Development and Deployment of a Large-Scale Dialog-based Intelligent Tutoring System

Shazia Afzal[1], **Tejas Indulal Dhamecha**[1], **Nirmal Mukhi**[2], **Renuka Sindhgatta**[3]*,
Smit Marvaniya[1], **Matthew Ventura**[4], **Jessica Yarbro**[4]
[1]IBM Research - India, [2]IBM Research - Yorktown Heights, NY, USA
[3]Queensland University of Technology, Brisbane, Australia, [4]Pearson Education, USA
{shaafzal, tidhamecha, smarvani}@in.ibm.com, nmukhi@us.ibm.com
renuka.sr@qut.edu.au, {matthew.ventura, jessica.yarbro}@pearson.com

Abstract

There are significant challenges involved in the design and implementation of a dialog-based tutoring system (DBT) ranging from domain engineering to natural language classification and eventually instantiating an adaptive, personalized dialog strategy. These issues are magnified when implementing such a system at scale and across domains. In this paper, we describe and reflect on the design, methods, decisions and assessments that led to the successful deployment of our AI driven DBT currently being used by several hundreds of college level students for practice and self-regulated study in diverse subjects like Sociology, Communications, and American Government.

1 Introduction

Intelligent Tutoring Systems (ITS) have been one of the ambitions of researchers working in the fields of computer-aided education, learning sciences, and, to an extent, computational linguistics (Brusilovsky et al., 1996; Graesser et al., 1999; Evens et al., 1997). A special case of ITS is Dialog-based Tutoring (DBT) which is based on the Socratic principle of cooperative dialogue meant to stimulate critical thinking and deeper comprehension (Carbonell, 1970; Graesser et al., 1999, 2005).

Dialog-based intelligent tutoring systems (DBT) capture the effectiveness of expert human teacher-learner interactions by using natural language dialogue. Since articulation of a response in natural language involves recall and reflection of relevant knowledge it facilitates deeper comprehension of content. DBT consists of a sequence of mixed initiative dialogue moves in natural language to steer learners through varying levels of content granularity. A conversation is triggered when the tutor poses a question which typically leads to a series of dialog turns directed towards finer reasoning on relevant concepts. The goal is to scaffold knowledge and provide constructive remediation akin to expert one-one human tutoring. A well known example of DBT is AutoTutor (Graesser et al., 1999, 2005) while other notable systems include Why2 (VanLehn et al., 2002), CIRCSIM-Tutor (Evens et al., 1997), GuruTutor (Olney et al., 2012), DeepTutor (Rus et al., 2013), and the GIFT framework (Sottilare et al., 2012).

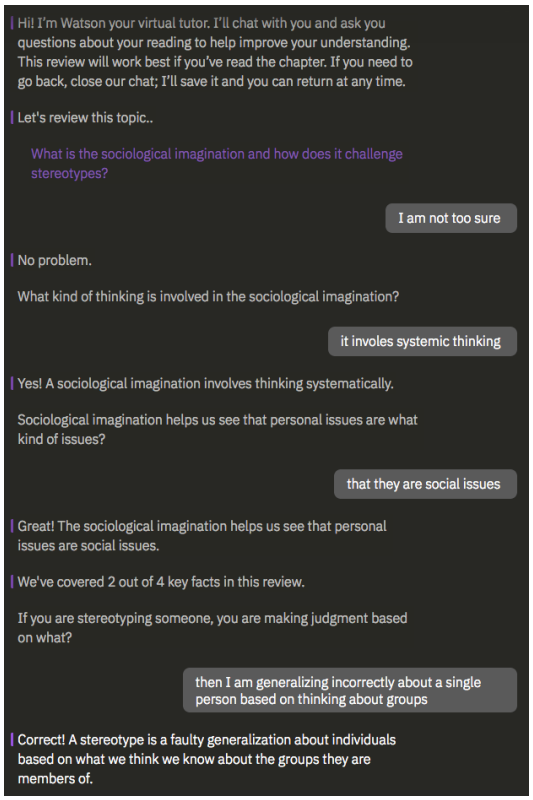

Figure 1: Screenshot of the Watson dialog-based tutor. Notice the transition from a broad question to focused question, intent classification of *I am not too sure* as need for help, and answer evaluation.

