Exploring Automated Question Answering Methods for Teaching Assistance

Brian Zylich, Adam Viola, Brokk Toggerson, Lara Al-Hariri, and Andrew Lan(✉)

University of Massachusetts Amherst, Amherst, USA
andrewlan@cs.umass.edu

Abstract. One important aspect of learning is through verbal interactions with teachers or teaching assistants (TAs), which requires significant effort and puts a heavy burden on teachers. Artificial intelligence has the potential to reduce their burden by automatically addressing the routine part of this interaction, which will free them up to focus on more important aspects of learning. We explore the use of automated question answering methods to power virtual TAs in online course discussion forums, which are heavily relied on during the COVID-19 pandemic as classes transition online. First, we focus on answering frequent and repetitive logistical questions and adopt a question answering framework that consists of two steps: retrieving relevant documents from a repository and extracting answers from retrieved documents. The document repository consists of course materials that contain information on course logistics, e.g., the syllabus, lecture slides, course emails, and prior discussion forum posts. This question answering framework can help virtual TAs decide whether a question is answerable and how to answer it. Second, we analyze the timing of student posts in discussion threads and develop a classifier to predict the timing of follow-up posts. This classifier can help virtual TAs decide whether to respond to a question and when to do so. We conduct experiments on data collected from an introductory physics course and discuss both the utility and limitations of our approach.

1 Introduction

Learning happens in many forms, including learning through self-regulated studies and learning through verbal interactions with a teacher. The latter is especially effective for problem solving [1,2] and when students experience negative emotions [3]. However, interacting with students individually requires a lot of effort from teachers, or sometimes teaching assistants (TAs), especially in large-scale educational settings such as online courses [4]. Teachers and TAs often face numerous tasks including reviewing the curriculum, teaching, creating and

B. Zylich—This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1938059.

© Springer Nature Switzerland AG 2020
I. I. Bittencourt et al. (Eds.): AIED 2020, LNAI 12163, pp. 610–622, 2020.
https://doi.org/10.1007/978-3-030-52237-7_49

grading assignments and exams, and answering questions in an online course discussion forum. These tasks not only put a heavy burden on teachers and TAs but also result in a slow and insufficient feedback cycle for students.

One solution to this problem is to use automated pedagogical agents driven by artificial intelligence (AI) to scale up teacher effort and interact with many students at the same time [5–8]. For example, using AI-driven *virtual TAs* in online course discussion forums such as Piazza has enjoyed some success [9–12]. However, there are many limitations to current virtual TAs and significant advances in AI must be made before they can become a reality. Below, we outline three major requirements virtual TAs have to satisfy:

- They have to be *comprehensive* and must decide whether to automatically answer a student question or to defer it to humans. This decision can be made by searching course content (e.g. textbooks, lecture notes, and supplementary materials) [13] for relevant information and evaluating their confidence in understanding the question and providing a satisfactory answer.
- They have to be *context-aware* and should decide on the best timing of an automated intervention. For duplicate questions (studied in [10]) and questions that students can discuss among themselves and resolve, virtual TAs should decide not to intervene since discussions facilitate engagement and peer learning [14]. For questions where misconceptions are formed among student responses, virtual TAs should decide to intervene immediately to clear up these misconceptions. This decision can be made by analyzing both interactions among students and topics in each discussion forum thread [15].
- They have to be *conversational* and should engage in meaningful conversations with students, such as asking a follow-up question when a student question is unclear or offering words of encouragement [6]. This requirement has been the focus of numerous existing works [5–8,16–18].

