trees” sub-topic while generating content to explain “depth-first search”, because “trees” is a prerequisite for “depth-first search”.

Measuring Knowledge:

To determine if the system is generating effective content for the user, it will need to measure their knowledge. Commonly, assessments like tests or quizzes are administered to evaluate how much a user understands a concept. Forms of testing such as longitudinal and cross-sectional are more granular, as they can measure a student’s gain in knowledge after going through a lesson [7]. In longitudinal testing, a student takes a test before undergoing a course. They will take a similar test after finishing the overall course. The difference in scores essentially highlights the student’s gain in knowledge as a result of undertaking the course. Longitudinal testing is certainly effective but will be tedious for the student since they need to take multiple tests. Cross-sectional testing avoids this problem, as the incoming students’ results can be compared with statistics from previous years. Our textbook generation system can gain valuable insight from using these methods of testing but there are limitations. Moreover, the system will need to avoid the limitations of these approaches, as longitudinal testing requires too much effort and cross-sectional requires statistics that the system might not have.

Another method to measure learning is to take advantage of learning objectives and detailed rubrics [8]. Having a checklist of objectives for a certain concept makes it easy to identify how much the student knows about the subject. Similarly, a rubric also breaks down a concept into weighted components. A student’s learning can be scored against a rubric to identify how much of an understanding they have of the concept. Usually, a textbook provides learning objectives for each chapter, highlighting the key points that the student needs to know. Also, teachers make use of rubrics to evaluate one’s understanding against these learning objectives. Typically these objectives and rubrics are generic and might not properly capture understanding for different kinds of people. As part of customizing textbook content for the individual, our system can also adapt these objectives to the student. Consequently, the system can gain better insight into the student’s level of learning.

In contrast to the typical test based approach, more qualitative assessment tools can be used to measure the user’s knowledge. For instance, methods like interviews, document analysis, reflective journaling and such can better evaluate a user’s insight [9]. Using objectives can actually limit a student’s learning as they could ignore various interpretations of how one’s learning is occurring. By disallowing room for subjective exploration, objectives can fall short in properly measuring understanding. Qualitative assessments can circumvent this problem but it is hard for our system to adopt. Conducting interviews or journaling can be quite expensive for the system and the user might not engage in these processes fully. Furthermore, textbook content might not require such level of qualitative assessment since the common focus for the student is to gather the fact-based knowledge for a concept.

Learning Preferences:

In the field of education, it is evident that there exists some grouping in which activities may fall, often with varying levels of success towards knowledge acquisition. Although there are likely a wide variety of factors that contribute to which strategy would work best, and when, it seems reasonable to assume that one such factor would be personal preferences towards different methods. It is expected in our system that a required component would be learning these different preferences for each student, in hopes that the system will better model output for each user as an individual. Luckily for us, there is past research in this area found in Carmona, Castillo, and Millán's paper *Discovering Student Preferences in E-Learning*.

In this research paper, the authors investigate the relationship between different teaching strategies and the multi-media format that the information is presented in. In this paper, they make the observation that a teaching system which generates content based on the user's preferences "is quite similar to a content-based recommender system" [10]. It is with this thought that we began to view this component of our system for what it truly is: a subsystem that attempts to minimize the error of output text based on some user interaction. In this research, the authors began to discover some interesting concepts worth exploring in our system. One such concept is that of an unbalanced feedback system, a negative review should not necessarily be as heavily weighted as a positive review, or vice versa. Another was that of the importance of model decay, to account for the inevitability of user preferences changing over time. One interesting implementation of theirs was that of a K-Dependence Bayesian Classifier, where they discovered that a rather small value of K resulted in near optimal results.

Plan of activities

The work needed to achieve the objectives can be broken down according to the following system diagram.