*Work done while at IBM Research - India.

Proceedings of NAACL-HLT 2019, pages 114–121
Minneapolis, Minnesota, June 2 - June 7, 2019.

Building a DBT is a challenging task as it involves balancing of conversational efficiency with the tutoring goal of personalized adaptive mentoring and knowledge assessment. In spite of sustained research and development efforts there are major challenges that prohibit their vast adoption in educational practice. Some crucial challenges can be identified as:

1.1 Student Response Analysis

Natural language classification is a critical component of any DBT and is the very basis for driving an effective conversation. Performing a context-based interpretation of a student utterance is even more challenging owing to the diversity of human language, differences in vocabulary and nuances. Consequently, current NLP and AI techniques have limitations when applied in an open-response scenario like interaction with a DBT. As machine learning techniques rely on good training data to deliver good performance, the considerable subjectivity inherent in training-data annotation, benchmarking, and reaction to misclassification errors further impacts the classification accuracy.

1.2 Content Design and Creation

Extensive content authoring is required to drive any efficient DBT. The content needs to be structured in a way that drives the tutoring agenda while ensuring that knowledge elements render naturally in a conversational flow. This requires a tremendous amount of manual and semi-automated effort from subject matter experts. Content schema is also closely tied to the nature of a domain as different subjects would have their respective challenges. For example, creating content for a factual subject like Maths or Physics is substantially different to a subject like Psychology.

1.3 Dialog Strategy

Devising a meaningful conversational strategy is a non-trivial aspect as it directly impacts learner engagement and therefore learning outcomes. There should be sufficient flexibility and variation in responses to cater dynamically to the state and requirements of individual students. Additionally, there should be scope to implement interventions and scaffolds in order to keep the learner motivated. For example, surfacing relevant examples from a textbook when a student struggles or displays lack of knowledge during interaction.

1.4 Evaluation

The overall efficacy of a learning system is primarily determined by the learning gains achieved. However, the overall learning experience is significantly impacted by other dimensions such as classification accuracy, response time, style of feedback, variation in language, robustness and usability issues. Evaluation should therefore involve a holistic assessment of all factors that eventually lead to improved learning outcomes.

1.5 Scalability and robustness

The underlying algorithm design is impacted by both the domain engineering effort involved in scaling the tutor across titles as well as the ability to handle several concurrent users. Existing architectures are more often monolithic, thereby limiting the scope of improvements and scale.

In the context of the above challenges we now describe the development and refinement of our Watson dialog-based tutoring system that has so far been used by over 2,000 college students in the domains of Sociology, Communications, and American Government. Development of this system involved creating AI modules for language understanding, designing system architecture for modularity and scalability, and a significant effort to update various designs and algorithms to incorporate student feedback received at various milestones. Our DBT differs from existing systems in terms of domain and load scalability. We approach the issue of scaling across domains using a semi-automated pipeline for authoring, validation and improvisation of content. Scaling to load is enabled by designing tutor modules as cloud-based REST micro-services.

In the following sections we describe the architecture and iterative refinement of our2 tutor followed by some qualitative and quantitative evaluations from field experiments.