1.1 Contributions

In this paper, we explore the use of automated *question answering* methods in natural language processing to power virtual TAs in online course discussion forums. We focus on the first two requirements outlined above and restrict ourselves to studying frequent and repetitive logistical questions; our goal is to reduce the burden on teachers and TAs by automating the routine part of teacher-student interaction. First, we adopt an open-domain question answering framework [19] that consists of two steps: *retrieving* relevant documents from a pool of course materials (including the syllabus, lecture slides, announcement emails, and previous discussion forum posts), and *extracting* an answer to the question from retrieved documents via automated, neural network-based methods [20]. We also use an *answerability classifier* to decide whether a question is answerable given information from the document pool. Second, we analyze the content and timing of student posts and develop classifiers based on multi-class and ordinal classification to predict the timing of follow-up posts in a discussion thread. These classifiers can be used to identify threads where student questions

are not likely to be resolved in time and thus require immediate intervention. We evaluate our system using a Piazza dataset collected from an introductory physics course. Quantitatively, we compare our system to both humans and IBM Watson on question answering performance. Qualitatively, we use several examples to illustrate where the model excels and where it falls short and discuss future advances needed for virtual TAs to become a reality.

1.2 Connections to Existing Work

Existing work on developing automated virtual TAs in course discussion forums is limited; the most relevant work is Jill Watson [9]. It requires a list of common question-answer pairs in a course; for every incoming question, it searches over all questions in the list and finds the most similar one, before deploying the corresponding answer if the similarity passes a certain threshold. The question-answer list can either be hand-crafted by teachers and TAs, which requires significant human effort, or come from discussion forum data collected in previous offerings of the same course, which is not applicable when the course is offered for the first time. On the contrary, our approach automatically retrieves relevant information from course materials and does not require human effort or prior data.

Another relevant recent work is the Curio SmartChat system [21] for question answering in self-paced middle school science courses. For an incoming student question, Curio Smartchat searches the content repository and either i) answers directly when the question is well-understood and can be matched to a question in the question-answer list, ii) recommends relevant content to the student if the question is not well-understood, or iii) responds with "small-talk" when there is no relevant content. Similarly, the Discussion-Bot system [11] retrieves relevant course documents and prior discussion posts for each discussion forum question and presents them to students after a series of rules-based post-processing steps. Despite similarities in document retrieval methods, our approach employs the additional step of automatic question answering using the retrieved documents to provide concise answers to student questions.

Most open-domain question answering systems, starting with DrQA [19], use document retrieval and answer extraction/ranking in a two-step approach. Term frequency-inverse document frequency (tf-idf) is an efficient way to retrieve documents based on their word overlap with a query, considering frequency in the document and the entire corpus. DrQA uses a tf-idf [22] look-up to retrieve documents and a recurrent neural network to extract answers from the retrieved documents. The work in [23] uses a ranking module that is jointly trained with the answer extraction module using reinforcement learning. The work in [24] explores how to better extract relevant information from documents before extracting an answer from retrieved information. The work in [25] links the document retriever with the question answering module by iteratively retrieving documents and updating the query accordingly. These systems are trained and evaluated on standard question answering datasets with highly structured text. In contrast, our goal is to explore their effectiveness on questions asked by students in online course discussion forums, which are often ill-posed or poorly written [21].

For the study of post timing in online (not necessarily course) discussion forums, relevant existing works include [26] which predicts which posts are helpful to answering a question post, [27] which predicts whether the user asking a question will accept an answer post, and [28] which predicts the timing of posts using point processes parameterized by neural networks. Instead, we use multi-class and ordinal classification for timing prediction since our dataset is not large enough for point process-based methods.

2 Methodology

We now detail our logistical question answering system. Code implementing these methods will be made available at https://github.com/bzylich/qa-for-ta.

2.1 Question Answering Framework

Our question answering framework builds on DrQA [19]. First, for each question, we retrieve a set of relevant documents using a document retriever. Then, we extract and rank short answers from each document using automated question answering methods. Finally, we add an answerability classifier to determine the probability that we are able to provide a satisfactory answer to the question.

Before retrieving documents, we split each document into paragraphs and merge short paragraphs together if they do not exceed 220 characters in total. Documents are then retrieved by calculating the inner product between the tf-idf [22,29] vector for a question and the tf-idf vectors for each document. We select the 5 documents with the highest scores as relevant sources of information for question answering. We compare variants of this retrieval approach in Sect. 3.

Following common question answering methods, we encode the text from a retrieved document and the question into a low-dimensional representation using a recurrent neural network (RNN) trained on the SQuAD [20] question answering dataset[1]. Then, we use another RNN to decode the start and end indices corresponding to the span of text in the retrieved document that best answers the question[2]. In this manner, we produce 5 candidate answers, one from each document, and rank them using their tf-idf document retrieval scores.