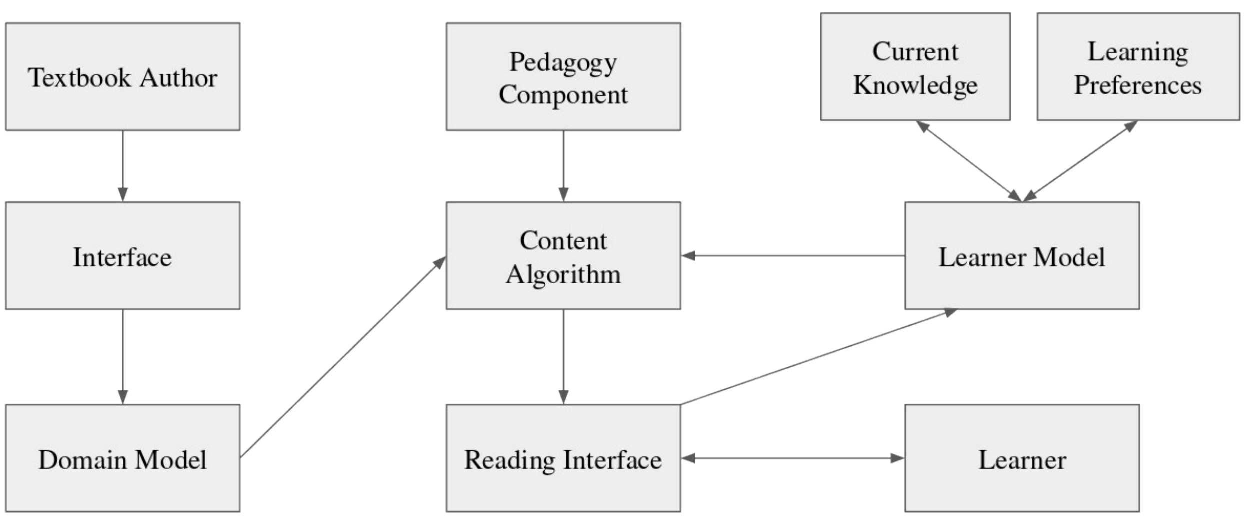

Figure 1: System Diagram

At the center of the diagram is the "Content Algorithm" which decides what to display to the learner on the "Reading Interface". The "Content Algorithm" requires input from three other components: the "Domain Model", the "Learner Model", and the "Pedagogy Component". The "Domain Model" encodes the expertise about the subject matter that the textbook author provides. The "Learner Model" contains the information necessary to customize the content to the specific user. There are two distinct sub-components of the "Learner Model": "Current Knowledge" and "Learning Preferences". The sub-components encode the learner's current knowledge about the subject matter, and preferences for different styles of content.

Both sub-components require information to be gathered from interaction with the learner. The “Pedagogy Component” describes the various techniques that are available to the system for presenting material. The work to understand and implement each component will be discussed individually, followed by a plan for evaluating the system in its entirety.

Domain Model:

The purpose of the domain model is to encode all the topic specific information that is useful for the content algorithm. As in a traditional textbook, that information is provided by an expert in the subject matter. Our research is focused on deciding what kind of information is needed, and how to encode that information so that it is most useful to the content algorithm. As mentioned in the prior work section, the starting points for this system’s domain model are: a "knowledge graph" made up of (entity, relationship, entity) triples to encode the information about the specific sub-topics, and a directed graph which encodes the relationships among sub-topics. The proposed research on the domain model focuses on two goals: enriching the information stored for each sub-topic, and making information from a sub-topic easy to identify from within a related sub-topic.

One idea for increasing the richness of the domain model implementation for this system comes from a paper [11] seeking to understand how to best explain concepts in general. That paper investigated how to present information to employees which describes the systems that they work with so that the employees can be trained most efficiently. While that work is not specifically focused on creating a representation for algorithmic use, it did inspire an idea that can be used to improve this system's domain model. The models in that paper are entity relationship models similar to the what was discussed in the prior work section, and those models are constructed based on a narrative provided by someone who understands the system. During the construction of the model from the narrative, facts are easily translated into the entity relationship form. Other information in the narrative is lost since that information does not have an obvious representation in the model. The lost information includes opinions, emphasis, and the nuances of language used by the author of the narrative. This information can be useful for this system because, unlike the systems which focus on answering relatively simple questions, textbook style content usually includes additional color and motivation for why the topic is worth studying. One option for making that information available to the content algorithm is to have the author include a narrative in the domain model along with the graph of triples. This direct inclusion of the narrative makes it possible for the system to perform an NLP analysis of the text when it is generating custom content. An alternative is to experiment with methods of encoding the individual opinions and emphasis explicitly. The more flexible NLP analysis seems to be a more suitable approach than an attempt at identifying and encoding all such information, but both options can be attempted.

An additional feature that could be added to enrich the model is example situations. Examples are commonly found in textbooks and are useful for providing explanations. Examples can be encoded in the domain model implementation as a set of entities which are labeled as a single example. The individual entities in the example can each have a relationship which identifies the more general entity to which it is an instance of. When the content algorithm wants to provide an example it can look for these "is instance" relationships to find the set of entities that form the full example.

