



Automated Data-Driven Generation of Personalized Pedagogical Interventions in Intelligent Tutoring Systems

Ekaterina Kochmar^{1,2} · Dung Do Vu^{1,3} · Robert Belfer¹ · Varun Gupta¹ · Iulian Vlad Serban¹ · Joelle Pineau^{1,4}

Accepted: 4 July 2021 / Published online: 27 July 2021
© The Author(s) 2021

Abstract

Intelligent tutoring systems (ITS) have been shown to be highly effective at promoting learning as compared to other computer-based instructional approaches. However, many ITS rely heavily on expert design and hand-crafted rules. This makes them difficult to build and transfer across domains and limits their potential efficacy. In this paper, we investigate how feedback in a large-scale ITS can be automatically generated in a data-driven way, and more specifically how personalization of feedback can lead to improvements in student performance outcomes. First, in this paper we propose a machine learning approach to generate personalized feedback in an automated way, which takes individual needs of students into account, while alleviating the need of expert intervention and design of hand-crafted rules. We leverage state-of-the-art machine learning and natural language processing techniques to provide students with personalized feedback using *hints* and *Wikipedia-based explanations*. Second, we demonstrate that personalized feedback leads to improved success rates at solving exercises in practice: our personalized feedback model is used in Korbit, a large-scale dialogue-based ITS with around 20,000 students launched in 2019. We present the results of experiments with students and show that the automated, data-driven, personalized feedback leads to a significant overall improvement of 22.95% in student performance outcomes and substantial improvements in the subjective evaluation of the feedback.

Keywords Intelligent tutoring systems · Dialogue-based tutoring systems · Natural language processing · Deep learning · Personalized learning · Personalized feedback · Data science education

✉ Ekaterina Kochmar
ekaterina@korbit.ai

Introduction

Personalized tutoring helps students achieve their learning goals effectively (Anania, 1983; Bloom, 1984; Burke, 1983; Hrastinski et al., 2019; Hume et al., 1996). Traditionally, such personalized tutoring has been provided by human tutors. The benefits of having a human tutor include a tutor's ability to understand the effective state of the student, and thus provide personalized feedback by adapting instructions accordingly. Conventional settings, such as teaching in larger groups of students, necessarily lead to challenges in addressing each student's personal needs, however one-on-one tutoring is generally seen as too costly to be conducted on a large scale in most societies, and is thus not readily available.

Intelligent Tutoring Systems (ITS), “*computer-based instructional systems with models of instructional content that specify what to teach, and teaching strategies that specify how to teach*” (Wenger, 1987), attempt to mimic personalized human tutoring in a computer-based environment and are a low-cost alternative to human tutors (Anderson et al., 1985; Nye et al., 2014). ITS are capable of providing step-by-step guidance during problem solving, tracking students' skills and knowledge development, and selecting problems on an individual basis. When compared to other computer-based learning environments (e.g., Massive Open Online Courses), ITS have been shown to be more effective in promoting learning, with the particular strength of ITS lying in their ability to deal with the interactive and personalized aspects of individual learning effectively (Hone & El Said, 2016; Kulik & Fletcher, 2016; VanLehn, 2011).

However, one major bottleneck to a wider-spread use of ITS is the expensive and laborious process of creating content and pedagogical interventions. Many ITS rely heavily on expert design and hand-crafted rules to generate system interventions, which makes them difficult to build and transfer across domains, and limits their potential efficacy and scalability (Folsom-Kovarik et al., 2010; Olney & Cade, 2015). In this paper, we address this major bottleneck in ITS development, and make two significant contributions.

First, we describe how state-of-the-art machine learning (ML) and natural language processing (NLP) techniques can be used to automatically generate data-driven *personalized hints* and *Wikipedia-based explanations*. Feedback generated this way takes the individual needs of students into account, does not require expert intervention or hand-crafted rules, and is expected to be easily scalable and transferable across domains. Second, we demonstrate that the personalized feedback leads to substantially improved student performance outcomes and improved subjective feedback evaluation in practice.

To support our claims, we utilize the personalized feedback models in *Korbit*, a large-scale dialogue-based ITS, which was launched in 2019 and today has around 20,000 students enrolled in courses on machine learning and data science. We present the results of the experiments run on the *Korbit* learning platform remotely between January and February, 2020, involving 796 annotated student–system interactions collected from 183 students enrolled for free. We measure student success rate as the proportion of instances where a student provides a correct solution after receiving a hint or explanation from our ITS. The results show that personalized

feedback provided on our platform significantly increases performance outcomes, as it leads to an average success rate of 60.47% at solving exercises on the platform. Moreover, we observe a substantial improvement in subjective feedback evaluation provided by the students.

Related Work

In this section, we first overview previous work related to the development of ITS in various domains, and then we discuss applications of NLP techniques in ITS for adaptivity, personalization and automated feedback generation.

Intelligent Tutoring Systems

Over the past two decades, many ITS have been successfully deployed to enhance teaching and improve students' learning experience in a number of domains and application areas. In particular, ITS have been actively used to teach technical subjects: from helping students acquire knowledge about mathematics (Büdenbender et al., 2002; Dietrich & Buckley, 2008; Goguadze et al., 2005; Hrastinski et al., 2019; Koedinger & Anderson, 1993; Melis & Siekmann, 2004; Passier & Jeuring, 2006; Sommer & Nuckols, 2004), logic (Abel et al., 2001; Andrews et al., 2004; Burstall, 1998; D'Agostino & Endriss, 1998; Hendriks et al., 2010; Scheines & Sieg, 1994; Stamper et al., 2013; Sufirin & Bornat, 1996), and algorithms (Leelawong & Biswas, 2008); to assisting students in knowledge and skill acquisition in natural sciences (Hume et al., 1996; Makatchev et al., 2011; Zhang & VanLehn, 2016; 2017); to teaching real-world applications. Apart from providing students with general assistance and feedback on their performance, ITS are able to address individual student characteristics (Graesser et al., 2017) and cognitive processes (Wu & Looi, 2010).

Since students differ in terms of their aptitudes and knowledge, personalized instruction in education is critical for effective learning. Personalization and adaptability of ITS to individual student needs have been shown to not only help students in independent learning, but also help teachers personalize feedback and instruction, in particular in blended and flipped-classroom environments (Baker, 2016; Holstein et al., 2017, 2019).

Many ITS incorporate explicit student models and consider the development of a personalized curriculum and personalized feedback (Albacete et al., 2019; Chi et al., 2011; Lin et al., 2013; Munshi & Biswas, 2019; Rus et al., 2014a, b). In this respect, *dialogue-based* ITS have been shown to be some of the most promising tools for learning (Ahn et al., 2018; Graesser et al. 2001, 2005, Nye et al., 2014; Ventura et al., 2018), as they simulate the familiar learning environment of student–tutor interaction, which helps improve student confidence and motivation and leads to a better learning experience. In particular, dialogue-based ITS mimic the familiar student–tutor interaction setting by asking students questions and presenting them with problem-solving exercises, while also providing students with the opportunity to pose their own questions, request hints and explanations, and engage in other types of communication with the tutor.

The tradition to structure tutoring around active dialogue and, in particular, in the manner of asking questions and eliciting answers related to the subject material, dates as far back as the Socratic method and Plato's academy (Mills et al., 1980). Previous research shows that when students attempt to provide answers, they get involved in such constructive activities as reflecting on the taught material, explaining material to themselves as well as to others, self-assessing and understanding the level of their knowledge, and connecting different areas of the subject, among others (Graesser & Person, 1994, 1995; Hrastinski et al., 2019; Hume et al., 1996). Such activities are central to reasoning and understanding (Ram, 1991; Webb, 1989). In addition, the selection of questions to present students with and the analysis of their performance in answering these questions is critical for curriculum structuring itself, both for human tutors and in ITS (Boaler & Brodie, 2004; Jiang, 2014). Here, ITS can structure their curriculum appropriately by selecting the questions according to each student's individual development.

At the same time, the main bottleneck in providing students with personalized feedback in ITS is the ability of such systems to address the multitude of possible scenarios in student–system interactions, and this is where methods of automated, data-driven feedback generation are of critical importance. Much of the work investigating personalized feedback incorporates or takes inspiration from research on student–teacher instructional scaffolding (Van de Pol et al., 2010; Wood, 2003).

In this paper, we focus on delivering personalized feedback in a dialogue-based ITS during problem-solving exercises. Such feedback includes hints, explanations, elaborations, and prompts, among other pedagogical interventions. Following up on the promising results from past research, we investigate how we can leverage large amounts of open-access data in creating educational content. Of particular relevance here is the line of related work, where researchers have investigated how machine learning and large-scale, open-access resources such as Wikipedia can be utilized to generate various types of educational content and interactions with the aim of scaling up computer-based learning systems and addressing the needs of their students (Brunskill et al., 2018; Dinan et al., 2018; Guo et al., 2016; Liu et al., 2012; Willis et al., 2019). In particular, it has been shown that the use of NLP techniques in application to Wikipedia may be helpful in generating pedagogically motivated concept maps to be used within an ITS (Lahti, 2009); identifying pre-requisite relations and sequencing among learning concepts to better model the learning path of the student and assess gaps in student's understanding of the subject (De Medio et al., 2016; Ramírez-Noriega et al., 2018; Talukdar & Cohen, 2012); and generating a variety of pedagogical interventions ranging from open questions (Liu et al., 2012; Shah et al., 2017) to multiple-choice quizzes (Guo et al., 2016; Tamura et al., 2015) across a number of subject domains.

