CHAPTER 9 – GENERATIVE AI FOR CONCEPT LEARNING: EVOLUTION OF THE MOBILE FACT AND CONCEPT TRAINING SYSTEM

Andrew M. Olney
The University of Memphis

Introduction

Intelligent tutoring systems, like all adaptive training systems, require three essential elements to deliver personalized instruction: a model of student performance, training material appropriate to that level of performance, and a method that can select the best training material at any moment. Models of student performance and the selection of optimal training material are interesting and challenging research problems, but they are well-bounded from a practical standpoint. For example, consider an algebra tutoring system for word problems (Singley et al., 1989). The student model can be constructed using ACT-R (Adaptive Control of Thought – Rational), and an analysis of knowledge components and knowledge tracing can be used to select the next optimal problem (Corbett & Anderson, 1994). However, the word problems themselves, i.e. the text which situates the problem in a real-world context, is relatively unbounded, simply because there are so many possible real-world contexts. This problem of creating many different permutations of training content must be overcome in order to fully adapt to learner preferences, interests, and prior knowledge (Sinatra, 2015): when a learner is given an algebra word problem on a familiar topic, it can increase their motivation to learn and activate prior knowledge that enhances learning performance.

For many years, the research community has focused on training material authoring tools, some of which include a high level of automation (Olney et al., 2015; Olney et al., 2021). Traditional authoring tools that generated text typically used a template-based system that required progressively deeper symbolic natural language processing (Gatt & Reiter, 2009; Gatt & Krahmer, 2018) including lexicons, knowledge bases, and other resources that in practice are extremely difficult and time consuming to construct. For example, the Cyc project was one of the longest-running knowledge base construction projects but has been widely criticized for not being usable or useful (Conesa et al., 2010; Domingos, 2015). In this earlier period of artificial intelligence (AI) research, knowledge base construction was a key step for curating the knowledge used to automatically generate training content.

Transformer-based large language models (LLMs; Vaswani et al., 2017) have dramatically changed the landscape of natural language generation (NLG) and thus the generation of training content. In contrast to traditional NLG approaches that separate knowledge from generation, LLMs typically consist of a single stage that generates text using knowledge implicitly represented in the weights of the network. This is particularly impressive given the common training objective of LLMs is to predict the next token (i.e. word) in text rather than to acquire knowledge per se. Even relatively early LLMs have been shown to be competitive with or outperform traditional knowledge bases for querying factual knowledge (Petroni et al., 2019), and the ability of LLMs to recall factual knowledge has been shown to increase with size (Roberts et al., 2020). The rapid progress in LLMs meant that by 2020, it was possible to use LLMs to create content for adaptive training systems.

This chapter reflects on the rapid advances of LLMs during the development of an adaptive training system and how these advances changed our development plans and extended the system beyond our original plans. The Mobile Fact and Concept Training System (MoFaCTS; Pavlik et al., 2020; Pavlik & Eglington, 2021),

is an adaptive training system designed for learning simple content (i.e. facts), that can be understood as a flashcard system with an enhanced student model. For a given learning task, MoFaCTS presents a set of items (flashcards) to the learner, and the learner must enter (type or speak) the correct answer for that item. If the learner is correct, MoFaCTS proceeds to the next item. If the learner is incorrect, MoFaCTS notifies the learner that the answer is incorrect and presents the correct answer for a set period of time (i.e. a forced study period). To select the next item, MoFaCTS uses a generalized model of the difficulty of each item but updates that model based on a learner's history of practice together with a cognitive model of forgetting. MoFaCTS estimates the probability that a learner will answer an item correctly based on the number of previous successful and unsuccessful trials and the time elapsed since the last trial. MoFaCTS then selects the item that is close to the threshold of being forgotten, based on an efficiency principle: practice with almost forgotten items takes less time than practice with forgotten items, because forgotten items trigger a forced study period. The focus of our project was to automatically generate items for MoFaCTS from a textbook for Anatomy and Physiology (A&P), a key course for nursing and allied health professions that requires a great deal of memorization.

Our original research plan was to use traditional statistical and symbolic methods to extract cloze items from the text, generate refutational feedback in response to incorrect answers, and to create paraphrases. To accomplish these tasks, we planned to leverage previous work on cloze item generation from textbooks (Olney et al., 2017), concept map extraction from text followed by NLG (Olney et al., 2010, 2012; Olney, 2018), and machine-translation techniques for paraphrase (Napoles et al., 2016). Ultimately, we only used our previous work on cloze item generation and accomplished everything else with LLMs – because LLMs allowed us to accomplish the same objectives in less time. Additionally, we were able to expand our project scope to include multiple-choice question generation, definition generation, and syllabification. The remainder of this chapter describes in detail each of these decisions, the resulting systems, and makes recommendations for future adaptive systems, including the Generalized Intelligent Framework for Tutoring (GIFT).