2 Design and Architecture

Our Watson dialog-based tutor provides remediation to learners through natural language discourse. Systematic turn-taking engages learners in a conversation style assessment of their mastery on domain knowledge. The tutor tracks the learners progress during the course of interaction and launches appropriate interventions according to pre-defined dialog strategies. The main components of the tutor are shown in Figure 2. The

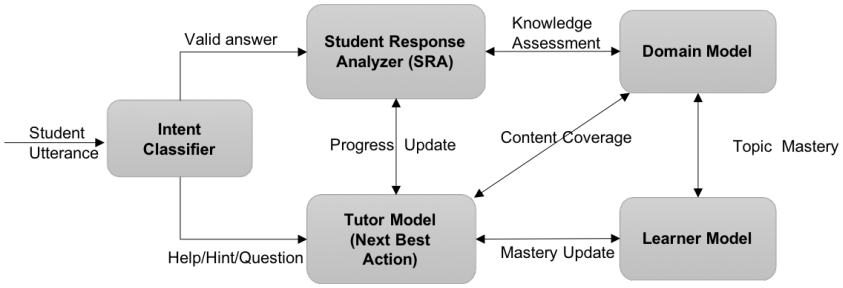

Figure 2: Simplified architecture of the tutor system.

most significant module of the tutor is the Natural Language Response Classifier comprising of two primary sub-components: the Intent Classifier and the Student Response Analyzer (SRA).

The Intent Classifier maps a student utterance to one of about 25 possible intent classes. The two main intent classes are: 1) a `valid on-topic answer` and 2) a `valid question`. Other intent classes include requests for help, requests for a hint, an expression of boredom, an insult, a greeting, and so on. This level of response classification is crucial to the effective working of the tutor as an error at this stage can have cascading effects on the entire dialog flow. This module is designed as a hierarchical classifier, going from broad intents to finer intents. The `valid on-topic answer` category requires domain/subject specific training whereas many meta-cognitive intents can be classified in domain agnostic manner.

Student Response Analysis (SRA) is the task of labeling student answers with categories that can help a dialog system to generate appropriate and effective feedback on errors. It is modeled as a classifier with underlying techniques similar to Textual Entailment (Dzikovska et al.), Semantic Textual Similarity (Agirre et al.), and Short Answer Grading (Mohler and Mihalcea, 2009). It takes the valid student answer and evaluates it against the model reference answer into one of 3 categories: `correct`, `partially correct`, and `incorrect` (Saha et al., 2018; Marvaniya et al., 2018). The SRA in our DBT thus makes use of state of art machine learning techniques to perform classification with macro-average F1 within 7% of that of human agreements; newer models (not reported here) have yielded results within 5% of human agreements. It uses an ensemble of semantic and syntactic features obtained from the tuple comprising of the question, student answer and the reference answer. This design makes it suitable for unseen-questions and, potentially for unseen-domains too. For a new domain or textbook, if student answers for training are not available, the existing base model can be used in *unseen-domain* setting. However, if the training data is available, the classification module can utilize it to improve the grading performance. In addition to response classification, SRA also performs a gap analysis on the student answer against the expected model answer to generate fill-in-the-blank (FITB) style prompts dynamically. The output of the Natural Language Response Classifier is used to drive the tutor strategy by continuous evaluation against the domain model and estimates of mastery from the learner model. The core of the tutoring framework is the domain model which is constructed by content experts as a hierarchy of learning objectives (LOs). Each LO is further structured into a sequence of assertions along with corresponding Hints and common misconceptions, if any. This formulation of content is designed to elicit knowledge gradually and allow fine-grained evaluation. With this domain model design schema, the broad task of evaluating student's understanding is broken down into set of more focused short answer evaluation.

This domain model design also enables conversational dialogue on a topic and micro-adaptation of tutorial strategy while allowing step-based assessment of learners mastery. Mastery is represented in the Learner Model which is updated on the basis of knowledge assessment on students' answers. The learner model also drives macro-adaptation between topics or LOs as defined in the pedagogical model. The pedagogical model formalizes the dialog response strategy and next-best action plan of the tutor using principles of formative feedback, hinting tactics and content coverage. Currently we use a rule-based dialog strategy instantiated for each student based on their current mastery and interaction history. (Shute,