Natural Language-based Interactions in ITS

A number of previous approaches designed dialogue-based ITS using natural language interface and allowing students to provide unrestricted input to the system (Benzmüller et al., 2007; Makatchev et al., 2011; Person et al. 2000; Stamper et al., 2013). Previous research shows that such unrestricted interaction helps support

meta-cognitive processes in students, while also helping the system identify misconceptions in students' reasoning (Makatchev et al., 2011). Since such systems are working towards providing students with an opportunity to interact with the tutor in an unrestricted manner, this leads to further challenges related to natural language *understanding* on the one hand, and to natural language-based *generation* of interactive and personalized feedback and interventions on the other hand. In this paper, we primarily focus on selection and generation of personalized feedback from existing natural language text.

In a tutorial dialogue, where one participant represents a teacher, an expert on the subject, or a more knowledgeable partner (in particular, such a partner may be represented by a human or an AI tutor) and another participant is a less knowledgeable partner (i.e., a student), hinting is a widely-used tactic (Hume et al., 1996). Hume et al. (1993) define a hint as “*a rhetorical device that is intended to either: (1) provide the student with a piece of information that the tutor hopes will stimulate the student's recall of the facts needed to answer a question, or (2) provide a piece of information that can facilitate the student's making an inference that is needed to arrive at an answer to a question or the prediction of system behavior*”. Hints are aimed at encouraging students to engage in active cognitive processes that are thought to promote deeper understanding and long-term retention. It is important to note that while hints are widely used by teachers to prompt students to correct their errors, they normally do not provide the full information the students need to solve a particular problem (Hume et al., 1996). Hume et al. (1996) identify hints that convey information needed to arrive at an answer and those that point students to the relevant information that they already possess as the two main types of hints used in practice. They further distinguish between hints in the form of *explanations*, *summaries*, *questions*, and *negative acknowledgements*. In this work, we focus on generating hints in the form of *explanations*, pointing students at the relevant information and conveying related facts without revealing the actual answer.

Previous work investigated the impact of data-driven hints on educational outcomes in terms of learning and persistence. In particular, Stamper et al. (2013) augment their Deep Thought logic tutor with a Hint Factory that generates data-driven, context-specific hints for an existing computer aided instructional tool. Specifically, hints are generated for logic proof solving indicating a goal expression to derive, the rule to apply next, the premises where the rule can be used, or the combination of all the above. The results show that students, who receive hints, attempt and complete significantly more problems compared to the control, no-hint group. Moreover, students who receive hints early in the learning process outperform all other students in the post-test. These results suggest that data-driven hints are effective in promoting learning, however, the data-driven component in the Hint Factory is primarily concerned with automated detection of the best hint sequence depending on the level of complexity and the amount of the full proof revealed. In contrast, our work addresses NLP-based generation of hints in a natural language and is potentially applicable to multiple domains.

There is a growing body of research on automated hint generation for programming exercises (McBroom et al., 2019; Price et al., 2019). Most work in this area is concerned with *detection or generation of a suitable sequence of hints* to provide

to students at specific points during their learning, and hints are mainly generated using templates combining mixed-language input (Rivers, 2017). This line of work is related to ours, however we note that hints related to programming exercises are mostly concerned with procedural knowledge, whereas our platform addresses both procedural and declarative knowledge. In addition, interactions on our platform are more open-ended and involve more unrestricted language.

NLP techniques have also been widely used to model natural language understanding (NLU) components within ITS. For instance, Benz Müller et al. (2007) introduce a dialogue system into a mathematical assistance tool, where a student builds a proof by producing natural language utterances, and the system provides them with domain-specific hints produced when the student is stuck or shows non-understanding of domain concepts. The NLU module in Benz Müller et al. (2007) uses a specialized syntactic parser and relies on an in-domain semantic interpretation. Similar to this work, Alevén et al. (2001) are mostly concerned with the challenges in NLU and the interpretation of a mixed language input from the student, rather than with the natural language-based generation of pedagogical interventions.

Finally, Zhang and VanLehn (2016) and Zhang and VanLehn (2017) consider the use of NLP techniques in automated generation and adaptation of questions on biology to learner profiles using semantic network, and thus alleviating the need for domain expert intervention. They show that students provided with adaptive question selection have larger learning gains than those with mal-adaptive question selection.

To summarize, in contrast to the previous work, we apply NLP techniques to generate hints expressed in a natural language. Since we do not rely on the use of hand-crafted rules or templates, our methodology can be applied to any input domain and potentially address both declarative and procedural knowledge.

Korbit Learning Platform

The Korbit learning platform is an e-learning platform, which hosts the Korbit ITS.¹ Korbit is a large-scale, open-domain, dialogue-based ITS, which uses machine learning, NLP and reinforcement learning to provide interactive, personalized learning online. Currently, the platform has around 20,000 students enrolled and is capable of teaching topics related to data science, machine learning, and artificial intelligence. The platform is highly modular and scalable, and is currently being expanded with more subjects and facilitated by the use of data-driven approaches presented in this paper.

Students enroll based on courses or skills they would like to study, which provides them with the first step in personalizing their learning experience. For instance, upon enrolling a student may choose which skills they would like to focus on (e.g., classification analysis, regression analysis, applying neural networks) and select among application domains (e.g., object detection in images, sentiment analysis in reviews, etc.). Once a student has enrolled, Korbit tutors them by alternating between short

¹<https://www.korbit.ai>

lecture videos and interactive comprehension and problem-solving exercises. During the interactive sessions, Korbit shows the student an exercise problem statement (e.g., a question). The student may then attempt to solve the exercise, ask for help, or even skip the exercise. If the student attempts to solve the exercise, their solution attempt is compared by an NLP-driven solution verification module against the expectation stored internally in our database (i.e. reference solution, which typically consists of one or two sentences containing all relevant information that should be included in the correct answer to the question posed). If their solution is classified as incorrect, then the inner-loop system will activate and respond with one of a dozen different pedagogical interventions, as Fig. 1 demonstrates. The pedagogical interventions include hints, explanations, elaborations, mathematical hints, concept tree diagrams, and multiple choice quiz answers. Each pedagogical intervention is chosen by an ensemble of machine learning models based on the student's learning profile and last solution attempt, which helps ensure high level of personalization in tutoring. At the moment, questions, reference solutions, and certain types of pedagogical interventions on our platform are not automatically generated but rather created manually by our course designers. We consider development of data-driven methods aimed at facilitating content creation our future work.

In this paper, we present experiments on the Korbit platform with actual students, who are enrolled in the courses on machine learning and data science on the free basis. These experiments involve automatically generated feedback varied based on how the pedagogical interventions were generated and how they were adapted to each unique student, as during the interactive sessions, questions and hints are selected for the student by our models. The highly scalable nature of pedagogical interventions generation ensures that Korbit can effectively address educational needs of a wide variety of students.

Automatically Generated Personalized Feedback

The Korbit ITS utilizes different types of data sources in order to automatically generate a large variety of personalized feedback. In this section, we describe in detail the automatic generation process for *personalized hints* and *Wikipedia-based explanations*. These constitute two of the many intervention types employed by the

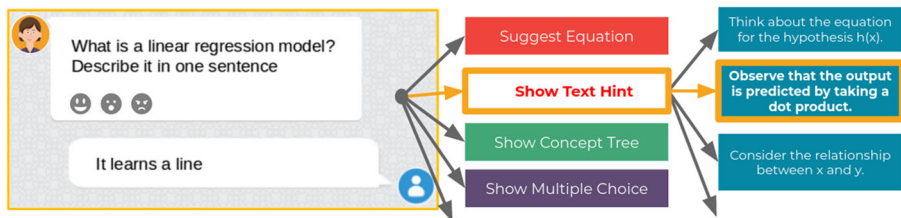


Fig. 1 The Korbit ITS: An example illustrating how the ITS inner-loop system selects the pedagogical intervention. The student gives an incorrect solution and afterwards receives a text hint

Korbit ITS. We also present three personalized feedback selection models applied to personalized hints.

Personalized Hint Generation and Selection

Personalized hints are generated using NLP techniques and assessed according to a number of metrics related to the quality of the feedback, as well as the past interaction of the system with the student.