The Role of Generative AI

Elaborated feedback

The purpose of elaborated feedback is to remediate a student's incorrect answer in a more useful way than just providing the correct answer. Consider the following example from our data, where underlining indicates the blank in the cloze item:

Source: The <u>cytoplasm</u> at the distal end of the axon is rich in mitochondria and contains many tiny vesicles that store neurotransmitter.

Student: acetylcholine

In this example, the student misunderstands cytoplasm because they made an error. However, it is likely that they also misunderstand acetylcholine and perhaps even the relationship between cytoplasm and acetylcholine. Therefore, our elaborated feedback should address the target concept, the incorrect answer concept, and their relationship (if it exists).

Our initial approach to this problem was based on our previous work on extracting concept maps from text (Olney et al., 2010, 2012; Olney, 2018). In that approach, the relationship between two concepts can be uncovered by finding the shorted path between the concepts in the map. Let us look at a simpler example than the cytoplasm/acetylcholine one above. Suppose that we extracted the following triples from the textbook:

sugar – has_type – monosaccharide

monosaccharide – has_type – fructose

sugar – has_type – disaccharide

disaccharide – has_type – sucrose

disaccharide – has_property– two joined monosaccharides

Suppose a student has confused fructose and sucrose and so answered a cloze item with sucrose when the correct answer is fructose. Using the above triples, an NLG system can create a paragraph explanation, e.g. "Sucrose and fructose are both sugars, but sucrose is a disaccharide and fructose is a monosaccharide. A disaccharide is characterized by two joined monosaccharides." For this kind of approach to work, the triples need to be extracted from the textbook without error, the dialogue planner that selects relevant triples needs to do so without error, and the triples need to be rendered to text using NLG without error.

One of the major obstacles we faced in applying our previous approach was that it relied on a variety of components that were old, idiosyncratic, and difficult to port to the web. For example, the Lunds Tekniska Högskola (LTH) semantic parser (Johansson & Nugues, 2008), a key component in our earlier work, implements a different theory of syntax than nearly all parsers from the last decade, which mostly use Stanford dependencies. Simply stated, Stanford dependencies are surface-syntax oriented, whereas the LTH parser is deep-syntax (meaning) oriented. Further, the LTH parser provides word sense disambiguation of predicates to specific PropBank frames, i.e. disambiguates word senses to specific patterns of semantic argument structures. Virtually no semantic parsers in the last decade attempt this disambiguation, since this is not a key component of shared task evaluation. Thus, modern parsers were not delivering the information needed for our knowledge extraction system to operate.

To implement our previous approach, we were progressively investing more time in "reinventing the wheel" rather than making progress on elaborated feedback in MoFaCTS. Thus, we decided to investigate recent advances in deep learning that focus on question answering. To leverage that work, we reformulated the problem of providing feedback as the problem of answering a student's implicit question, i.e., the question they should ask once they realize their error. In the first example, this might be, "What is the relationship between acetylcholine and cytoplasm?" To explore this idea, we developed a deep learning question answering system based on recent work in long-form question answering, which returns a paragraph explanation to a question rather than a single word or phrase answer. Importantly, we leveraged a model developed using the "Explain Like I'm Five" (ELI5) forum on Reddit (www.reddit.com/r/explainlikeimfive/). Because this forum attempts to answer questions in the simplest terms possible, it provides a compelling dataset for building models with the same properties. Here is the answer to the relationship question in our running example, from the current version of our system, where the first two sentences are MoFaCTS standard feedback, followed by the underlined elaborated feedback:

Acetylcholine is not right. The correct answer is cytoplasm. <u>Acetylcholine is synthesized in the cytoplasm of nerve terminals by the enzyme choline acetyltransferase and is then transported into synaptic vesicles.</u>

In our best performing system (Olney, 2021), we used an early retrieval-augmented generation approach where we provided definitions for the key terms (cytoplasm and acetylcholine) and passages from the A&P textbook used in the course to the question answering model. Definitions were obtained from the textbook glossary, or if missing, from Wikipedia by first applying a wikifier (Cheng & Roth, 2013) to the correctly filled-in cloze item sentence in order to get Wikipedia page ids for the key terms in the sentence. These page ids were then used to query the corresponding Wikipedia pages for their first paragraph of text, which

was used as a proxy for a definition. Additional supporting documents were retrieved using an Elasticsearch (https://www.elastic.co/elasticsearch/) index of the A&P textbook with the synthetic question as the query.