The document pool contains documents published by the instructor, TAs, and even other students that contain information about the course. These documents include the course syllabus, announcement emails from instructors, class notes, practice problems, the course textbook, and previous student posts on discussion forums. However, some student questions may not be answerable given the current document pool at a point in time. For example, students may ask about the time and location of the final exam during the first week of the semester

[1] SQuAD consists of Wikipedia articles and crowdsourced questions with answers.
[2] We extract a span of text from the document rather than generating an open-ended answer since tools for the latter are still unreliable [30,31].

when final exams are not yet set by the university. Therefore, we need an answerability classifier that can determine the probability that a question is answerable given a document in the available document pool. To train this classifier, we use the *bert-base-uncased* variant of BERT [32]. BERT is a state-of-the-art pre-trained neural language model that has been used to improve performance on a variety of downstream natural language processing tasks. We fine-tune BERT using adapters [33] with size 256 on the SQuAD 2.0 dataset [34] and the Natural Questions dataset [35] that contain human-generated answerability labels for question-document pairs. We then apply this trained classifier to filter out retrieved documents that are similar to the question but cannot be used to answer it. Specifically, we pass each question and each retrieved document separately to the answerability classifier; If the answerability classifier indicates that the question is answerable given the document with high probability, we keep that document; otherwise, it is discarded. This answerability classifier helps us to i) only answer a question when we are confident in providing a satisfactory answer and ii) improve the quality of the document retriever.

2.2 Post Timing Prediction

Automated question answering systems can potentially answer many questions immediately after they are posted, which will be effective in reducing human effort for straightforward logistical questions. However, for knowledge-related questions, there exists a clear connection between balanced collaboration in problem solving and effective student learning [1]. If the system always answers questions immediately, it will stifle useful discussion between students that promotes peer learning. Thus, there is a need for the system to decide when to automatically answer a question by predicting the length of time until the next post in a thread.

Given the sequential nature of discussion threads, we employ an RNN to predict the time until the next post of a particular thread, where each discrete time step of the network corresponds to a post. At each time step, our input to the RNN is a vector concatenating the following information: the textual embedding of the current post (generated using BERT), the time between the previous and current post, and a one-hot encoded representation of the Piazza post type. For the output at each time step, it is difficult to formulate time prediction as a regression problem since time between posts ranges from seconds to days or even infinity for the end-of-thread (EOT) post. Therefore, we formulate it as a k-class classification problem where the first $k-1$ classes are time intervals, e.g. [5 mins, 1 hr), and the final class is for EOT posts.

We use two different loss functions to train the RNN. The first loss function, cross entropy, uses the softmax function and assumes the classes are not ordered:

$$L = -\log(e^{y_g} / \textstyle\sum_i e^{y_i}).$$

In this case, the output at each time step is a length-k vector that is used to predict the probability for each output class and y_i denotes its ith entry.

g corresponds to the actual time bin the post belongs to. However, given the ordered nature of the time intervals, we also use another loss function which is a threshold-based generalization of the hinge loss [36]:

$$L = \sum_{i=1}^{k-1} \max\{0, 1 - s(i,g)(\theta_i - y)\}, \quad s(i,g) = \begin{cases} -1, & i < g \\ 1, & i \geq g \end{cases}.$$

In this case, the output at each time step is a scalar y; $-\infty = \theta_0 < \theta_1 < ... < \theta_{k-1} < \theta_k = \infty$ denotes a set of thresholds that partition the possible values of the RNN scalar output into k bins. In other words, this loss compares the scalar output with each threshold to put it into a bin, penalizing outputs farther from their actual time bin.