In terms of making information more easily available between sub-topics, the directed graph of sub-topics can be implemented directly within the main "knowledge graph". All of the entities for all sub-topics are part of one graph, and the directed sub-topic graph for a given topic is identified by a particular relationship. Using the data structures course outline described in the prior work section as an example, that relationship could be "is prerequisite for in data structures". By following those "prerequisite" relationships, a short search can be done to find related information. More explicit references can be added to the main graph so that searching the perquisite graph is only used as a fallback method. These explicit reference relations can be added by the domain expert and can include relationships such as "is based on" and "is similar to". These relationships can take the place of explicit references to chapters and sub-sections that are found in existing textbooks.

Measuring Learning:

As part of the learner model, the system can determine information about the user to generate customized textbook content. The first part of the learner model involves measuring the user's knowledge. This is because the system will need to determine whether the textbook content it has generated is effective in helping the user learn about the concepts.

The way our system can quickly measure one's understanding of a concept is by using mini-quizzes. For a given concept, the system first generates a list of learning objectives as well as a corresponding rubric from the domain model. Alternatively, these objectives can be input from some predefined source, whether they were included in previous versions of textbooks or commonly available on the web. Once these objectives are in place, the system can use NLP to identify the meaning of each learning objective and in turn, generate a question. These quizzes need to be small and simple such that there will be little overhead in measuring knowledge. Moreover, since a chapter could have multiple concepts and in turn, quizzes, the student will not lose motivation as quickly by taking a small assessment per concept. Therefore, the system will generate a multiple-choice, short answer or fill-in-the-blank type questions for these mini-quizzes.

After the student takes the quiz, the system proceeds to grade the questions. Instead of considering each question with equal weight, the system uses the rubric it generated with the learning objectives to compute the score. Some principles are more important for a concept and therefore, should be weighted more in the quiz. Finally, using the results of the quiz, the system can accordingly adjust the content it should generate for the textbook. If the student scored well on some parts of the quiz, the corresponding sections can be trimmed down. On the other hand, the system can identify the learner's knowledge gap from the low scores and rebuff the associated content. By going through the reworked section, the student can gather the material more efficiently the second time around.

Learning Preferences:

The purpose of the learning preferences module is to, quite obviously, learn the preferences of each user such that we can best generate text which will allow them to learn the most from. To begin, we determined one of the key features that we would need to investigate is how we can break up different teaching styles. Upon some investigation, we came to a conclusion that many teaching styles can be categorized under the following categories: discovery, example, scaffolding, explicit explanation, or feedback. The Discovery style is when the teacher guides the student to discover the new knowledge using only their past knowledge. The Example style is, of course, when the teacher performs an example to demonstrate the new knowledge. The scaffolding style is when the teacher solves the problem at hand, and then goes back and explains why they performed each step. The explicit explanation style is when the teacher gives direct instruction, such as reading the definition of the new knowledge. The feedback style is when the teacher has each student perform some task relating to the new knowledge, upon which they then give feedback.

Another key feature of the learning preferences module is determining how we can collect feedback from the user, as well as how we can create a model to accurately represent that feedback without overfitting. To do so, we felt that it would be important to allow the user to provide three variations of feedback: positive, negative, or neutral. In order to accomplish this, a system needed to be devised that would enable the user to provide easy feedback to the system without being too intrusive to the learning process. As such, we felt it best to go with the standard thumbs up or thumbs down system, where neither would represent a neutral opinion on a teaching style. Past this, we also thought it evident that the content of each teaching style was just as important to catch user preferences for as the teaching structure itself was. Because of this, we thought it important to allow the user to apply this feedback also the words that make up each teaching style. This allows our system to act very modular, as each teaching style may have unique preferences, such as words that relate to the user's hobbies being more prevalent in examples, where more scientific terms may be more useful in the explicit explanation teaching style. To leave feedback on a teaching style or word within one, all the user must do is highlight the segment they desire to leave feedback on, as they would in most word processing software, and determine what feedback they wish to leave.

Locking into any particular machine learning strategy at this point would be foolish, as without experimentation it is exceedingly difficult to state with confidence what would perform best. That said, it seems clear that a good place to start would be a statistics-based approach such as reinforcement learning, as they contain many of the characteristics of the systems that we desire for a system like ours. For one, a statistics-based system offers us the capability of being extremely malleable as we are able to impact the learning rate of our system. Another important feature is that statistics-based approaches enable for rapid testing of feedback weights. Both of these attributes are desired traits, as the prior research on learning preference systems indicates that these may be key aspects to creating a successful system.