In contrast to the previously proposed concept map approach, in the neural question answering approach we never parsed the textbook or created a knowledge representation for further processing. We did not even retrain the question answering model and simply used the stock model available from HuggingFace for ELI5 (https://yjernite.github.io/lfqa.html). Our evaluation of the system's performance used nurses and similar experts and screened out ineffective/unreliable raters using intra- and inter-rater reliability metrics (see Olney,2021 for details). The results suggest that both the fluency and correctness of answers are quite high; comparing to our other evaluations which use these metrics, at this time we speculate that the system performance is close to human in both fluency and correctness, though in our evaluation we did not directly compare to human elaborated feedback. Similarly, we did not compare the system performance to our planned symbolic approach, but compared to our earlier findings (Olney et al., 2012), a casual observer can see that performance is dramatically better. In short, the performance and development speed benefits of LLMs (relative to traditional symbolic approaches) were quite striking based on our previous experience.

Reformulating elaborated feedback as an answer to a question the student did not ask (but should have), brings elaborated feedback into parity with tutorial dialogue in the AutoTutor framework (Nye et al., 2014). Simply stated, all AutoTutor sessions for learning content (rather than adult literacy) are based on a seed question and ideal answer. The rest of the dialogue is then derived from the ideal answer with the goal of eliciting that ideal answer from the student. Therefore, all hints, prompts, and related dialogue are simply transformations of sentences that comprise the ideal answer into tutor questions and assertions. Because of the symmetry between ideal answers and elaborated feedback, we can use the elaborated feedback paragraphs generated by our long form question answering model as the basis for a tutorial dialogue session that remediates the student's incorrect answer. We additionally implemented this tutorial dialogue in MoFaCTS using traditional symbolic methods but with elaborated feedback as the basis of the dialogue content.

Paraphrases

The purpose of paraphrasing is to force the student to deeply encode the knowledge in the item rather than superficial features of the item. Anyone who has used paper flashcards has likely experienced the tendency towards superficial encoding, for example remembering the answer based on a stain or bend on the card rather than a knowledge-based association. This same kind of processing happens with computer-based items where participants tend to read the first few words of the item and remember the answer rather than remember the answer based on knowledge by reading the entire item and reasoning to the answer.

Paraphrasing changes the words used in the item to remove superficial cues while preserving the meaning of the text. Paraphrasing makes relying on superficial cues less effective and, in theory, promotes deeper processing. Cloze items based on paraphrases of text from the A&P textbook should thus have these beneficial properties.

Our proposed approach to paraphrasing was to use statistical methods for paraphrasing described in Napoles et al. (2016), which frames paraphrasing as a machine translation problem. Essentially any transformation of text that approximately preserves meaning can be framed as a machine translation problem with a specific objective, e.g. both shortening text and changing the reading level of text involve changing the surface form of text with an additional objective. The method of Napoles et al. (2016) is based on first obtaining snippets of English that have been translated to other languages and then back to English (with an associated change in surface form) and then using statistical machine translation techniques to search through the possible space of snippets and assemble those that best minimize an error function for the paraphrase task. The

classic problem with systems like this is that there can be errors in the translation data, errors associated with the snippet boundaries, and errors associated with the search process.

Building on recent advances in LLMs, we instead developed a paraphrase method based on backtranslation of the A&P textbook (Olney, 2021). Backtranslation is the process of translating from English into another language and then back into English. The "other" language is called a pivot language, and more than one pivot language can be used in a backtranslation process, e.g. English – Czech – Russian – English. Leveraging the wide variety of pivot languages that have been explored by other researchers, we evaluated Czech, Russian, Chinese, Persian, Arabic, Hindi, Turkish, and Welsh as pivot languages and found that Czech and Russian introduced the most lexical/syntactic variety while introducing the fewest meaning errors. We further evaluated combining Czech and Russian into a double pivot, as in the previous example, and found this increased lexical/syntactic variety over each pivot individually without noticeably increasing meaning errors. We then backtranslated the textbook using the Google Translate API (Application Programming Interface), which is a high-quality neural machine translation, and used that backtranslation data to train a text-to-text paraphrase model using the T5 (Text-to-Text Transfer Transformer) LLM (Raffel et al., 2020). Our training data consisted of sentence pairs (original and backtranslated) such that T5 was trained to produce the backtranslated sentence when the original sentence was input, and vice versa. An example source sentence and system paraphrase are given below, with the changed text underlined:

Source: Another layer of connective tissue, called the perimysium, extends inward from the epimysium and separates the muscle tissue into small sections.