3 Experiments

3.1 Dataset

Our dataset, which we dub PhysicsForum, consists of 2004 posts in 663 threads in the online discussion forum of an introductory physics course. 802 of these posts are questions asked by students (640 primary questions and 162 follow-up questions). We manually divide these questions into different types: 140 conceptual (regarding a specific concept), 250 reasoning (apply concepts to specific problems), 172 logistics (pertain to class structure, not content), 18 factual (content related, with definitive answer), 213 not answerable (needs human intervention, eg., grading related), and 9 off-topic (unrelated to course content/structure). We focus on logistics questions and content-related factual questions since automated question answering methods are mainly developed for factual questions. In addition to the discussion forum data, there are 288 course material documents including the syllabus, assignments, notes, announcement emails from the instructor, exams, and sections of the electronic course textbook.

3.2 Question Answering

Experimental Setup. To more easily judge for correctness, we use the 172 logistical questions from the PhysicsForum dataset where students ask about assignment due dates, exam locations, grading policy, etc. Since our goal is to test document retrieval and question answering performance, we retrieve documents from the entire pool of course documents and discussion forum posts, regardless of the availability at the time a question was posted. We include several variants of our proposed question answering system in an ablation study. First, we compare retrieving documents via an inner product of tf-idf vectors (our system) versus cosine similarity [29], i.e., normalizing the inner product by dividing by document length (+N). Second, we compare splitting documents into paragraphs before document retrieval (our system) versus not splitting (−P).

Table 1. Document retrieval performance for all question answering systems.

System	Human	Watson	DrQA	Ours-P+N	Ours+N	Ours
Top-1 %	*80.2%*	38.4%	17.4%	51.2%	52.9%	**61.6%**
Top-3 %	*86.0%*	–	42.4%	79.7%	75.6%	**83.7%**
Top-5 %	*86.0%*	–	55.8%	89.0%	83.7%	**90.7%**

Table 2. Answer extraction performance for all question answering systems.

System	Human	Watson	Ours
Top-1 %	*80.2%*	38.4%	**41.3%**
Top-3 %	*86.0%*	–	**66.3%**
Top-5 %	*86.0%*	–	**77.3%**

Evaluation Metric and Baselines. We use human judgment to evaluate the performance of all systems[3]; for document retrieval, we label a document according to whether or not it contains information that could be used to answer the question, and for answer extraction, we label an answer span according to whether or not it is a satisfactory answer to the question. Sometimes, a satisfactory answer is not ranked first; this situation occurs when a question has some ambiguity, which makes the answer extraction model favor generic, indirect answers. Since our system produces a list of 5 answers with their rankings, we use Top-k accuracy to characterize the percentage of questions where at least one satisfactory answer is included in the top-k ranked answers, with $k \in \{1, 3, 5\}$.

We compare our question answering system against several baselines. The first baseline is the actual performance of course staff and other students (Human) in our dataset. We note that most questions elicit between 1–3 answers from students or course staff, causing Top-1 accuracy to differ slightly from Top-3 and Top-5 accuracies. The second baseline is the IBM Watson Assistant (Watson), which is given the 15 most commonly asked logistical questions in our dataset and the corresponding answers. We expect this baseline to be a near-optimal version of Jill Watson [9] because it knows a-priori the exact questions asked by the students. The Top-3 and Top-5 metrics do not apply to Watson because Watson does not provide a ranked list of answers; Instead, Watson randomly deploys one response from a pool of possible responses to a given question. The third baseline is the unaltered version of DrQA [19].

Results and Discussion. Table 1 shows the performance of document retrieval for all systems on the PhysicsForum dataset. Here, we only consider whether the retrieved paragraph contains information that could answer the question. For

[3] To gauge the subjectivity of our metric, we randomly sampled 50 question-answer pairs (across questions, answer ranks, and systems) and found moderate agreement between 2 independent labelers (80% agreement, Cohen's Kappa [37] = 0.554).