Pedagogy Component:

The pedagogy component contains several different models of how knowledge and skills are exchanged in an educational context, considering the interactions that take place during learning. These individual pedagogy models vary greatly, as they reflect the different social, political, cultural contexts from which they emerge, and which may apply to the specific learner in different amounts – for example, decisions regarding the curriculum, disciplinary practices, student testing, he language or dialect used by the teacher, and more can empower or disempower students, facilitating or hindering learning on an individual basis. These decisions affect the individual by transmitting norms, values, and beliefs that are not explicit parts of the curriculum, but rather are side effects of the method of transmission. Some may be beneficial, though others might contrast with the background of the individual, creating unnecessary conflict which would slow the learning process.

Determining which approach or approaches to use to most efficiently teach the learner relies as well on information stored in the learner model. Without an understanding of the specific learner for whom material is being created, it is unlikely that a randomly selected set of pedagogical traits would be optimal. However, at the beginning of the interactions between our system and an individual, a statistical method may be used to determine a "predictive best-fit" model. As the interactions between the system and the learner continue to develop, the learner model develops, and the learnings stored there begin to influence the model that the pedagogy component provides, selecting a model that would allow the system to better instruct the learner than the predictive model would; the pedagogy component's supplied model is heavily influenced by the learning preferences module. Logic for the selection of the pedagogy model based on the learner model lives in the content generation module.

Content Generation:
The content generation component is the core of our proposed system. It utilizes the domain models, the pedagogy models, and the learner model to produce the educational content to be shown to the learner. It does so by combining the specific root information exposed by the domain model with the learner model to determine the subset of information that the learner model indicated that the learner would need to spend additional time mastering. Of course, because mastery is not a binary value, the output here would provide a measure of the emphasis needed on different sections. The more accurate the learner model becomes, the more exactly the information derived here will fit the gaps in the learner's understanding.

Once the specific information to be shown is determined, the content generation algorithm would then select an individual pedagogy model, which would determine how to best present that information to the learner. The specific model chosen for this process will have been selected from the pool, as briefly mentioned above, by using the learner model and a statistical machine learning method to best-guess the most effective approach. While a deeper investigation of which specific algorithms should be used for this process is required, they would likely include natural language processing on initial interactions between the learner and the system to categorize the learner into the pedagogy model pool: NLP clues about the social, political, and cultural contexts that the learner occupies would be used to indicate the relative values of different pedagogy models that can be used here. Further interactions would allow this categorization to be repeated, to ensure that the best possible model is selected.

After selection of a pedagogy model completes, it is applied to the information package generated in the first step, creating a user-specific educational package, which is provided to the reading interface, and to the learner. This application involves a translation between the native format of the content stored in the domain model into the ideal format indicated by the pedagogy model. This process would be completed in a similar way as automated language translation today; with large corpuses of information stored in the pedagogy component, contributed to by individual learners over time, and unsupervised deep learning methods to match content to individuals and to cluster that information into the models themselves.

Evaluation:
We believe that our system will be successful as it combines a text generation system with a mindset where both the content and the user of the system are equally important. There are many systems that implement these ideas separately, such as text generators for a particular subject, or recommendation systems that mould to the feedback of their users. We believe standard text to be unsuccessful as it fails to recognize that each user may have different preferences that the system never accounts for, and as such the system generates text that may be meaningless to the user while still being recognizable text to others. We feel that our system successfully incorporates a recommendation system, a method that has been largely successful in creating personalized content, into this word of text. By doing so, we believe that users will better be able to understand the content that the text describes, in a way that writing a single block of text for all users cannot possibly achieve.

In order to test our system, we find it important to differentiate between different phases of testing that our system will undergo. First, it is important to test how different models that we create compare against each other in varying situations. Second, it is important to test our implementation with standard approach to determine if there has been any improvement.

Prior to testing our chosen approach against standard approaches to text learning, we must first be confident in our model. As such, creating our model of text generation for different users is going to rely heavily upon an iterative process. Likely the highest level of internal testing we will need to execute is determining whether our grouping of teaching styles is comprehensive. We will also need to test other characteristics of our system such as the machine learning technique that we utilize and any implementation options that we may have for that system. Next, a large amount of effort will need to be exhausted fine tuning the variables for our system to ensure that our system meets all of the different characteristics we believe a successful personalized text generation system must achieve. All of this must be done to determine whether we have a model that we believe will hold up to a rigorous competition against standard approaches.