Hint Generation

The system generates a large set of hints by applying linguistic patterns to all expectations (i.e. reference solutions) available in our database. Table 1 demonstrates some examples of hints generated using our 3-step algorithm detailed below:

1. *Identification of keywords and keyphrases*: Keywords and keyphrases include nouns and noun phrases within the question provided by the ITS, and are automatically identified using linguistic analysis with spaCy.² spaCy is used in this work, since it provides us with useful functionality and helps us with linguistic analysis including lemmatization, part-of-speech tagging, dependency parsing, chunking, and named entity recognition, among other steps. In the examples in Table 1 keywords and phrases are marked with text boxes: for instance, *overfitting* and *underfitting*, as well as *logistic regression* and *linear regression* are automatically identified as keywords and phrases.
2. *Identification of an appropriate sentence span*: It would seem likely the best hints should not include keywords, keyphrases and related words as they may reveal the exact solution to the student. We apply state-of-the-art dependency parsing with spaCy to eliminate parts of the expectation sentences that contain keywords and phrases: for instance, the first example in Table 1 contains two clauses – *A model is underfitting* and *when it has a high bias*. The first clause is filtered out since it contains *underfitting*, a term related to one of the key terms. Specifically, we define key terms as nouns or noun phrases, and one of the key terms extracted from the question here is the noun *underfitting*. We consider the verb form *underfitting* used in the expectation a related word, as we would for any other morphologically related term. At the same time, *it has a high bias* is considered as a candidate for hint generation. Similarly, among the two clauses in the second example from Table 1, namely *I would use logistic regression* and *because the outputs are discrete*, the first one is filtered out since it contains the keyphrase *logistic regression*, while the second one is considered as a candidate for hint generation.
3. *Generation of a grammatically correct hint* is done automatically using discourse-based modifications. Specifically, to convert clauses extracted from expectations in the previous step, we use discourse-based modifications such as *Think about the case when* or *Think about the following*: to produce well-formed

²<https://spacy.io>

Table 1 Text-based hint generation

Question	Expectation	Generated hint
What is the difference between overfitting and underfitting ?	A model is underfitting when it has a high bias.	<u>Think about the case when</u> it has a high bias.
Would you use linear regression or logistic regression to model a classification problem ?	I would use logistic regression , because the outputs are discrete.	<u>Think about the following</u> : the outputs are discrete.

Keywords and phrases are marked with boxes, discourse-based modifications are underlined

sentences. Thus, in the first example from Table 1 we complement the partial hint extracted from an expectation in step (2) (*it has a high bias*) with a discourse modification *Think about the case when*, and in the second example we use *Think about the following*: to complement the partial hint *the outputs are discrete*. To generate a diverse set of hints, other discourse modifiers include such verbs as *note*, *observe*, *recall*, and *consider*.

Analysis of hint transparency: Our goal in providing students with the hints is to prompt them to correct their errors and remedy their misunderstandings without revealing all the information they need to solve a problem, which is a tactic commonly used in practice by human tutors (Hume et al., 1996). All questions and reference solutions in our database refer to the material covered in the lecture videos, thus ensuring that the students taking courses on our platform are able to answer the questions based on the material covered. By providing students with the hints we aim to give them a nudge in the right direction: for instance, the first hint from Table 1 suggests that they should connect the idea of a *high bias* with the concepts of *underfitting* and *overfitting* and give an appropriate answer as a result. The hint provides them with partial information that can be used to give a correct answer to this question, without actually revealing the full answer: the solution verification module in this case would expect to see *high bias* connected to *underfitting* as one of the correct answers (the student may follow a different route in answering this question and talk about *variance* instead), yet the student may still incorrectly link *high bias* to *overfitting*.

It is important to make sure that hints provided by our system do not reveal full answers to the students. Below we describe the main categories of questions used on our platform and estimate a typical hint “transparency” for each category:

1. Around 20% of the questions on our platform ask students to provide *definitions*: for instance, “*What is gradient descent?*” is an example of such a question. A sample reference solution to such a question contains a single-sentence definition. This type of questions are, possibly, the easiest for the students to answer,

- and a special precaution is taken when generating hints for these types of questions in order not to reveal the full correct answer. A special case is represented by reference solutions of the form “*This is X*”, where *X* is a domain-specific concept that the question addresses. Such cases are handled by a different algorithm: the reference solution is used to generate a cloze test-style hint with the domain-specific concept “masked”. For instance, one of the hints generated is “*This is the irreducible X. You need to define what X is.*”, with *X* being *noise*.
2. A further 23% of the questions on our platform ask for both *a definition* (or *identification* of a domain-specific concept) *and an explanation or justification*: “*You are given a dataset of images of wildlife in Africa. You are tasked with building a model which can identify animals in the images. Is this a regression or classification problem? Explain why.*” is an example of such a question. A typical hint in this case will reveal some information needed for the correct answer (for example, “*Observe that each animal is a separate class.*”). To answer questions of this type satisfactorily, the student will need to connect the ideas from the hint with the relevant domain concepts, and additionally provide an explanation for the answer.
 3. Another 10% of the questions on the platform ask students to *contrast* domain concept and identify the difference. The question “*What is the difference between a closed-form solution and gradient descent?*”, as well as the first question in Table 1, are examples of such questions. In this case, a hint may reveal information related to one of the concepts (e.g., *high bias*), but the student will still need to link it to the correct domain concept and explain the difference between the concepts.
 4. Finally, 47% of the questions ask for an *explanation* or *elaboration*, as, for example, “*Why is linear regression a parametric model?*” does. In this case, a hint may reveal some aspects of the correct answer (e.g., properties of *linear regression* or of *parametric models*), but the student will still need to connect the ideas in order for their answer to be accepted as correct.

Personalized Hints Selection

Once hints are generated with the algorithm described above, they are evaluated based on their quality and appropriateness for each student. The appropriateness of this selection determines the quality of the personalized feedback provided to the student. Since the machine learning model applied here returns scores assigned to each hint, we can also produce a ranking order reflecting the appropriateness of each hint for each particular student. We employ a machine learning approach and utilize the Random Forest classifier from the `scikit-learn`³ suite (Breiman, 2001). The algorithm considers various sets of features, described below. The sets of features considered define the complexity of the feedback selection model and we show that the models get more complex in terms of personalization involved.

³<https://scikit-learn.org>

1. **BASELINE MODEL** relies on the use of **linguistic features**, which assess the quality of the hint or explanation from the linguistic perspective only. These features do not take into account personal aspects of the student–system interaction and only assess generated feedback (i.e., hint) in isolation. This set contains a total of 14 features that are aimed at capturing various aspects of the generated feedback, including its quality, grammaticality and appropriateness to the question. We describe these features below.
 - We measure the *length* of the hint in terms of the number of words. This feature helps the algorithm learn how comprehensive suggested feedback is. For example, it can be expected that in practice students will find very short hints not informative enough, while they might find extremely long ones confusing or overwhelming.
 - *Completeness of the parse tree* is measured using the proportion of sentences in the hint that contain a complete subject-verb structure: for instance, this feature would penalize incomplete sentences like “*Note that grow with the size of the dataset*”, which would be generated by the hint-generation algorithm described in “[Hint Generation](#)” using a combination of a discourse-based modification “*Note that*” and the partial hint “*grow with the size of the dataset*” extracted from an expectation after the keyphrase “*non-parametric models*” is eliminated. This feature helps the algorithm capture the grammaticality aspect of the hint – in practice, students are likely to find ungrammatical hints confusing.
 - *Perplexity score* is estimated for a binary language model built on the basis on the in-domain (*machine learning*) dataset crawled from Wikipedia (see “[Wikipedia-Based Explanations](#)”). This feature helps the algorithm assess the quality, fluency and grammaticality of generated feedback.
 - *Keyword overlap* and *topic overlap* between the hint and the question help assess the fit of generated feedback for the question: the more related feedback is to the question, the higher is the overlap between the two in terms of words and topics. Here, we define “topics” narrowly as the titles of the Wikipedia articles that contain possible definitions of the keywords and phrases (see “[Wikipedia-Based Explanations](#)” for more details on our Wikipedia-based approach). In practice, students are likely to find topically related feedback more helpful.
 - *Average uniqueness score* of the keywords in the hint is estimated as an average of the inverse-document frequencies of the keywords according to their use across reference solutions. This feature helps the algorithm estimate how informative a keyword or phrase is: the more frequently it occurs in reference solutions to various questions in our database, the less specific it is about any given question. An example of such generic keyword is *model*: as it is used widely across multiple reference solutions, addressing supervised as well as unsupervised models, regression as well as classification models, its relative contribution to any specific hint and its relative informativeness are low.