Paraphrase: Another layer of connective tissue, called the perimysium, extends inward from the epimysium, dividing the muscle tissue into small portions.

An expert evaluation, again with nurses and doctors as participants, found high fluency for the paraphrases, with approximately 40% of paraphrases being judged as more fluent than the original text. Meaning was also rated highly, with 70% of all meaning ratings above 75 on a 100-point scale. We did not directly compare human generated paraphrases in the evaluation, but based on later work with similar evaluations, we suspect that the performance of the system is close to human in most cases for both fluency and meaning. Similarly we did not directly compare the performance of the T5-based system to the system of Napoles et al. (2016) for A&P paraphrases, so we can only speculate as to the difference in paraphrase quality by pointing to the general finding that joint inference reduces error more than a pipeline approach (cf. Finkel et al., 2006) and that a text-to-text neural approach embodies joint inference in this way, even down to the features themselves. Creating the training data and training T5 to perform this paraphrase task took less than a week of effort, which created time for additional research outside the scope of our grant proposal.

Multiple Choice Questions

Multiple choice questions (MCQs) can be used both for learning and for assessing learning. As items presented in MoFaCTS, they could reduce the difficulty of a cloze item by giving a hint, since the learner only needs to recognize the correct answer among the options. In the same way, MCQs could reinforce differentiation between confusable concepts listed among the answer options. When used for learning, MCQs could be paired with immediate feedback, like the elaborated feedback previously discussed.

We explored MCQ generation with LLMs in two papers; in what follows we describe the initial system development (Olney, 2022) and most recent human evaluation (Olney, 2023). Similar to our approach to elaborated feedback, our approach to MCQ with LLMs was based in question answering, specifically MCQ answering, inspired by a system called Macaw (Tafjord & Clark, 2021). We used Macaw, which is based on the T5 model, for question generation, as it leverages multiple questions-answering datasets by casting them into a common format based on slots, including A (answer), Q (question), M (multiple choice options),

and C (context), and input/output specifications called angles. For example, a simple dataset that pairs answers with questions could support the angles Q→ A (for question answering) as well as A → Q (for question generation). Datasets with more slots can support more angle combinations. By putting datasets into this common format, it was possible to train Macaw on a variety of question-oriented tasks, including question generation. However, the original evaluation focused on question answering, so it was unclear how effective Macaw would be for MCQ generation.

To use Macaw successfully, we first evaluated what angles yielded the best questions. We evaluated three angles with increasing specificity, C → QMA (which generates question, options, and answer from text alone), AC → QM (which generates question and options from text and answer), and QAC → M (which generates only the options given the other inputs). Our expectation was that providing the most information, which simplifies the task, would give the best results, but paradoxically it was the intermediate angle AC → QM that yielded the best results: it gave four distinct options 83% of the time compared to QAC → M which yielded four distinct options 67% of the time and C → QMA, which never yielded four distinct options. In order to increase the effectiveness of AC → QM, we leveraged the inherent instability of LLMs where they provide quite different outputs based on minor changes to the input by using the paraphrase system previously discussed to generate new angle inputs whenever Macaw failed to generate four distinct outputs. We repeated paraphrase generation up to 10 times (an arbitrary limit that could have been extended) and increased the success of AC → QM to 97.5% of items.

We conducted a human evaluation study using nurses and doctors as participants with questions from three sources: an A&P textbook from OpenStax (Betts et al., 2017; Textbook), our application of Macaw (paraphrase-augmented AC → QM; hereafter Macaw+), and BingChat, a GPT-4 service that was later rebranded at Copilot. All MCQs from OpenStax were rewritten as sentences with answers, and these sentences/answers were used as input to both Macaw+ and BingChat. For BingChat, they were included in the following prompt:

Write a multiple choice question using the following sentence and answer. Convert the sentence into a question that matches the answer. Use JSON format.