The next phase of testing must be to determine if our model has successfully reached better performance than that of standard approaches to text learning. In order to do determine reputable results in this area, we find it best to operate a formal experiment to determine the difference in effectiveness between the two approaches. The operation of such an experiment can best be described as follows. First, it would be most preferable to take a random sample of literate students from all households (home schooled and not), although it seems more likely that we would only be able to operate on a small subset of the real population. From this random sample we would randomly assign each student to either a control group, which would receive standard text, or the experimental group, which would use our system. We feel that it is most appropriate to construct this experiment as a double blind experiment, such that we can eliminate any bias resulting from the method each group is exposed to. To do this, software must be constructed that can take feedback regardless of the group each student belongs to and a computer (not operated by the researchers or participants) must make the group assignments.

When serving content, this system must then choose from a random category to determine the subject of knowledge to be presented, and then using that either present them with static content pulled from pre-written text or text generated from our experimental system. In order to measure the differences between these two strategies, we can require that the user's take a pretest on the topic prior to reading the material presented, and then take a posttest on similar content after having read the material. From this data, we can then determine the net gain in average score for each group and determine if there was any benefit in knowledge recall between the two strategies. In order to properly account for the training of the experimental system, this experiment could be repeated many times such that we can later measure important information such as how quickly it converges to maximum gain between the two strategies. From these different tests, we hope that we can create a successful system that we can state without bias improves learning from text content.

Intellectual merits

As mentioned in the summary, there are several benefits to this research. By incorporating a knowledge graph, a learner model, and a series of pedagogy models, the combination of content selection capabilities with targeted presentation styles should enhance the end result beyond that of either technique used alone. While running the content generation algorithm, multiple different content formats can be derived that represent different pedagogies; between this and the learner model, the content algorithm can narrow down which specific values in the pedagogy models best fit the learner, refining the content generation further, and thus increasing the efficiency of the overall system, and allowing individuals to be taught quicker with less human intervention. The data gathered regarding which pedagogy models yield the best results can be used both to feed back into the set of pedagogy models in a standard genetic algorithm (or similar) to create more and more generally useful pedagogy models (if possible), and to provide deeper insights into the ways that humans learn most effectively.

The language translation capability required by the content generation algorithm when incorporating the pedagogy models that deals with formats and presentation styles, rather than strict definition of words and phrases, would advance current frontiers of understanding the human learning process, as well as a fuller understanding of human language.

Our domain model would continue to improve upon knowledge graph generation, implementing features of several modern systems. By utilizing the best features of each of these systems, it is likely that the end result created here would be more capable and more general than the modern approaches. As mentioned above, we intend to take best practices and learnings from Bera et. al., Brusilovsky et. al., Stewart et. al. We hope that our efforts in this field will be expanded upon in the future, possibly leading to a reinforcement learning model that learns from our successes, which may use human input to train and focus the ontology generated into a more accurate system overall.

As the learner models become more and more powerful as the feedback system continues to run, it is possible that the model will become accurate enough to experiment with. A pool of realistic learner models can be used, without a true human learner involved whatsoever, to derive different presentation approaches that may be more applicable to larger sets of the learner population. This method would enhance the results obtained via the pedagogy back-propagation algorithm mentioned in the first paragraph in this section, providing a potentially exponential speedup, as the real-time constraint of human interaction would no longer apply. At this point, rapid prototyping of presentation types and formats could be achieved, with realistic expectations of how well they would work for human learners. A playground could be provided for human experts as well; imagine a wave of educators able to try bold new strategies on a pool of simulated students from different backgrounds and cultures without concern for failure, and imagine the best of these new strategies implemented in our self-improving system.

Broader impact

If the process of human learning can be improved, and education enhanced, this has tremendous benefits, both in the short and in the long term for humanity as a whole. By receiving individualized content, a user obtains greater ease of learning, allowing students to more efficiently use their time. This would create the possibility of learning faster, enabling students to cover more material in a shorter period of time. Students with extra time can then spend that time in a variety of ways, such as in social groups, or drilling down into material that interests them, either with our system or with more human-centric personalized education. Without the need to teach a classroom full of students the same material, teachers and instructors can spend focused attention and time on those students who are truly interested in what they have to teach. Certainly, it is clear that reducing the amount of time and effort needed to spend in a process that every single person undergoes has tremendous potential.

The high school dropout rate is heavily influenced by failing grades and boredom [12]. One of the causes of this effect is the limited number of teachers available per student; it is well-documented that student engagement is inversely correlated to the class size [13]. With personalized content generation allowing for more accessible learning, student engagement (and therefore graduation rates) can only increase. Those interested in the material can dig deeper, and those who would be pushed away by a presentation that they struggle to comprehend would remain.