- *Ambiguity of the keywords* is further estimated as the number of senses associated with a word in WordNet,⁴ which we access via the NLTK interface.⁵ This feature, similarly to the *uniqueness score*, helps the algorithm capture informativeness of the hint derived from its keyword content.
 - Features based on the *proportion of lexical items of a certain type* (for instance, pronouns and named entities) are used as further proxies for specificity of the hint's content. A high number of pronouns used in the hint would make it less clear for students; similarly, the use of named entities in the hints should be minimized as these are rarely informative in the data science and machine learning domains.
2. SHALLOW PERSONALIZATION MODEL relies on the combination of **linguistic features** pertaining to the hint that are used by the BASELINE MODEL and **performance-based features**. Performance-based features are extracted from the data available on our platform and they take into account past student performance. In particular, they include the total number of questions presented by the ITS to the student, the number of all attempts as well as only the past attempts at answering the question, the proportion of correctly and incorrectly answered questions in total as well as at the particular point in the student–system dialogue, and the total length of the student–system dialogue interaction. As compared to the BASELINE MODEL, the SHALLOW PERSONALIZATION MODEL takes a more personalized approach. In particular, we believe that this set of the past student performance features helps the model capture student's strength and their knowledge of the subject to a considerable extent. With the addition of 8 performance-related features to the linguistic features described above, this model uses a total of 22 features.
 3. DISCOURSE PERSONALIZATION MODEL, in addition to the 22 features described above, takes into account the student's utterance immediately preceding the hint given and up to 4 previous interaction turns between the student and the system, thus considering up to 9 utterances from the student and the system in total. The number of the previous dialogue interaction turns to take into account was selected to maximize overall coverage of interactions that were available on our platform at the time of the experiments. The model then analyzes the set of 9 utterances from the linguistic point of view by taking the *proportion of keywords*, the *proportion of topics overlapping* between the question and each of the statements, and the *perplexity score* for each of the statements (features defined as above). Thus, this final model is the most expressive of all three, as it relies on 49 features in total and combines **linguistic features** pertaining to the hint (14 features described above), **performance-based features** (8 features described above), and **linguistic features** applied to the student–system interactions (3 types of linguistic features applied to 9 statements produced by the student or the system in the previous interaction turns).

⁴<https://wordnet.princeton.edu>

⁵<http://www.nltk.org>

Thus, our feedback selection models get increasingly more complex in terms of the amount of personalization involved – from no personalization in the BASELINE MODEL based on linguistic quality of the hint only, to the SHALLOW PERSONALIZATION MODEL that adds high-level, quantitative student performance metrics, to the DISCOURSE PERSONALIZATION MODEL that also takes into account dialogue-based interactions between the student and the platform.

The models are trained and evaluated on a collection of 450 previously recorded student–system interactions of up to 4 turns in length. These student–system interactions represent historical data extracted from our platform as they were recorded from an earlier version of the Korbit ITS, which selected the hints to show uniformly and randomly, i.e. without any consideration for the student performance or the hint quality. The models are trained in a binary classification setting to predict if a student with specific performance characteristics and given a specific hint will correctly solve the exercise in their next attempt. Once the model is trained on such historical data and learns to associate features from the feedback selection models with the success at solving the exercise based on the provided hint, it can be applied to select the most appropriate hints in practice (see “Pilot Study”).

Table 2 shows the results in terms of accuracy and F1 score calculated based on 50-fold cross-validation applied to the historical data. The RANDOM model, which does not apply any hint selection and simply provides a hint from the set of available hints at random, achieves an accuracy of $53.64\% \pm 3.99\%$ and an F1 score of $48.21\% \pm 3.63\%$. The BASELINE system that relies on the linguistic features only to select the best matching explanation reaches slightly higher performance. Taking individual performance measures into account brings considerable improvements in the results, with the SHALLOW PERSONALIZATION model achieving an accuracy of $68.75\% \pm 4.06\%$ and an F1 score of $62.23\% \pm 4.49\%$. The best performing model overall uses DISCOURSE PERSONALIZATION and achieves $86.71\% \pm 3.34\%$ accuracy and $84.81\% \pm 3.97\%$ F1 score, which are statistically significant improvements at a 95% confidence level over all other models. Therefore, we should expect the DISCOURSE PERSONALIZATION model to select the most appropriate personalized feedback in practice. We put this assumption to test in the user studies described in “Pilot Study”.

Table 2 Accuracy and F1 scores of different hint selection models (with 95% confidence intervals) calculated based on cross-validation with $k = 50$ folds

Model	Accuracy	F1-score
RANDOM	$53.64\% \pm 3.99\%$	$48.21\% \pm 3.63\%$
BASELINE (No Personalization)	$60.57\% \pm 4.45\%$	$54.90\% \pm 4.74\%$
SHALLOW PERSONALIZATION	$68.75\% \pm 4.06\%$	$62.23\% \pm 4.49\%$
DISCOURSE PERSONALIZATION	$86.71\% \pm 3.34\%$	$84.81\% \pm 3.97\%$

Best results are highlighted in bold; * indicates statistical significance compared to the baseline model at a 95% confidence level

Wikipedia-Based Explanations

Wikipedia-based explanations may provide alternative ways of helping students to understand and remember concepts more effectively. With over 6 million articles containing over 3.5 billion words, English Wikipedia provides extensive material for the NLP component of our system, that we attempt to leverage in this work. In addition, the hierarchical structure of the hyperlinks imposed by the Wikipedia format facilitates identification of the sets of pages related to the topic. Furthermore, the format adopted for Wikipedia articles themselves, where the first sentence typically provides the definition (or a high-quality explanation) of the title concept and the first paragraph presents a concise description of the topic (Kapugama et al., 2016), makes information extraction easier. Thus, we assume that by generating a large set of Wikipedia-based explanations for the subject domain (e.g. hundreds of explanations for each exercise in *Korbit*), a personalized feedback model may be able to target a larger set of student knowledge gaps and provide more effective help.

To generate Wikipedia-based explanations, we use a multi-stage generation pipeline. This pipeline is illustrated in Fig. 2. The major stages in the pipeline include:

1. Extracting keywords and keyphrases from questions and expectations (i.e. reference solutions)
2. Identifying all relevant Wikipedia articles related to the domain keywords and keyphrases
3. Extracting high-quality explanations and generating candidate explanations based on these keywords using relevant articles
4. Extracting features for candidate explanations classification
5. Evaluating candidate explanations with respect to their quality level
6. Selecting all relevant Wikipedia explanations

In the first stage, all relevant domain keywords and keyphrases are extracted from the reference questions and solutions by extracting noun phrases and pronouns using a procedure similar to the one presented in “[Hint Generation](#)”. We use spaCy for all steps that involve linguistic analysis. Next, using the identified domain keywords and

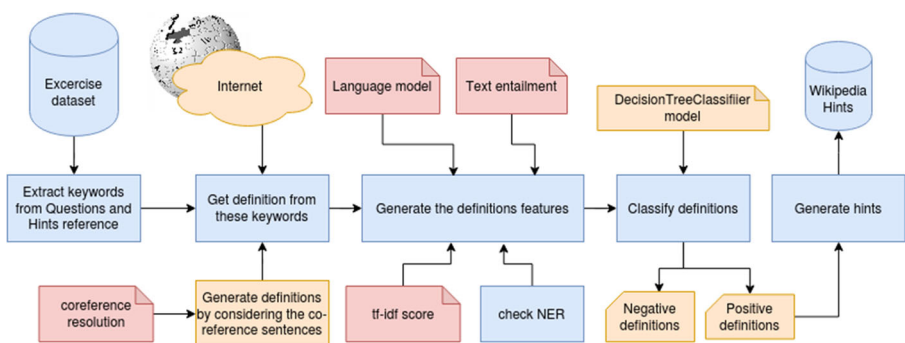


Fig. 2 The *Wikipedia explanations* multi-stage generation pipeline. “*Positive definitions*” refer to the high-quality explanations, while “*negative definitions*” are low-quality ones

Table 3 Examples of keywords and relevant articles identified by our method

Domains	Keywords	Relevant articles
“Machine Learning”	“Autonomous car”	“Glossary of artificial intelligence”
“Estimation theory”		“Microbotics”
“Deep learning”		“Autonomous things” “Self-driving car”
“Robotics”	“Linear regression”	“Statistical learning theory”
		“Deep learning”
		“Linear regression”
		“Pattern recognition”

keyphrases, we select relevant Wikipedia articles that are further used to extract and generate Wikipedia-based explanations. To help the algorithm identify all relevant articles, we disambiguate abbreviations frequently used for technical terms following Schwartz and Hearst (2003), and collect synonyms for the keywords using the WikiSynonyms API.⁶ Table 3 presents some examples of extracted keywords and relevant articles identified by our method.

This means that we can extract and generate explanations on, e.g., *self-driving car* and *autonomous things* for questions asking learners about *autonomous car*, since the articles on *self-driving car* and *autonomous things* are related to our key concept *autonomous car*.