Sentence: <sentence>

Answer: <answer>

Note the above approach closely matches the content of questions across the conditions; in theory each condition will have a semantically identical question and the exact same answer and vary most in their answer options. Participants were asked to rate questions on seven different scales, including question (correctness and fluency), correct answer (truly correct and present in options), answer options (number of options that are correct and number of options that are distinct) and combined quality. Results from the analysis indicated that there were only significant differences between the conditions for the 'answer present in options' and the combined quality scores. There were no significant pairwise differences for the former, but combined quality post hoc analysis indicated that the Textbook condition had greater combined quality than Macaw+ and BingChat, which were not different from each other. Follow up error analyses suggested that the primary failure mode of Macaw+ was to fail to generate four distinct answer options, especially when the options were in some kind of list. In contrast, the primary failure mode of BingChat was to fail to generate a question with a given correct answer. Instead, it would sometimes generate a question whose correct answer was not the given answer. Additional analyses showed that when questions with incorrect distinct options are eliminated, Macaw+ was not significantly different from Textbook on combined quality, but even using the same approach with 'answer present in options,' BingChat was still significantly worse than Textbook on combined quality. Altogether this suggests the best solution is to take the simple approach of checking for distinct options with Macaw+ and discarding questions without four

distinct options, both of which are technically simple to accomplish. With this simple check, Macaw+ questions are likely to be indistinguishable from human-authored questions in the A&P domain, and we suspect in all domains, since Macaw was not specifically trained on A&P.

Morphology generation

Learners struggle with vocabulary in all domains, but vocabulary can be particularly challenging in biology domains, where words are often neoclassical compounds using Greek and Latin morphemes. Traditional vocabulary instruction often teaches learners to recognize morphemes in order to learn vocabulary. For example, the word "abduction" which means "to move away/apart" can be recognized as ab+duct+ion or "away lead process" while related English words "duct" and "adduction" can be recognized as an antonym since "ad" means "towards." In terms of cloze items, these morphemes can provide hints as partial completions for the missing word, e.g. "__duction," which is the primary way they were used in MoFaCTS. though one could also design cloze item practice to learn morphemes.

Our approach to this problem was to use Wiktionary as training data for GPT-2 with the task of generating morphemes and their corresponding meanings for a given input word (Yarbro & Olney, 2021b). Wiktionary is a Wikimedia project, similar to Wikipedia. Wiktionary includes dictionary information as well as etymologies and pronunciations. In piloting, it became clear that GPT-2 was much better at performing this task when given a definition for the target word in addition to the word itself. Presumably definitions helped GPT-2 by allowing it to relate the meaning of the morphemes generated to the overall meaning of the word. Our evaluation found that when the definition of the input word was provided, the accuracy of morpheme segmentation for English morphemes was 92.5% and for non-English morphemes was 89%. Similarly the morpheme meaning correctness measured by the ROUGE-L metric was .81 for English morphemes and .95 for non-English morphemes.

Definition generation

Definitions are a core part of learning vocabulary and are complementary to morpheme generation described above. Providing definitions for words in context, on demand, can improve reading comprehension for unknown words and facilitate vocabulary learning.

Using the same approach as described for morpheme generation, we created a definition generation model that takes a word in context and generates a definition (Yarbro & Olney, 2021a). Some example generation outputs are:

Context: You can bank on it

Definition: Have faith or confidence in.

Context: A bank of snow

Definition: A slope, mass, or mound of a particular substance

We conducted an evaluation using four domains (American Government, A&P, Astronomy, and Psychology) that compared human definitions, generated definitions using a sentence of context, and generated definitions using multiple sentences of context. Interestingly, using multiple sentences of context was significantly worse across all domains for definition accuracy than using a single sentence of context, though there was no difference in terms of definition fluency. Human definitions were rated significantly more accurate than short context generated definitions across topics, but interestingly for A&P the difference between human and short context definitions was negligible. This suggests that for some

domains, our generated definitions may be indistinguishable from human definitions, though our generated definitions are clearly worse in some domains, like American Government and Astronomy.

Discussion and Recommendations for Future Research

Our experience during the transition period towards using Transformer LLMs for NLG appears to be broadly representative of the field of adaptive training using language as an instructional medium. Our effort in content generation on the MoFaCTS project shifted from an effort spanning knowledge extraction/engineering and content generation to a single step of content generation. Additionally, the shift from formal representations in content generation to a LLM approach of text-in and text-out significantly increased the speed of system development and the quality of generations. This is a startling change from our previous work (Olney et al., 2010, 2012; Olney, 2018) where the AI-generated text always needed human review to ensure quality/accuracy. LLM-generated content in the MoFaCTS project is typically always suitable for immediate use by students, though complete accuracy is never guaranteed. It is clear that Transformer LLMs have changed the landscape of adaptive training and are poised to become the dominant paradigm in AI for the foreseeable future. Additionally, Transformer LLMs are increasing the participation of researchers in the field by reducing the expertise demands for creating adaptive systems: it is no longer necessary to have a Ph.D. in AI/ML (artificial intelligence/machine learning) to create effective adaptive training systems.