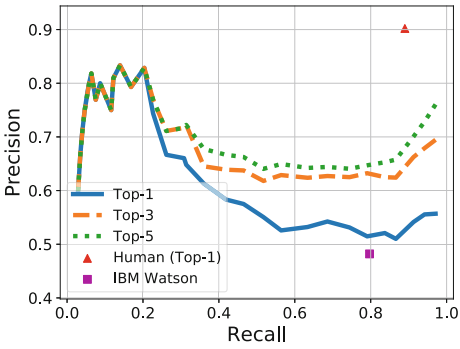

Fig. 1. Our system can use the answerability classifier to be selective in the questions it answers to improve precision at the expense of recall.

the Human and Watson baselines which do not answer every question, unanswered questions are treated as incorrect to ensure that recall is the same for all systems (since our system has 100% recall). We see that despite generally not reaching human-level performance, especially with the top-ranked answer, our system is the best-performing automated question answering method. Moreover, our system can automatically provide a satisfactory answer within its top-5 ranked answers for 90.7% of the questions, while the course staff answered only 86% of the questions. We also see that our system slightly outperforms its variants and significantly outperforms the original DrQA system that does not split documents into paragraphs in the document retrieval step. Even without this split, Ours-P+N still outperforms DrQA using the normalized cosine similarity values to rank documents instead of using inner products. These results suggest that splitting documents into paragraphs before the document retrieval process eliminates the need of normalization for the inner products between tf-idf vectors. The likely reason is that course content documents such as the syllabus and textbook sections are very long; splitting them into smaller paragraphs enables more past discussion forum posts to be selected. These threads can be highly relevant despite being relatively short.

Table 2 shows the answer extraction performance for all systems on the PhysicsForum dataset. We see that the performance on answer extraction is significantly worse than document retrieval, which means that the short span of text extracted from the document often fails to capture key information. This observation is not surprising since current answer extraction methods are designed for well-structured documents and questions with simple answers, while questions asked by real students often require complex answers using information from different pieces of the document. Therefore, if our system is to be deployed in real online course discussion forums, it is necessary to provide the entire retrieved document in addition to the short answer in case the answer itself is not satisfactory but its context still contains relevant information.

Question: How is our grade impacted if we forgot to do the catme team evaluation…
Top Answer: …There will be two team evaluations during this semester, one after the first three labs, and one after the last three. Each team evaluation will be worth 5%.
Source: Lab syllabus

Question: Does anyone know if we can still get credit for late Perusall comments?
Top Answer: …no extensions will be given." No credit for late homework.
Source: Forum post

Question: I did not receive full credit for the exam 1 corrections and there were no comments on moodle. How do I find out where I lost points?
Top Answer: …there will be a correction aspect to the exams…you will, after the exam be able to turn in some corrections to regain 15% of your lost points back…
Source: Course syllabus

Question: Where can I find the SI materials from the review session?
Top Answer: …There is a link on your moodle under "Supplemental Instruction" called "Location of SI materials"…Tuesdays and Sundays are used to go over class material…I upload worksheets 2 hours before every session…
Source: Forum post

Fig. 2. Examples of correct (left) and incorrect (right) answers provided by the system. The short answer chosen by the answer extraction model is underlined.

To improve the precision of our question answering system (at the expense of recall), we can use our answerability classifier's confidence value output to select questions to answer. Figure 1 shows the precision-recall curves with varying confidence threshold values for our system. The Human and Watson baselines are represented as points on the plot because neither supports multiple thresholding values. The precision shown for these baselines is higher than that in Table 1 since we are now using their actual recall values. We see that despite not reaching human-level performance, our system is capable of achieving 80% precision at 20% recall, which decreases to 50% at 80% recall. Moreover, our system slightly outperforms the IBM Watson-based system that was developed with knowledge of the questions (and corresponding answers) that were actually present in our dataset. This result means that our system can be readily used to enhance existing dialogue-based question answering systems that require non-trivial effort by instructors and TAs. Moreover, our system does not need to be warm-started using discussion forum data from previous offerings of the course; instead, all the instructor has to do is upload materials into the document pool.

Figure 2 shows several example questions and answers both when our system performed well and poorly. Figure 2(a) shows that our system can harness both course content documents and other forum posts to answer student questions. Figure 2(b) shows two cases where our system fails to extract a satisfactory answer. In one case, the text similarity-based document retriever finds a document about exam corrections, but the document does not answer the question of how to determine why the student did not receive full credit on their corrections. In the other case, the document contains relevant information about how students can access materials from a review session; this information is distributed throughout the document. However, existing answer extraction methods can only select a single short span in the text, causing the model to select a span that addresses the related (but different) question of when review sessions are held.