In the next step, we create a set of *extracted Wikipedia-based explanations* and generate *candidate Wikipedia-based explanations*. Specifically, we extract the explanations from the first sentence in each article relying on the idea that in a typical Wikipedia article the first sentence provides the definition or a high-quality explanation of the title concept (Kapugama et al., 2016). The other candidate Wikipedia-based explanations are generated from the rest of the article, as described below. To ensure that the generated *candidate Wikipedia-based explanations* are clear and on-topic, we apply co-reference resolution to substitute pronouns with the key terms they represent using the implementation by Clark and Manning (2016). We use the pre-trained model⁷ that is reported to achieve an F1 score in the range of 65 – 75, depending on the data. We annotate the *extracted Wikipedia-based explanations* (i.e., the explanations derived from the first sentence in the Wikipedia article) to be “high quality” explanations, since they are normally highly relevant to the topic, grammatically correct, and describe the article topic clearly and succinctly. At the same time, since further *candidate Wikipedia-based explanations* are automatically generated from different parts of the Wikipedia article, one can assume that many of them would be of “low quality”: for instance, they may contain irrelevant, off-topic information or they may be grammatically incorrect. We create a training set of such explanations, marking them “low quality”. Next, in order to select the most appropriate *candidate Wikipedia-based explanations*, we train a binary classification model

⁶<https://rapidapi.com/ipeirotis/api/wikisynonyms>

⁷<https://github.com/clarkkev/deep-coref>

to classify an explanation as being either “high quality” or “low quality” based on its linguistic features.

By training a machine learning algorithm on such mostly high quality and mostly low quality examples, we aim to be able to identify high quality explanations among the wider set of examples relying on the idea that the algorithm learns the features of the high quality explanations and is, therefore, able to pick those not only among the extracted explanations, but also among the ones generated using our NLP pipeline. We experimented with a range of classification models and selected the best performing one based on our preliminary experiments. This best performing classification model is then used to select the set of best *candidate Wikipedia-based explanations*. In what follows, we refer to this set of explanations as the *generated Wikipedia-based explanations* since they were generated by our pipeline.

A 2GB in-domain (*machine learning*) dataset was crawled from Wikipedia and re-sampled using the SMOTE algorithm (Mathew et al., 2018) to tackle the over-sampling problem stemming from the fact that there are fewer “high quality” explanations (extracted from the first sentence of the Wikipedia articles) than “low quality” explanations (generated from the rest of the article using our pipeline): while the exact number of generated explanations per concept depends on the length of the Wikipedia article, on the average the number of generated explanations is 23.55 times higher than the number of extracted ones. In the binary classification setup, we treat the extracted explanations as a positive class, and the generated explanations as a negative class, and we aim to train an algorithm to distinguish between the two classes. This algorithm then is assumed to be able to select high quality Wikipedia explanations under the assumption that some of these may come from the generated examples.

We extracted a number of features using NLP techniques, including a range of length-based features (measuring the length of the extracted or generated definition in terms of the number of characters and in terms of the number of words, as well as the length of the title phrase of the Wikipedia article used for the explanation extraction or generation), co-reference resolution score (Clark & Manning, 2016), language model score for the model built using a state-of-the-art LSTM neural network (Merity et al., 2017),⁸ textual entailment-based relations using a state-of-the-art attention-based neural network (Parikh et al., 2016), TF-IDF scores, and named entity classes (Nothman et al., 2013). The dataset was split into 60% training, 10% validation, and 30% test subsets, and we experimented with a range of models, including Decision Tree classifiers (Breiman et al., 1984), Random Forests (Breiman, 2001), Logistic Regression (Bishop, 2006), and Support Vector Machines (Smola & Schölkopf, 2004), using the `scikit-learn` implementation. Among those, the Decision Tree classifier performed best, yielding an F1 score of 80.32% in distinguishing between high and low-quality explanations on the validation set. We therefore use this classifier to further identify high-quality explanations in the test set.

Table 4 shows an example of an explanation extracted by our algorithm from the first sentence of a Wikipedia article and an example of an automatically generated Wikipedia-based explanation that was detected as being high quality by our algorithm. The first sentence presents an explanation related to the question on the

⁸We use an adaptation of the model from <https://github.com/salesforce/awd-lstm-lm>

Table 4 Examples of Wikipedia-based explanations

Question	Wikipedia explanations	Label
How many human drivers would be needed to drive an autonomous car?	A self-driving car , also known as an autonomous vehicle (AV) connected and autonomous vehicle (CAV), driverless car, robot car, or robotic car, is a vehicle that is capable of sensing its environment and moving safely <i>with little or no human input</i> .	Extracted
	Different methods and levels of autonomy can be achieved through monitoring and remote control from a nearby manned ship, an onshore control or through artificial intelligence and machine learning, <i>letting the vessel itself decide the course of action</i> .	Generated

Identified keywords are marked with boxes, and information that helps guide a student is highlighted in italics

“autonomous cars” and is *extracted* from the Wikipedia article on “Self-driving car”, which is identified as relevant to the key term “autonomous cars” in the early steps of the pipeline. The second sentence is an explanation *generated* by our pipeline on the basis of the text available in the Wikipedia article on “Autonomous cargo ship”, which is also considered relevant by our algorithm. The bits of the explanations that are most relevant and should help students answer the question are highlighted in italics: for instance, the information that a self-driving car uses little or no human input and that an autonomous ship decides the course of action itself may nudge students in the right direction and help them answer how many human drivers are needed to drive an autonomous vehicle.

Pilot Study

This section presents the results obtained with the Korbit ITS using personalized feedback. In these experiments, we evaluate the personalized hints and Wikipedia-based explanations using a set of 796 annotated student–system interactions, collected from 183 students enrolled for free and studying the machine learning course on the Korbit learning platform remotely between January and February, 2020.

Students from around the world can sign up on the platform and need only provide their email address. This makes it difficult to accurately assess the student demographics. We use the Google Analytics tool to estimate the aggregate demographics of all the visitors of the Korbit learning platform website.⁹ Although this will also

⁹<https://analytics.google.com>

include visitors who did not sign up to study on the platform and participate in the study, we expect that the demographics estimated here will be largely representative of the 183 students in our study. Based on this, we estimate that $\sim 51\%$ of students come from Asia, $\sim 22\%$ of students come from North America, Central America or South America, $\sim 13\%$ of students come from Africa, $\sim 12.5\%$ of students come from Europe, and $\sim 1.5\%$ come from Oceania. Furthermore, we estimate that $\sim 70\%$ of students are male and that the majority of students are between 18 and 35 years old.

Personalized Hints

To evaluate the personalized hints, a hint is selected at uniform random from one of the personalized feedback selection models when a student gives an incorrect solution. Afterwards, the student *success rate* at solving exercises is measured as the proportion of instances where a student provides a correct solution after receiving a personalized hint. We believe that the student *success rate* estimated as their ability to answer the posed question correctly after being provided with a hint shows hint efficacy, and in the future experiments we also plan to measure student *learning gains* by testing their knowledge and understanding of the relevant concepts in delayed post-tests.

Since it is possible for the ITS to provide several pedagogical interventions for a given exercise, we separate the success rate observed in students for *all attempts* from those for students who received a personalized hint or explanation *before their second attempt* at the exercise. The correctness of the student answer on the platform is assessed by our automated student solution verification module. For the purposes of accurately measuring student performance outcomes in this experiment, all student solutions and their correctness status assigned by the automated solution verification module were double-checked by domain experts (members of the Korbit team). Human annotators agreed with the system's assessment in 80.53% of the cases; in other cases, human expert annotation of the student solution was used as the gold standard.

The results are given in Table 5. In line with the results from Table 2, the DISCOURSE PERSONALIZATION MODEL leads to the highest student success rate at 48.53% followed by the SHALLOW PERSONALIZATION MODEL at 46.51% and the BASELINE MODEL at 39.47% for all attempts. Furthermore, the difference between the success rate for the DISCOURSE PERSONALIZATION MODEL and BASELINE MODEL for the students before their second attempt is statistically significant at 95% confidence level based on a z-test ($p = 0.03$). These results strongly support the hypothesis that automatically generated personalized hints lead to substantial improvements in student performance outcomes.

Wikipedia-based Explanations

To evaluate the Wikipedia-based explanations, we conduct a second experiment. When the student gives an incorrect solution, the system shows two randomly-selected subject-related Wikipedia-based explanations (one extracted and one generated) and asks the student to select *the most helpful one*, or to select if *both are*

Table 5 Student success rates for personalized hints with 95% confidence intervals (C.I.)

Model	All attempts		Before second attempt	
	Mean	95% C.I.	Mean	95% C.I.
BASELINE (No Personalization)	39.47%	[24.04%, 56.61%]	37.93%	[20.69%, 57.74%]
SHALLOW PERSONALIZATION	46.51%	[31.18%, 62.34%]	51.43%	[33.99%, 68.62%]
DISCOURSE PERSONALIZATION	48.53%	[36.22%, 60.97%]	60.47%*	[44.41%, 75.02%]

After being shown a hint or explanation, their success rate was determined by whether they solved the exercise in their next attempt. Best results are highlighted in bold; * indicates statistical significance compared to baseline model at a 95% confidence level

equally helpful, or if *neither of them is helpful*. The system then asks the student to attempt the exercise again, based on which the student's success rate is measured. It should be noted that since the student receives two hints at once, the observed success rates are influenced by both hints shown.