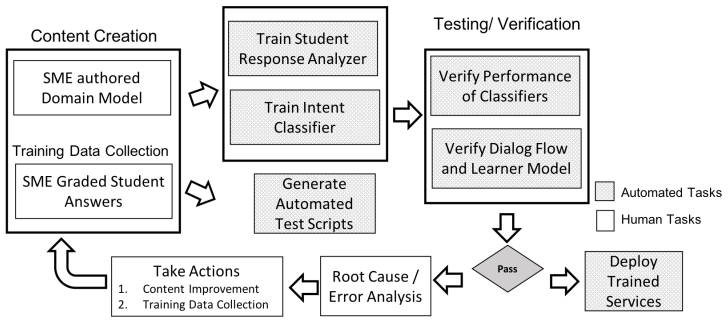

Figure 3: Sequence of tasks carried out to deploy the tutor for a new subject or domain

2008; Hume et al., 1996). The interface model is the front-end that triggers the UI to enable Tutor-Learner interaction in natural language. As our DBT is conceived to be used as a supplement to digital learning content, it is made strategically available to students for revision and practice.

3 Deployment and Scalability

All modules of the tutor are functionally isolated and are manifested as RESTful microservice APIs. This decoupling is important for scalable design. The overarching framework of Orchestration connects all the services. It is implemented in OpenWhisk (Mukhi et al., 2017) which follows a serverless computing based cloud-computing execution model. The front-end communicates with the Orchestration via broadly abstracted tutor APIs only. Depending on the functionality, some modules are domain agnostic (e.g. learner model, next-best-action), whereas others are not. For the domain dependent modules, a set of microservice instances are spawned per domain. In terms of the storage facility, NoSQL IBM Cloudant®, in-memory data structure store Redis, and Cloud Object Storage System™ are employed. Relatively small and structured data (e.g. domain content) is maintained in Cloudant; high frequency update data (e.g. student progress) is cached in Redis; whereas large AI models are stored in Cloud Object Storage. To cater to computational and traffic load, horizontal (in terms of processor and memory) and vertical (in terms of number of instances) scaling is facilitated by Kubernetes and Nginx API Gateway. Specifically in the context of commercial ITS systems, scalability has two aspects: development & load management.

3.1 Scalable to Develop

This corresponds to the ability to scale the tutor to new subjects/domains. The overall turnaround time to make the tutor usable for a new subject should be reasonably low. Figure 3 shows key tasks required to have the tutor system support a new subject such as creation of the domain model, retraining/transfer-learning of AI/NLP modules, testing the dialog and AI module performance, configuration tuning, and deploying the system. Some of these, such as domain model creation, evaluating and fine tuning the model are human labor intensive, whereas others have been automated. By design, all of these are distributable across multiple humans and machines which significantly helps formulate a *factory model* for preparing the tutor for a new subject. The training and deployment time for a new subject ranges between 6 hours - 72 hours, depending on the amount of refinement needed in the scoring models.

One of the key design elements for development scalability is re-usability of AI/NLP components with minimal modifications across various subjects. The SRA module uses a base model that can be used across all subjects. The SRA with only the base model yields a modest accuracy; it is extended with domain specific unstructured (text) and structured (student answers) to transfer it to the subject/domain. This facilitates *bootstrapping* the module and solves the problem of *cold start* often encountered in industry AI product development.

3.2 Scalable to Load

This refers to the runtime load tolerance of the tutor including simultaneous access to possibly several hundred students. Moreover, student enrollments may vastly differ per subjects and student activity may be very high during certain periods (e.g. prior to tests), and limited at other times (e.g. breaks). All these aspects require dynamic allocation of resources to various tutor modules.

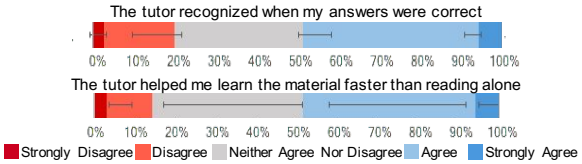

Figure 4: Response classification related survey question results illustrating positive impacts of tutor.