However, with this change in landscape there are new concerns and new opportunities for research. The field continues to be broadly concerned with accuracy of generated output (sometimes referred to as hallucinations) and the general control of such output to avoid undesirable characteristics. Additionally, the output of these systems is now so good that automated evaluation approaches generally are not sensitive enough and human evaluations are needed (van der Lee et al., 2019). The approach to human evaluation we used in the MoFaCTS project (Olney, 2023), which borrowed ideas from evaluation of machine translation, is a good model for future researchers to assess where the systems are working as intended. Additionally, we recommend continuous feedback from learners for any questionable content, such that it can be flagged and reviewed by an instructor or expert. In our experience, it is not adequate to ask instructors to review all content because many will not do it.

Curiously, the current landscape of Generative AI and how to incorporate it into systems is very comparable to working with a human collaborator who is not fully trusted. A human collaborator can manifest factual errors, etc., just like a Transformer LLM. How we deal with this situation is what we should have been doing in our adaptive learning systems all along – undertaking rigorous evaluations of effectiveness and performing ongoing telemetry and improvement to remove weaknesses. Hopefully by reducing demand for development time, Generative AI will increase researcher time to engage in evaluation, field deployment, and continuous improvement of fielded systems.

Recommendations for GIFT and STEEL-R Overall

Our success with LLMs suggests that they are broadly applicable to GIFT and its use in STEEL-R (Synthetic Training Environment Experiential Learning for Readiness) , as well as future intelligent tutoring systems generally. Our work specifically informs GIFT's Pedagogical Model and Domain Model. In the Pedagogical Model, LLMs can provide refutational feedback. This personalized feedback would provide an explanation of the learner's error rather than providing a generic correct answer. This capability could also be more generally applied in the synthetic training environments of STEEL-R. A straightforward application would be refutational feedback of team communications. If there was a misunderstanding or breakdown in communications, an LLM could provide feedback based on the history of communications

and a synthetic question like "Why was erroneous action A undertaken based on this conversational history?" As suggested by this example, STEEL-R would only need qualitative information about raw data in order to have an LLM provide feedback this way, e.g. correct/erroneous. For example, in shooting training, raw data might be coordinates of target hits, but qualitatively these could be described as left/right and high/low. Based on some training history, a feedback-oriented LLM could suggest trigger control or breath control remediation using this qualitative data. The Pedagogical Model could also be supported by our work on definition generation and morpheme generation. When the learning task involves definition-rich doctrine or specialty training, LLMs can provide in-context definitions instantly and break down less familiar words into units/subwords that make them easier to remember. For example, "amphibious" means dual life, i.e. life on both land and water.

In the Domain model, LLMs can create MCQs automatically based on domain content. Automatic generation of questions can both lower assessment costs and speed up development of training materials. LLMs can further be used for a variety of domain authoring tasks like generating introductory text, creating learning objectives, or summarizing/selecting text based on specific objectives and the learner's knowledge state. For STEEL-R, LLMs could be used to develop and illustrate the Knowledge, Skills, Abilities, and Attitudes (KSAAs) being assessed. Additionally, LLMs could be used to design or assist with the design of training scenarios based on learning objectives, including narrative elements, potential challenges, and specific tasks. LLMs have already shown great promise in producing programming code from specifications; to the extent that a synthetic training environment can be defined by code, there is potential for LLMs to create code that implements a given scenario.

Conclusions

One of the great bottlenecks in the development of intelligent tutoring systems has been the creation of training content: adaptive training systems need exponentially more content than static systems in order to provide personalized instruction to a variety of learners. Before LLMs, symbolic AI systems were used to help author intelligent tutoring systems, but these symbolic AI systems were themselves exceedingly difficult and time-consuming to construct.

In this chapter, we explained how Transformer-based LLMs have dramatically changed the status quo for generation of training content. Our project developing the MoFaCTS system for Anatomy & Physiology straddled the historical inflection point of LLMs, leading us to pivot from traditional statistical and symbolic methods to an LLM-dominated approach. Our experience appears to be broadly representative of the field of adaptive training: the shift from formal representations in content generation to a LLM approach of text-in and text-out significantly increased the speed of system development and the quality of generations. This work, and LLMs generally, are broadly applicable to GIFT and STEEL-R, particularly for feedback, content generation, and assessment.