The results are given in Table 6. As would be expected, students find the explanations extracted from the first sentences of the Wikipedia articles more helpful on average since such explanations usually are of high quality: they are selected as helpful 55.66% of the time, while the explanations automatically generated from the other parts of the Wikipedia articles are selected 44.44% of the time. However, when both types of explanations are shown, at least one of them is rated as helpful 83.33% of the time, meaning that the students find Wikipedia-based explanations unhelpful in 16.67% of the cases only. This difference in results between both types and each individual type is significant at a 95% confidence level. This suggests that, although generated explanations are perceived to be less helpful on average, students are far more likely to rate the feedback as overall helpful when both types of explanations are shown to them as compared to only showing extracted explanations. Lastly, as shown in Table 6, the student success rates appear to be highly similar

Table 6 Student preferences and success rates for Wikipedia-based explanations

Explanation	Student preference		Student success rates	
	Mean	95% C. I.	Mean	95% C. I.
Extracted	55.56%	[43.37%, 67.28%]	16.00%	[4.54%, 36.08%]
Generated	44.44%	[32.72%, 56.63%]	16.67%	[3.58%, 41.42%]
Extracted, Generated or Both Preferred	83.33%*	[72.70%, 91.08%]	17.65%	[6.76%, 34.53%]

Students were shown two explanations (an extracted one and a generated one) and asked which one they found most useful. Afterwards, their success rate was determined by whether they solved the exercise in their next attempt. Best results are highlighted in bold; * indicates statistical significance compared to all other explanation preference classes at a 95% confidence level

for both extracted and generated explanations, with no statistically significant difference between the two types. Taken together, these results support the hypothesis that generated Wikipedia-based explanations can provide helpful feedback.

At the same time, despite the fact that students find Wikipedia-based explanations helpful, the success rates for such explanations are overall quite low: 17.65% as compared to 60.47% for personalized hints. We believe that these results can be attributed to the following reasons: firstly, as the examples of the extracted and generated explanations from Table 4 demonstrate, Wikipedia-based explanations may, on the one hand, help guide a student in the right direction, but on the other hand, they may also be only broadly related to the questions on our platform. In other words, unlike hints that are generated from our reference solutions and are, therefore, adapted to the content covered by the questions, Wikipedia-based explanations may be less informative when a specific question is considered. Secondly, our primary goal in the experiments with Wikipedia-based explanations was to establish whether it is possible to leverage large amounts of material available on Wikipedia to generate useful explanations. We believe that our results are promising, but future experiments should investigate how to close the gap between the success rates achieved by our personalized hints and those achieved by Wikipedia-based explanations. We conclude that the results of our experiments support our assumption that personalization in pedagogical interventions is important, and future experiments with Wikipedia-based explanations will focus on personalization of this type of interventions.

Conclusions and Future Work

In this paper, we have proposed methods for automated generation and personalization of feedback in an intelligent tutoring system (ITS). In particular, we have focused on generation and personalization of text-based *hints*, and extraction and generation of *Wikipedia-based explanations* leveraging large amounts of potentially useful data available for learner needs on Wikipedia.

We generate each of these types of feedback in a fully automated manner, using data-driven approaches and state-of-the-art machine learning and natural language processing techniques, with the available input data being the only bottleneck for this approach. We have conducted several experiments investigating the utility of the personalized feedback, including measuring student success rates and student's subjective preferences for each type of feedback. The experiments strongly support our hypothesis that the personalized hints help to significantly improve student performance outcomes and that Wikipedia-based explanations can provide helpful feedback.

In this work, we have showed that personalized feedback automatically generated in a data-driven way leads to improved performance outcomes measured as the success rate in the students' ability to answer the questions on the material correctly after being provided with an informative hint. This is a crucial first step towards solving one of the major bottlenecks for large-scale ITS, which have often relied on expert design and hand-crafted rules in the past. Future work will investigate scalability and transferability of our personalized data-driven feedback models across multiple

domains. Specifically, we plan to conduct experiments with other STEM subjects and we believe that the approach developed and proposed in this paper can be transferred to learning material in other technical domains. In this work, we focus primarily on ITS, but we believe that this approach can be applied to other contexts and learning environments to help students (e.g., in self-paced and autodidactic learning), teachers (specifically in flipped and blended classrooms), and course designers.

Despite promising results achieved in this work, we acknowledge that this research is in early stages. One limitation of the current work is that we measure performance outcomes as the success rate in answering the question immediately after personalized feedback is provided. We believe that observed improvements are important as they show that the generated hints and explanations are helpful and guide students in the right direction. However, future experiments on our platform will address student learning gains and their ability to retain knowledge, which can be tested using delayed post-tests on the relevant concepts, as well as to perform near transfer (i.e., testing knowledge of the same concept in a similar context), and far transfer (i.e., testing knowledge of the same concept in a new context). Such future experiments will seek to further support usefulness of the automatically generated personalized feedback.

An additional challenge for ITS that teach technical subjects, such as machine learning, data science and artificial intelligence, lies in the combination of various modalities and the use of mixed language that are involved in generating the pedagogical interventions and the provision of feedback. It is important that an ITS in these domains must evaluate answers expressed in a purely textual form and provided by the students in response to the questions that are, likewise, expressed in a natural language (e.g., “*What is a linear regression model?*”). However, in addition, ITS focusing on technical domains must also handle other modalities, such as mathematical equations, chemical equations, source code, and so on. For example, an ITS teaching machine learning will often have to evaluate and provide feedback on mathematical expressions. On the one hand, such expressions may be included in student answers: e.g. a mathematical expression would be expected as a response to the question “*Define the sum-of-squares error function*” from an ITS. On the other hand, mathematical expressions may be included in the mixed-modality questions, which may further combine them with textual content, as does a question like “*Suppose the output is categorical with 10 categories ($y = 1, 2, \dots, 10$). If $y_i = 9$, then what would its corresponding one-hot vector representation be?*”. This proved to be particularly challenging in the past, with many systems aiming to provide feedback on mathematical expressions resorting to hand-crafted rules (Büdenbender et al., 2002; Gogvadze et al., 2005; Hennecke, 1999), or involving a human tutor (Cukurova et al., 2017; Hrastinski et al., 2019). In addition, as Benz Müller et al. (2007) and Dietrich and Buckley (2008) note, students’ responses using mixed language are often characterized by underspecification and ambiguity, with the latter being typical of both natural language and mathematical expressions.

Math equations are particularly challenging to evaluate and give feedback on because equivalent mathematical expressions can have different string representations. Moreover, the notation between different students may vary, and the notation itself can be ambiguous (Dietrich & Buckley, 2008). For example, the equation

“ $y(x + 5)$ ” has two interpretations, as shown in Fig. 3: y could be a function or a term multiplied by $x + 5$. Our ongoing research is concerned with the models capable of analyzing math equations in addition to purely text-based content and providing relevant feedback. Preliminary results show that our data-driven mathematical hints provide students with useful insights. In the future, we also plan to expand the set of hints with those on programming exercises and investigate students’ performance outcomes from the feedback that complements textual hints with mathematical equations and code snippets or instructions relevant for the specific taught concepts.

We have showed that students find generated hints and Wikipedia-based explanations helpful. Future work should also investigate how and what types of hints and explanations may improve student performance outcomes, as well as their interplay with student learning profiles and knowledge gaps. In particular, we plan to investigate how varying hint complexity and the level of hint transparency can be used in instructional scaffolding. In addition, we will explore how large amounts of available learning material can be leveraged to generate further pedagogical interventions in a data-driven way.

Of particular importance for the future work is development of models capable of explanatory formative feedback. Such models can be applied both to mathematical hints, providing students with further insights as to why their equations may be incorrect, and to textual hints and explanations, identifying what is missing or what is conceptually incorrect in the given answer and providing students with the guidance towards fixing the missing or incorrect ideas in their answers. Future work should also investigate the interplay between the granularity of such formative feedback and various student learning profiles.

Finally, it should be noted that there has been a massive increase in the use of ITS, and more broadly online learning platforms, separate from and alongside traditional human teacher–student interactions (for example, in flipped classrooms and blended learning environments). Therefore, it is important that future research looks closely into such aspects of the learning process as student motivation, engagement and managing of students’ emotional states. Of particular interest are such questions as whether tutoring via an ITS should mimic human tutoring or rather provide students with an alternative means of learning, and which aspects of the learning process are best addressed with an ITS tutor versus a human one.