With even a 5% lower high school dropout rate, the estimated total financial benefit would be over $50B/year. These benefits consider the decrease in crime rates and jail costs (high school dropouts are more likely to commit murder, assault, robberies, and other crimes), the decrease in medical costs through Medicare (dropouts are generally less healthy, require more medical care, and die earlier), and benefits to the economy via an influx of more skilled labor ("increasing the national high school graduation rate to 90 percent for just one high school class would create as many as 65,700 new jobs and boost the national economy by as much as $10.9 billion"). "The time to act is now. In an increasingly global economy, American high school students must achieve at increasingly higher levels to allow the country to maintain its competitive advantage. Ensuring that all secondary students are prepared to succeed in college and work is a giant step in the right direction and will benefit individuals and society for decades to come" [14]. There are an incredible number of other sources with a similar message - even improving just high school education is in fact a silver bullet for a wide range of societal problems, and any improvement will have lasting effects on the students, and in the world.

In the field of continuing education, in recent years there has been a huge influx of massive, online, open, courses. These courses present the same material in the same way to every user, and individual tutoring or assistance is currently rare. With a system that could generate this individualized content, the efficacy of these online courses could grow to rival that of traditional institutions. An Ivy League education could be made available to the entire world, without the highly exclusive applications process made requisite by the limitations of today's professors.

References

1. Luckin, R., Holmes, W., Griffiths, M. & Forcier, L. B. (2016). Intelligence Unleashed. An argument for AI in Education. London: Pearson.
2. Brusilovsky, Peter. & Schwarz, Elmar. & Weber, Gerhard. (1996). *A Tool for Developing Adaptive Electronic Textbooks on WWW*. [Washington D.C.] : Distributed by ERIC Clearinghouse, https://eric.ed.gov/?id=ED427653
3. Stewart, C., Cristea, Alexandra I., Brailsford, T. and Ashman, H. (2005) *'Authoring once, delivering many' : creating reusable adaptive courseware.* In: 4th IASTED International Conference on Web-Based Education (WBE 2005), Grindelwald, Switzerland, 21-23 Feb 2005 pp. 21-23.
4. El-Bakry, Hazem & A. Saleh, Ahmed & T. Asfour, Taghreed & Mastorakis, Nikos. (2011). Adaptive E-Learning Based on Learner's Styles. Bulletin of Electrical Engineering and Informatics. 2. 440-448. 10.12928/eei.v2i4.189.
5. A. Garcia-Duran, A. Bordes, N. Usunier, Y. Grandvalet, Combining two and three-way embedding models for link prediction in knowledge bases, J. Artif. Intell. Res. 55 (2016) 715–742.
6. Nurjanah, D., Fiqri, M. "Graph-based Domain Model for Adaptive Learning Path Recommendation", 2017 IEEE Global Engineering Education Conference (EDUCON), p. 375
7. Lovett, S., & Johnson, J. (2012). Measuring learning through cross sectional testing. *Journal of the Scholarship of Teaching and Learning*, *12*(4), 43-57.
8. Stefl-Mabry, J. (2004). Building rubrics into powerful learning tools. Knowledge Quest, 32(5), 21-25.
9. Newhart, D. (2015). To Learn More about Learning: The Value-Added Role of Qualitative Approaches to Assessment. Research & Practice in Assessment, 10, 5-11.
10. Carmona, Cristina, Gladys Castillo, and Eva Millán. "Discovering student preferences in e-learning." Proceedings of the international workshop on applying data mining in e-learning. 2007.
11. Bera, P., Burton-Jones, A., and Wand, Y. 2011. "Guidelines for Designing Visual Ontologies to Support Knowledge Identification," MIS Quarterly (35:4), pp. 883-908.
12. Doll, J. J., Eslami, Z., & Walters, L. (2013). "Understanding why students drop out of high school, according to their own reports." SAGE Open, 3(4), 2158244013503834. https://doi.org/10.1177/2158244013503834.

13. Blatchford, P., Bassett, P., & Brown, P. (2011). "Examining the effect of class size on classroom engagement and teacher–pupil interaction." Learning and Instruction, Volume 21, Issue 6, December 2011, Pages 715-730.
14. "The High Cost of High School Dropouts: The Economic Case for Reducing the High School Dropout Rate," Alliance for Excellent Education, https://all4ed.org/take-action/action-academy/the-economic-case-for-reducing-the-high-school-dropout-rate/