Acknowledgements

This material is based upon work supported by the Institute of Education Sciences under Grant R305A190448 and by the National Science Foundation under Grants 1918751 and 1934745.

References

Betts, J. G., Desaix, P., Johnson, E., Johnson, J. E., Korol, O., Kruse, D., Poe, B., Wise, J. A., Womble, M., & Young, K. A. (2017). *Anatomy and Physiology*. OpenStax.

Cheng, X., & Roth, D. (2013). Relational Inference for Wikification. In D. Yarowsky, T. Baldwin, A. Korhonen, K. Livescu, & S. Bethard (Eds.), *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 1787–1796). Association for Computational Linguistics. https://aclanthology.org/D13-1184

Conesa, J., Storey, V. C., & Sugumaran, V. (2010). Usability of upper level ontologies: The case of ResearchCyc. *Data & Knowledge Engineering*, *69*(4), 343–356. https://doi.org/10.1016/j.datak.2009.08.002

Corbett, A.T., & Anderson, J.R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model User-Adap Inter, 4*, 253–278.https://doi.org/10.1007/BF01099821

Domingos, P. (2015). *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. Basic Books.

Finkel, J. R., Manning, C. D., & Ng, A. Y. (2006). Solving the Problem of Cascading Errors: Approximate Bayesian Inference for Linguistic Annotation Pipelines. In D. Jurafsky & E. Gaussier (Eds.), *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing* (pp. 618–626). Association for Computational Linguistics. https://aclanthology.org/W06-1673

Gatt, A., & Krahmer, E. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Int. Res.*, *61*(1), 65–170.

Gatt, A., & Reiter, E. (2009). SimpleNLG: a realisation engine for practical applications. *ENLG '09: Proceedings of the 12th European Workshop on Natural Language Generation*, 90–93.

Johansson, R., & Nugues, P. (2008). Dependency-based syntactic-semantic analysis with PropBank and NomBank. *CoNLL '08: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, 183–187.

Napoles, C., Callison-Burch, C., & Post, M. (2016). Sentential Paraphrasing as Black-Box Machine Translation. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations* (pp. 62-66). San Diego, California: Association for Computational Linguistics.

Nye, B. D., Graesser, A. C., & Hu, X. (2014). AutoTutor and Family: A Review of 17 Years of Natural Language Tutoring. *International Journal of Artificial Intelligence in Education*, *24*(4), 427–469. https://doi.org/10.1007/s40593-014-0029-5

Olney, A. M. (2018). Using novices to scale up intelligent tutoring systems. In Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2018.

Olney, A. M. (2021). Generating Response-Specific Elaborated Feedback Using Long-Form Neural Question Answering. *Proceedings of the Eighth ACM Conference on Learning @ Scale*, 27–36. https://doi.org/10.1145/3430895.3460131

Olney, A. M. (2021). Paraphrasing Academic Text: A Study of Back-Translating Anatomy and Physiology with Transformers. In I. Roll, D. McNamara, S. Sosnovsky, R. Luckin, & V. Dimitrova (Eds.), *Proceedings of the 22nd International Conference on Artificial Intelligence in Education* (pp. 279–284). Springer International Publishing.

Olney, A. M. (2022). Generating Multiple Choice Questions with a Multi-Angle Question Answering Model. In S. E. Fancsali & V. Rus (Eds.), *Proceedings of the 3rd Workshop of the Learner Data Institute* (pp. 18–23). https://doi.org/10.5281/zenodo.7761561

Olney, A. M. (2023). Generating multiple choice questions from a textbook: LLMs match human performance on most metrics. In S. Moore, J. Stamper, R. Tong, C. Cao, Z. Liu, X. Hu, Y. Lu, J. Liang, H. Khosravi, P. Denny, A. Singh, & C. Brooks (Eds.), *Proceedings of Empowering Education with LLMs—The Next-Gen Interface and Content Generation*. CEUR-WS.org. https://ceur-ws.org/Vol-3487/paper7.pdf

Olney, A. M., Brawner, K., Pavlik, P., & Koedinger, K. R. (2015). Emerging Trends in Automated Authoring. *Design recommendations for adaptive intelligent tutoring systems: Learner modeling, Vol. 3 of Adaptive Tutoring* (pp. 227-242). Orlando: U.S. Army Research Laboratory.