Some number crunching heavy modules benefit more from vertical scaling by virtue of inherent parallelization whereas gating/orchestrating modules typically benefit more from horizontal scaling. Our tutor can support 600 concurrent users (corresponding to $\sim$ 300K students using the tutor regularly over a semester) with error rate $< 0.5\%$ and response time $< 3s$ on average. The error rates and response times are much better for more typical loads (5 – 50 concurrent students) observed in current deployments.

4 Iterative Improvement of the System: Development Life Cycle

Various micro-experiments were done after every few internal releases to receive feedback from potential end users (sub-sampled student population). These micro-experiments helped evaluate the product features and robustness. Following are some of the key improvisations.

4.1 Evolution of the Response Classifier

Originally, the response classification was binary: `correct` or `incorrect`. A misclassification therefore resulted in either severe penalty or leniency. To mitigate this risk, the module was retrained to a three-way classifier, with the low-risk third class as `partially correct`. Doing so, indeed reduced the chance of `correct`-to-`incorrect` and vice-versa misclassification. However, in later trials we observed that if a student's answer is misclassified, it is better to suggest a finer follow up activity, than just providing partial grade. For example, by performing a gap analysis on the students answer against the expected one we dynamically generate fill-in-the-blank (FITB) style prompts or use encouragement pumps based on mastery.

4.2 Content Revision

During trials, we encountered cases, where student answers were correct in context of the textbook, but were wrong as per the authored content. This called for a need to update/enrich the authored content based on the actual student answers. As a solution, the students were given the option to rate the tutor's response using a thumbs up/down sign. These ratings are tracked to draw an SME (subject matter expert)'s attention to improve/rectify the content. This semi-automated approach allows for continuous user feedback and incremental training of the underlying modules.

4.3 Improvising the Conversation Tone

Feedback from trials was also leveraged to fine tune the tone of the tutor responses to make it more motivating, less pedantic, personable and more transparent. For example, if a student answer was classified as incorrect, the prompts `That's incorrect!`, `That seems to be a wrong answer` and `That does not match with what I have` convey the same message but with different levels of critique and encouragement. Similarly, variation in the feedback language was personalized according to students ability level and punctuation was strategically used to give a sense of enthusiasm and positivity. The overall persona and the personalized behavior were carefully revised and tuned according to situational needs. See (Afzal et al., 2019) for more details on the design and evolution of the tutor personality.

4.4 Dialog Flow Refinement

The dialog flow is essentially the entire orchestration of a tutoring goal, personalized for each student depending on their progress and mastery on specific content. Although, our initial dialog flow included two main activities of student answering questions and asking questions, we added additional activities to the dialog to adapt it to student mastery and improve engagement. These included recommending question asking based on questions asked by peers, presenting simpler binary true/false questions or presenting a fill-in-the blank question if the computed hint question difficulty is high and student mastery is low for the topic.

5 Evaluation

As of August 2018, the Watson dialog-based tutor is available with 3 titles commercially, with a number of additional titles in development (Ventura

		Human-2		
		C	P	I
Human-1	C	57.6	20.2	22.2
	P	5.8	27.2	67.0
	I	3.1	5.3	91.6
Macro-F1: 60.2%				

		Tutor		
		C	P	I
Human-1	C	64.8	19.5	15.7
	P	31.9	23.4	44.7
	I	5.7	20.8	73.5
Macro-F1: 53.2%				

		Tutor		
		C	P	I
Human-2	C	69.0	17.4	13.6
	P	32.0	34.4	33.6
	I	11.4	13.8	74.8
Macro-F1: 55.9%				

Table 1: Confusion matrices (values in percentage) of Human-1 vs Human-2 and Humans vs Tutor for response classification during a learning experiment. C, P, and I represent `correct`, `partially correct`, and `incorrect` grade.

et al., 2018). During the fall 2018 semester, the commercially available titles were used by over 2000 students across more than 200 higher education institutions (Pearson, 2018). The system is expected to be used across a multitude of higher education institutions in the USA and will be piloted on five additional titles.