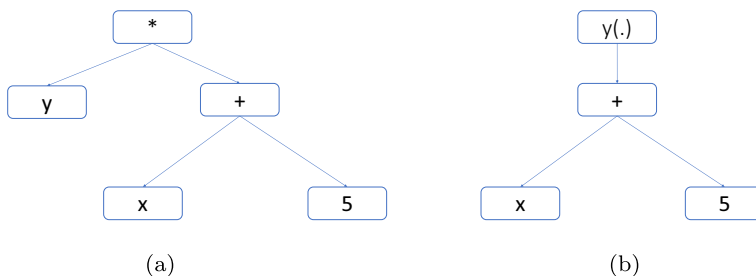


Fig. 3 Two interpretations of the equation “ $y(x + 5)$ ”

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abel, A., Chang, B.-Y. E., & Pfenning, F. (2001). Human-readable machine-verifiable proofs for teaching constructive logic. *PTP, I*.
- Ahn, J.-W., Chang, M., Watson, P., Tejjwani, R., Sundararajan, S., Abuelsaad, T., & Prabhu, S. (2018). Adaptive Visual Dialog for Intelligent Tutoring Systems. In *International Conference on Artificial Intelligence in Education* (pp. 413–418). Springer.
- Albacete, P., Jordan, P., Katz, S., Chounta, I.-A., & McLaren, B.M. (2019). The Impact of Student Model Updates on Contingent Scaffolding in a Natural-Language Tutoring System. In *International Conference on Artificial Intelligence in Education* (pp. 37–47). Springer.
- Aleven, V., Popescu, O., & Koedinger, K.R. (2001). Towards tutorial dialog to support self-explanation: Adding natural language understanding to a cognitive tutor. In *Proceedings of Artificial Intelligence in Education* (pp. 246–255). Citeseer.
- Anania, J. (1983). The Influence of Instructional Conditions on Student Learning and Achievement. *Evaluation in Education: An International Review Series*, 7(1), 3–76.
- Anderson, J. R., Boyle, C. F., & Reiser, B.J. (1985). Intelligent tutoring systems. *Science*, 228(4698), 456–462.
- Andrews, P. B., Brown, C. E., Pfenning, F., Bishop, M., Issar, S., & Xi, H. (2004). Etps: A system to help students write formal proofs. *Journal of Automated Reasoning*, 32(1), 75–92.
- Baker, R. S. (2016). Stupid tutoring systems, intelligent humans. *International Journal of Artificial Intelligence in Education*, 26(2), 600–614.
- Benzmüller, C., Horacek, H., Kruijff-Korabayova, I., Pinkal, M., Siekmann, J., & Wolska, M. (2007). Natural language dialog with a tutor system for mathematical proofs. In *Cognitive Systems* (pp. 1–14). Springer.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Berlin: Springer. <http://research.microsoft.com/en-us/um/people/cmbishop/prml/>.
- Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher*, 13(6), 4–16.
- Boaler, J., & Brodie, K. (2004). The importance, nature, and impact of teacher questions. In *Proceedings of the twenty-sixth annual meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education*, (Vol. 2 pp. 774–782).
- Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C.J. (1984). *Classification and Regression Trees*. Monterey: Wadsworth and Brooks.
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>.
- Brunskill, E., Mu, T., Goel, K., & Bragg, J. (2018). Automatic Curriculum Generation Applied to Teaching Novices a Short Bach Piano Segment. In *NeurIPS Demonstrations*.
- Büdenbender, J., Frischauf, A., Gogvadze, G., Melis, E., Libbrecht, P., & Ullrich, C. (2002). Using computer algebra systems as cognitive tools. In *International Conference on Intelligent Tutoring Systems* (pp. 802–810). Springer.
- Burke, A. J. (1983). *Students' potential for learning contrasted under tutorial and group approaches to instruction*. Ph.D. Thesis, University of Chicago, Joseph Regenstein Library, Department of Photoduplication.
- Burstall, R. (1998). Teaching people to write proofs: a tool. In *CafeOBJ Symposium, Numazu, Japan*.
- Chi, M., Koedinger, K., Gordon, G., Jordan, P., & Vanlehn, K. (2011). Instructional Factors Analysis: A Cognitive Model For Multiple Instructional Interventions. In *EDM 2011 - Proceedings of the 4th International Conference on Educational Data Mining* (pp. 61–70).

- Clark, K., & Manning, C. (2016). Deep Reinforcement Learning for Mention-Ranking Coreference Models. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2256–2262.
- Cukurova, M., Mavrikis, M., Luckin, R., Clark, J., & Crawford, C. (2017). Interaction analysis in online maths human tutoring: The case of third space learning. In *international conference on artificial intelligence in education* (pp. 636–643). Springer.
- D'Agostino, M., & Endriss, U. (1998). Winke: A proof assistant for teaching logic. In *Proceedings of the First International Workshop on Labelled Deduction*, Vol. 1998. Citeseer.
- De Medio, C., Gasparetti, F., Limongelli, C., Sciarrone, F., & Temperini, M. (2016). Automatic Extraction of Prerequisites Among Learning Objects Using Wikipedia-based Content Analysis. In *International conference on intelligent tutoring systems* (pp. 375–381). Springer.
- Dietrich, D., & Buckley, M. (2008). Verification of human-level proof steps in mathematics education. *Teaching Mathematics and Computer Science*, 6(2), 345–362.
- Dinan, E., Roller, S., Shuster, K., Fan, A., Auli, M., & Weston, J. (2018). Wizard of wikipedia: Knowledge-powered conversational agents. arXiv:1811.01241.
- Folsom-Kovarik, J. T., Schatz, S., & Nicholson, D. (2010). Plan ahead: Pricing ITS learner models. In *Proceedings of the 19th Behavior Representation in Modeling & Simulation (BRIMS) Conference* (pp. 47–54).
- Goguadze, G., Palomo, A. G., & Melis, E. (2005). Interactivity of Exercises in ActiveMath. In *ICCE* (pp. 109–115).
- Graesser, A. C., Cai, Z., Morgan, B., & Wang, L. (2017). Assessment with computer agents that engage in conversational dialogues and trialogues with learners. *Computers in Human Behavior*, 76, 607–616. <https://doi.org/https://doi.org/10.1016/j.chb.2017.03.041>, <http://www.sciencedirect.com/science/article/pii/S074756321730198X>.
- Graesser, A. C., Chipman, P., Haynes, B. C., & Olney, A. (2005). AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, 48(4), 612–618.
- Graesser, A. C., & Person, N. K. (1994). Question asking during tutoring. *American Educational Research Journal*, 31(1), 104–137.
- Graesser, A. C., Person, N. K., & Magliano, J.P. (1995). Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology*, 9(6), 495–522.
- Graesser, A. C., VanLehn, K., Rosé, C. P., Jordan, P. W., & Harter, D. (2001). Intelligent tutoring systems with conversational dialogue. *AI Magazine*, 22(4), 39–39.
- Guo, Q., Kulkarni, C., Kittur, A., Bigham, J. P., & Brunskill, E. (2016). Questimator: Generating knowledge assessments for arbitrary topics. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16)*. AAAI Press.
- Hendriks, M., Kaliszyk, C., Van Raamsdonk, F., & Wiedijk, F. (2010). Teaching logic using a state-of-the-art proof assistant. *Acta Didactica Napocensia*, 3(2), 35–48.
- Hennecke, M. (1999). *Online Diagnose in intelligenten mathematischen Lehr-Lern-Systemen*. VDI-Verlag.
- Holstein, K., McLaren, B. M., & Aleven, V. (2017). Intelligent tutors as teachers' aides: Exploring teacher needs for real-time analytics in blended classrooms. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference* (pp. 257–266).
- Holstein, K., McLaren, B. M., & Aleven, V. (2019). Designing for complementarity: Teacher and student needs for orchestration support in AI-enhanced classrooms. In *International Conference on Artificial Intelligence in Education* (pp. 157–171). Springer.
- Hone, K., & El Said, G. (2016). Exploring the factors affecting MOOC retention: A survey study. *Computers & Education*, 98, 157–168. <https://doi.org/10.1016/j.compedu.2016.03.016>.
- Hrastinski, S., Stenbom, S., Benjaminsson, S., & Jansson, M. (2019). Identifying and exploring the effects of different types of tutor questions in individual online synchronous tutoring in mathematics. *Interactive Learning Environments*, 0(0), 1–13. <https://doi.org/10.1080/10494820.2019.1583674>.
- Hume, G., Michael, J., Rovick, A., & Evens, M. (1996). Hinting as a tactic in one-on-one tutoring. *The Journal of the Learning Sciences*, 5(1), 23–47.
- Hume, G. D., Michael, J. A., Rovick, A. A., & Evens, M.W. (1993). The use of hints as a tutorial tactic. In *Proceedings of the 15th Annual Conference of the Cognitive Science Society* (pp. 563–568).
- Jiang, Y. (2014). Exploring teacher questioning as a formative assessment strategy. *RELC Journal*, 45(3), 287–304.