Olney, A. M., Gilbert, S. B., & Rivers, K. (2021). Preface to the Special Issue on Creating and Improving Adaptive Learning: Smart Authoring Tools and Processes. I*nternational Journal of Artificial Intelligence in Education,* 32*,* 1-3. https://doi.org/10.1007/s40593-021-00277-9

Olney, A. M., Graesser, A. C., & Person, N. K. (2010). Tutorial Dialog in Natural Language. In R. Nkambou, J. Bourdeau, & R. Mizoguchi (Eds.), *Advances in Intelligent Tutoring Systems, Studies in Computational Intelligence* (Vol. 308, pp. 181-206). Berlin: Springer-Verlag.

Olney, A. M., Graesser, A. C., & Person, N. K. (2012). Question Generation from Concept Maps. *Dialogue & Discourse*, 3(2), 75-99.

Olney, A. M., Pavlik Jr., P. J., & Maass, J. K. (2017). Improving Reading Comprehension with Automatically Generated Cloze Item Practice. In E. André, R. Baker, X. Hu, M. M. T. Rodrigo, & B. du Boulay (Eds.), *Artificial Intelligence in Education* (pp. 262–273). Springer. https://doi.org/10.1007/978-3-319-61425-0

Pavlik, P. I., & Eglington, L. G. (2021). The Mobile Fact and Concept Textbook System (MoFaCTS) Computational Model and Scheduling System. In S. Sosnovsky, P. Brusilovsky, R. Baraniuk, & A. Lan (Eds.), *Proceedings of the Third International Workshop on Intelligent Textbooks 2021* (Vol. 2895, pp. 93–107). CEUR. https://ceur-ws.org/Vol-2895/#paper07

Pavlik Jr., P. I., Olney, A. M., Banker, A., Eglington, L., & Yarbro, J. (2020). The Mobile Fact and Concept Textbook System (MoFaCTS). In S. Sosnovsky, P. Brusilovsky, R. Baraniuk, & A. Lan (Eds.), *Proceedings of the Second International Workshop on Intelligent Textbooks 2020 co-located with 21st International Conference on Artificial Intelligence in Education (AIED 2020)* (pp. 35–49).

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., & Miller, A. (2019). Language Models as Knowledge Bases? *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2463–2473. https://doi.org/10.18653/v1/D19-1250

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, *21*(140), 1–67.

Roberts, A., Raffel, C., & Shazeer, N. (2020). How Much Knowledge Can You Pack Into the Parameters of a Language Model? *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 5418–5426.

Sinatra, A. M. (2015). A Personalized GIFT: Recommendations for Authoring Personalization in the Generalized Intelligent Framework for Tutoring. In D. Schmorrow & C. M. Fidopiastis (Eds.), *Foundations of Augmented Cognition—9th International Conference, AC 2015, Held as Part of HCI International 2015, Los Angeles, CA, USA, August 2-7, 2015, Proceedings* (Vol. 9183, pp. 675–682). Springer. https://doi.org/10.1007/978-3-319-20816-9_64

Singley, M. K., Anderson, J. R., Gevins, J. S., & Hoffman, D. (1989). The algebra word problem tutor. *Artificial Intelligence and Education*, 267-275.

Tafjord, O., & Clark, P. (2021). *General-Purpose Question-Answering with Macaw*. arXiv. https://doi.org/10.48550/ARXIV.2109.02593

van der Lee, C., Gatt, A., van Miltenburg, E., Wubben, S., & Krahmer, E. (2019). Best practices for the human evaluation of automatically generated text. Proceedings of the 12th International Conference on Natural Language Generation, 355–368. https://doi.org/10.18653/v1/W19-8643

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010.

Yarbro, J. T., & Olney, A. M. (2021a). Contextual Definition Generation. In S. A. Sosnovsky, P. Brusilovsky, R. G. Baraniuk, & A. S. Lan (Eds.), *Proceedings of the Third International Workshop on Intelligent Textbooks* (Vol. 2895, pp. 74–83). CEUR-WS.org.

Yarbro, J. T., & Olney, A. M. (2021b). WikiMorph: Learning to Decompose Words into Morphological Structures. In I. Roll, D. McNamara, S. Sosnovsky, R. Luckin, & V. Dimitrova (Eds.), *Proceedings of the 22nd International Conference on Artificial Intelligence in Education* (pp. 406–411). Springer International Publishing.