Here, we present our initial results gathered from two controlled learning experiments (LE) conducted 7 months apart. The first LE was conducted with 39 students while the second LE had 102 students. The demographic profile was similar with average GPA of $\sim$ 3.5 and 70% female ratio. Here we discuss some relevant results on the following key dimensions:

5.1 Classification

For our two-level classification system - at intent and answer assessment level (SRA), we observed that intent classification of student utterances is 95% accurate, and the SRA system for scoring answers is within 7% of human inter-rater reliability for 3 of the 4 learning objectives tested. Figure 4 shows the student responses to survey questions that measure the efficacy of the tutor while Table 1 illustrates the evaluation of classification through formal metrics. The confusion matrices are reported for 50 transcripts pertaining to total of 1,065 student responses. Note that, `I`-to-`C` misclassification is 2.6-8.3% higher than that of human disagreement; whereas `C`-to-`I` misclassification rate is better than human disagreement. User's feedback on how well the tutor understood their responses showed a statistically significant across the two LEs.

5.2 Validity

The tutor's estimate of student mastery scores was found to correlate significantly with the student's self-perception ($r = 0.32, p < 0.05$) and actual post–test learning measures ($r = 0.53, p < 0.001$). This is a strong indicator of the validity of the tutor's mastery measurement. Note that this validity is crucial as it is the basis for adjusting the dialog strategy to enable true personalization of the tutoring experience.

5.3 Dialog Quality

This was estimated by surveying the SMEs and students who used the system. The SMEs manually scored 236 conversational transcripts and took a survey after each transcript. In these surveys, the tutor was rated as providing an effective tutoring conversation 70% of the time, transitioning effectively through the conversation 75% of the time and providing appropriate feedback 66% of the time. Students' self–report was also fairly positive about the overall experience, feedback, value and usability of the tutor.

5.4 Iterative Improvement

Figure 5 reports the improvements observed across the two LEs on some survey items. These improvements are attributed to content refinement, response and intent classification improvement, and dialog refinement.

Perhaps more gratifying than the generally positive results are the expressions of enthusiasm, interest and engagement from students working with the Watson dialog-based tutor. Comments such as *"I wasn't having to go over reading material and lecture material over and over to understand it more fully because I grasped it the first time. I could take that time to devote to another subject, or my family"* was evidence of the truly beneficial societal impact that such a system could have, if applied the right way.

6 Discussion: Human Subjectivity

We observed that the subjectivity inherent in human language impacts the content authoring, creation of training data and, eventually, the end user experience. The domain model authoring raises

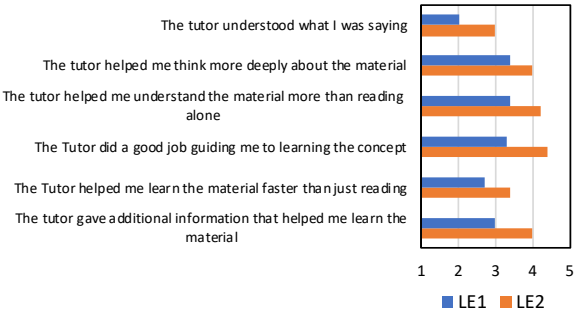

Figure 5: Comparison of LE1 & LE2 on survey items scored on 1-5 scale.

issues pertaining to the granularity of content and the resulting ambiguity expected in responses. For example (See Figure 6a), we observe that *recall* type question with `WHAT` interrogative pronouns are more likely to be answered correctly compared to `HOW` questions which may be ambiguous due to relative openness of expected answers. This implies that classification accuracy can be improved if a more standardized vocabulary is followed to limit the variation in content creation. However, this is a debatable proposition since it undervalues the richness of domain knowledge and may not be practical for non-factual domains.