- Kapugama, K. D. C. G., Lorensuhewa, S. A. S., & Kalyani, M.A.L. (2016). Enhancing Wikipedia search results using Text Mining. In *2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer)* (pp. 168–175).
- Koedinger, K. R., & Anderson, J. R. (1993). Reifying implicit planning in geometry: Guidelines for model-based intelligent tutoring system design. *Computers as cognitive tools*, 15–46.
- Kulik, J. A., & Fletcher, J. D. (2016). Effectiveness of intelligent tutoring systems: a meta-analytic review. *Review of Educational Research*, 86(1), 42–78.
- Lahti, L. (2009). Guided generation of pedagogical concept maps from the Wikipedia. In *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education* (pp. 1741–1750). Association for the Advancement of Computing in Education (AACE).
- Leelawong, K., & Biswas, G. (2008). Designing learning by teaching agents: The Betty's Brain system. *International Journal of Artificial Intelligence in Education*, 18(3), 181–208.
- Lin, C. F., Yeh, Y.-C., Hung, Y. H., & Chang, R.I. (2013). Data mining for providing a personalized learning path in creativity: An application of decision trees. *Computers & Education*, 68, 199–210. <https://doi.org/10.1016/j.compedu.2013.05.009>, <http://www.sciencedirect.com/science/article/pii/S0360131513001309>.
- Liu, M., Calvo, R. A., Aditomo, A., & Pizzato, L.A. (2012). Using Wikipedia and Conceptual Graph Structures to Generate Questions for Academic Writing Support. *IEEE Transactions on Learning Technologies*, 5(3), 251–263.
- Liu, M., Calvo, R. A., & Rus, V. (2012). G-Asks: An intelligent automatic question generation system for academic writing support. *Dialogue & Discourse*, 3(2), 101–124.
- Makatchev, M., Jordan, P. W., Pappuswamy, U., & VanLehn, K. (2011). Representation and reasoning for deeper natural language understanding in a physics tutoring system. *AAAI*.
- Mathew, J., Pang, C. K., Luo, M., & Leong, W.H. (2018). Classification of Imbalanced Data by Oversampling in Kernel Space of Support Vector Machines. *IEEE Transactions on Neural Networks and Learning Systems*, 29(9), 4065–4076. <https://doi.org/10.1109/TNNLS.2017.2751612>.
- McBroom, J., Koprinska, I., & Yacef, K. (2019). A survey of automated programming hint generation—the hints framework. arXiv:1908.11566.
- Melis, E., & Siekmann, J. (2004). ActiveMath: An Intelligent Tutoring System for Mathematics. In L. Rutkowski, J. H. Siekmann, R. Tadeusiewicz, & L. A. Zadeh (Eds.) *Artificial Intelligence and Soft Computing - ICAISC 2004* (pp. 91–101). Berlin: Springer.
- Merity, S., Keskar, N. S., & Socher, R. (2017). Regularizing and Optimizing LSTM Language Models. arXiv:1708.02182.
- Mills, S. R., Rice, C. T., Berliner, D. C., & Rosseau, E.W. (1980). The correspondence between teacher questions and student answers in classroom discourse. *The Journal of Experimental Education*, 48(3), 194–204.
- Munshi, A., & Biswas, G. (2019). Personalization in OELEs: Developing a Data-Driven Framework to Model and Scaffold SRL Processes. In *International Conference on Artificial Intelligence in Education* (pp. 354–358). Springer.
- Nothman, J., Ringland, N., Radford, W., Murphy, T., & Curran, J. (2013). Learning multilingual named entity recognition from Wikipedia. *Artificial Intelligence*, 194, 151–175. <https://doi.org/10.1016/j.artint.2012.03.006>.
- 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.
- Olney, A. M., & Cade, W. L. (2015). Authoring intelligent tutoring systems using human computation: designing for intrinsic motivation. In *International conference on augmented cognition* (pp. 628–639). Springer.
- Parikh, A., Täckström, O., Das, D., & Uszkoreit, J. (2016). A Decomposable Attention Model for Natural Language Inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 2249–2255). Austin: Association for Computational Linguistics. <https://www.aclweb.org/anthology/D16-1244>.
- Passier, H., & Jeuring, J. T. (2006). *Feedback in an interactive equation solver*. UU WINFI Informatica en Informatiekunde.
- Person, N. K., Graesser, A. C., Harter, D., Mathews, E., Group, T. R., et al. (2000). Dialog move generation and conversation management in autotutor. In *Building Dialog Systems for Tutorial Applications-Papers from the AAAI Fall Symposium* (pp. 45–51).

- Price, T. W., Dong, Y., Zhi, R., Paaßen, B., Lytle, N., Cateté, V., & Barnes, T. (2019). A comparison of the quality of data-driven programming hint generation algorithms. *International Journal of Artificial Intelligence in Education*, 29(3), 368–395.
- Ram, A. (1991). A theory of questions and question asking. *Journal of the Learning Sciences*, 1(3–4), 273–318.
- Ramírez-Noriega, A., Juárez-Ramírez, R., Jiménez, S., Martínez-Ramírez, Y., & Figueroa Pérez, J. (2018). Determination of the course sequencing to intelligent tutoring systems using an ontology and Wikipedia. *Journal of Intelligent & Fuzzy Systems*, 34(5), 3177–3185.
- Rivers, K. (2017). Automated data-driven hint generation for learning programming.
- Rus, V., Stefanescu, D., Baggett, W., Niraula, N., Franceschetti, D., & Graesser, A.C. (2014a). Macro-adaptation in conversational intelligent tutoring matters. In *International Conference on Intelligent Tutoring Systems* (pp. 242–247). Springer.
- Rus, V., Stefanescu, D., Niraula, N., & Graesser, A.C. (2014b). DeepTutor: towards macro-and micro-adaptive conversational intelligent tutoring at scale. In *Proceedings of the first ACM conference on Learning@ Scale conference* (pp. 209–210).
- Scheines, R., & Sieg, W. (1994). Computer environments for proof construction. *Interactive Learning Environments*, 4(2), 159–169.
- Schwartz, A., & Hearst, M. (2003). A Simple Algorithm for Identifying Abbreviation Definitions in Biomedical Text. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 4, 451–62. <https://doi.org/10.1142/9789812776303.0042>.
- Shah, R., Shah, D., & Kurup, L. (2017). Automatic question generation for intelligent tutoring systems. In *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)* (pp. 127–132). IEEE.
- Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, 14(3), 199–222. <https://doi.org/10.1023/B:STCO.0000035301.49549.88>.
- Sommer, R., & Nuckols, G. (2004). A proof environment for teaching mathematics. *Journal of Automated Reasoning*, 32(3), 227–258.
- Stamper, J. C., Eagle, M., Barnes, T., & Croy, M. (2013). Experimental Evaluation of Automatic Hint Generation for Logic Tutor. *International Journal of Artificial Intelligence in Education*, 22(1–2), 3–17.
- Sufri, B., & Bornat, R. (1996). *User interfaces for generic proof assistants part i: Interpreting gestures*. York: Proceedings of User Interfaces for Theorem Provers (UITP-06).
- Talukdar, P. P., & Cohen, W. W. (2012). Crowdsourced Comprehension: Predicting Prerequisite Structure in Wikipedia. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP* (pp. 307–315). Association for Computational Linguistics.
- Tamura, Y., Takase, Y., Hayashi, Y., & Nakano, Y.I. (2015). Generating quizzes for history learning based on Wikipedia articles. In *International Conference on Learning and Collaboration Technologies* (pp. 337–346). Springer.
- Van de Pol, J., Volman, M., & Beishuizen, J. (2010). Scaffolding in teacher–student interaction: A decade of research. *Educational Psychology Review*, 22(3), 271–296.
- VanLehn, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist*, 46(4), 197–221. <https://doi.org/10.1080/00461520.2011.611369>.
- Ventura, M., Chang, M., Foltz, P., Mukhi, N., Yarbro, J., Salverda, A. P., Behrens, J., Ahn, J. w., Ma, T., Dhamecha, T. I., & et al. (2018). Preliminary evaluations of a dialogue-based digital tutor. In *International Conference on Artificial Intelligence in Education* (pp. 480–483). Springer.
- Webb, N. M. (1989). Peer interaction and learning in small groups. *International Journal of Educational Research*, 13(1), 21–39.
- Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*. Los Altos: Morgan Kaufmann.
- Willis, A., Davis, G., Ruan, S., Manoharan, L., Landay, J., & Brunskill, E. (2019). Key Phrase Extraction for Generating Educational Question-Answer Pairs. In *Proceedings of the Sixth (2019) ACM Conference on Learning@ Scale* (pp. 1–10).
- Wood, D. (2003). The Why? What? When? and How? of Tutoring: The Development of Helping and Tutoring Skills in Children. *Literacy Teaching and Learning*, 7, 1–30.
- Wu, L., & Looi, C.-K. (2010). Agent Prompts: Scaffolding Students for Productive Reflection in an Intelligent Learning Environment. In V. Alevén, J. Kay, & J. Mostow (Eds.) *Intelligent Tutoring Systems* (pp. 426–428). Berlin: Springer.

- Zhang, L., & VanLehn, K. (2016). How do machine-generated questions compare to human-generated questions?. *Research and practice in technology enhanced learning*, 11(1), 1–28.
- Zhang, L., & VanLehn, K. (2017). Adaptively selecting biology questions generated from a semantic network. *Interactive Learning Environments*, 25(7), 828–846.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Ekaterina Kochmar^{1,2}  · Dung Do Vu^{1,3} · Robert Belfer¹ · Varun Gupta¹ · Iulian Vlad Serban¹ · Joelle Pineau^{1,4}

¹ Korbit Technologies Inc., Quebec, Canada

² University of Bath, Bath, England

³ École de Technologie Supérieure, Quebec, Canada

⁴ McGill University & MILA (Quebec Artificial Intelligence Institute), Quebec, Canada