Additionally, given a domain authored content like a QA pair, it is hard even for experts to determine the boundaries of correctness in student responses with respect to a reference model answer. This poor inter-rater agreement directly impacts the effectiveness of training the underlying scoring modules. To give an estimate, in one of our training data sets, three groundtruth labels were obtained from domain experts for each student answer. There was disagreement on 50% of the answers marked as partially correct by experts! Although this is a classical problem in ML/NLP, the consequences of scoring a correct answer incorrectly or vice-versa are more profound in a learning system. Human teachers are more flexible and sometimes abstract in their evaluation of an open-ended answer which is difficult to replicate in an AI system.

Finally, it is the delivery of an overall engaging and valuable experience that matters to human end users. Even here, there is remarkable subjectivity in users' perception and tolerance to the type and timing of tutor misclassifications (Afzal et al., 2018). We cannot measure efficacy solely in terms of absolute and restrictive metrics like accuracy and macro-average F1. Our best proxies, then, are

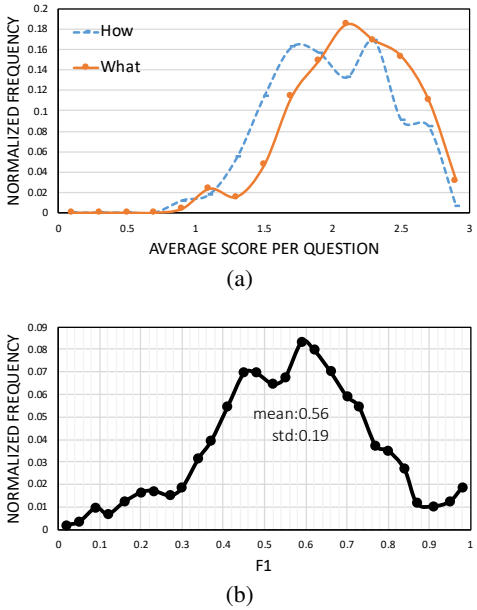

Figure 6: Quantifying issues related to human subjectivity: (a) ambiguity in answer correctness as function of question type, (b) poor inter-annotator agreement.

qualitative, detailed and timely feedback from the end users to learn and improvise over time.

7 Summary

Conversational tutoring is an important form of next-generation personalized adaptive educational technology. In this paper we have described the design and iterative development of Watson dialog-based tutor – a large-scale DBT that is optimized to scale across domain/subjects as well as usage. Its modules are functionally isolated to facilitate development and runtime scalability. We have described various challenges related to content creation and design including their impact on classification performance, refinement of feedback phrasing and tone, and dialog strategy. We have highlighted issues that arise from the inherent diversity of human language and how they impact the functioning of the tutor and the generated learning experiences. On the design side, automation of content extraction techniques can significantly speed up the content scaling process and allow building of richer domain models by making use of learning material from additional sources. On the experience side, substantial effort is needed to accurately understand natural language and use it strategically to deliver a naturalistic conversational interface that replicates the effectiveness of human teacher-learner interactions.

References

Shazia Afzal, Bryan Dempsey, Cassius D'Helon, Nirmal Mukhi, Milena Pribic, Aaron Sickler, Peggy Strong, Mira Vanchiswar, and Lorin Wilde. 2019. The personality of ai systems in education: Experiences with the watson tutor, a one-on-one virtual tutoring system. *Childhood Education*, 95(1):44–52.

Shazia Afzal, Vinay Shashidhar, Renuka Sindhgatta, and Bikram Sengupta. 2018. Impact of tutor errors on student engagement in a dialog based intelligent tutoring system. In *ITS*, pages 267–273. Springer.

Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. Semeval-2012 task 6: A pilot on semantic textual similarity. In *SemEval-2012*, pages 385–393. ACL.

Peter Brusilovsky, Elmar Schwarz, and Gerhard Weber. 1996. ELM-ART: an intelligent tutoring system on world wide web. In *ITS*, pages 261–269.

Jaime R Carbonell. 1970. AI in CAI: An artificial-intelligence approach to computer-assisted instruction. *IEEE TMMS*, 11(4):190–202.

Myroslava Dzikovska, Rodney Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa Trang Dang. Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. In *SemEval-2013*, volume 2, pages 263–274. ACL.

Martha W Evens, Ru-Charn Chang, Yoon Hee Lee, Leem Seop Shim, Chong Woo Woo, Yuemei Zhang, Joel A Michael, and Allen A Rovick. 1997. Circsim-tutor: An intelligent tutoring system using natural language dialogue. In *ANLP: Descriptions of system demonstrations and videos*, pages 13–14. ACL.

Arthur C Graesser, Patrick Chipman, Brian C Haynes, and Andrew Olney. 2005. Autotutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE ToE*, 48(4):612–618.

Arthur C Graesser, Katja Wiemer-Hastings, Peter Wiemer-Hastings, Roger Kreuz, Tutoring Research Group, et al. 1999. AutoTutor: a simulation of a human tutor. *Cognitive Systems Research*, 1(1):35–51.

Gregory Hume, Joel Michael, Allen Rovick, and Martha Evens. 1996. The use of hints by human and computer tutors: the consequences of the tutoring protocol. In *ICLS*, pages 135–142.

Smit Marvaniya, Swarnadeep Saha, Tejas I Dhamecha, Peter Foltz, Renuka Sindhgatta, and Bikram Sengupta. 2018. Creating scoring rubric from representative student answers for improved short answer grading. In *CIKM*, pages 993–1002. ACM.

Michael Mohler and Rada Mihalcea. 2009. Text-to-text semantic similarity for automatic short answer grading. In *EACL*, pages 567–575. ACL.

Nirmal K Mukhi, Srijith Prabhu, and Bruce Slawson. 2017. Using a serverless framework for implementing a cognitive tutor: experiences and issues. In *ACM WoSC*, pages 11–15.

Andrew M Olney, Natalie K Person, and Arthur C Graesser. 2012. Guru: designing a conversational expert intelligent tutoring system. In *Cross-disciplinary advances in applied natural language processing: Issues and approaches*, pages 156–171. IGI Global.

Pearson. 2018. Revel with watson. https://www.pearson.com/us/higher-education/products-services-teaching/digital-learning-environments/revel-with-watson.html.

Vasile Rus, Sidney DMello, Xiangen Hu, and Arthur Graesser. 2013. Recent advances in conversational intelligent tutoring systems. *AI magazine*, 34(3):42–54.

Swarnadeep Saha, Tejas I Dhamecha, Smit Marvaniya, Renuka Sindhgatta, and Bikram Sengupta. 2018. Sentence level or token level features for automatic short answer grading?: Use both. In *AIED*, pages 503–517.

Valerie J Shute. 2008. Focus on formative feedback. *Sage RER*, 78(1):153–189.

Robert A Sottilare, Keith W Brawner, Benjamin S Goldberg, and Heather K Holden. 2012. The generalized intelligent framework for tutoring (GIFT). *Orlando, FL: US Army Research Laboratory–Human Research & Engineering Directorate*.

Kurt VanLehn, Pamela W Jordan, Carolyn P Rosé, Dumisizwe Bhembe, Michael Böttner, Andy Gaydos, Maxim Makatchev, Umarani Pappuswamy, Michael Ringenberg, Antonio Roque, et al. 2002. The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In *ITS*, pages 158–167.

Matthew Ventura, Maria Chang, Peter Foltz, Nirmal Mukhi, Jessica Yarbro, Anne Pier Salverda, John Behrens, Jae-wook Ahn, Tengfei Ma, Tejas I Dhamecha, et al. 2018. Preliminary evaluations of a dialogue-based digital tutor. In *AIED*, pages 480–483.