Question: How do you determine the direction of an electric field?
Top Answer: The electric field from a positive charge points away from the charge. The electric field from a negative charge points towards the charge.
Source: Forum post

Fig. 3. Beyond logistical questions, our system may also be used to answer content-related factual questions.

In addition to logistical questions, we also applied our system to the 18 questions we labeled as factual [38] in the PhysicsForum dataset. On these questions, our document retrieval system had Top-1, Top-3, and Top-5 accuracies of 44.4%, 88.9%, and 88.9%, respectively. Figure 3 shows an example where the system answers correctly. This example and the system's performance suggest that our system may be applicable beyond the limited domain of logistical questions, which we will explore in future work.

3.3 Post Timing Prediction

Experimental Setup. For the next post timing prediction task, we consider all threads in the PhysicsForum dataset. Using RNNs as the base model, we compare several variants of our method, including varying the number of discrete time bins as $k \in \{4, 8\}$, using the two different loss functions, cross entropy (C) and ordinal hinge loss (O), and the addition of two input features (A). These two additional features are derived from the answerability classifier; one is the maximum predicted probability of a satisfactory answer to the main question across all available documents, and the other is the maximum predicted probability of a satisfactory answer across only previous answer posts in the thread. We select bin boundaries such that EOT posts make up their own bin while the remaining posts are evenly placed into the other $k - 1$ bins. We train the RNN on a subset of 522 randomly sampled threads of the PhysicsForum dataset and evaluate it on the remaining 141 held-out threads. We evaluate the model using the time bin prediction accuracy (ACC) metric, which is simply the portion of correct predictions, and accuracy within one bin (ACC1), which is the portion of predictions that differ from the the actual time bin by at most 1. We repeat our experiments over 10 randomly sampled training and test sets.

Results and Discussion. Table 3 shows the mean performance across all training and test sets for all variations of our model. For reference, a majority-class model (EOT posts) achieves an ACC of 33.1%. We see that the ordinal loss provides a slight advantage over the cross entropy loss on both metrics under almost all settings. This observation suggests that the ordinal loss function that considers bin ordering by penalizing predictions farther from their actual bin more heavily is more effective than the cross entropy loss that does not consider bin ordering. We also see that the use of the answerability classifier improves the timing prediction performance but only marginally. This observation suggests that knowing whether the question in a discussion forum thread is answerable or whether it has already been answered can benefit next post timing prediction. However, this benefit is limited by the performance of the answerability classifier, which is not as accurate on the PhysicsForum dataset as it is on the standard SQuAD and Natural Questions datasets since real student questions are often ill-posed or poorly written. Nevertheless, the next post timing predictor can help virtual TAs to predict whether a student question is likely going to be answered by other students soon and decide whether to answer it immediately.

Table 3. Comparison between variants of the RNN using the cross entropy loss (C), ordinal loss (O), and the answerability classifier features (A) for predicting the discrete time bin of the time until next post.

System	$k = 4$				$k = 8$			
	RNN-C	RNN-O	RNN-CA	RNN-OA	RNN-C	RNN-O	RNN-CA	RNN-OA
ACC	75.2%	76.6%	75.4%	74.7%	69.8%	72.0%	70.0%	72.3%
ACC1	89.3%	91.9%	89.5%	92.3%	78.1%	79.5%	77.8%	80.6%

4 Conclusions and Future Work

In this paper, we have developed an automated system for logistical question answering in online course discussion forums and discussed how it can help the development of virtual teaching assistants. In addition to analyzing students' interactions with our system in a live course setting, avenues of future work include i) exploring what type of course content-based questions can be answered, ii) improving our system by fine-tuning neural language models on course content to adapt to student-generated text, and iii) developing methods that can automatically identify misconceptions in student posts.

References

1. Graesser, A.C., Person, N.K., Magliano, J.P.: Collaborative dialogue patterns in naturalistic one-to-one tutoring. Appl. Cogn. Psychol. **9**, 495–522 (1995)
2. Heffernan, N.T., Koedinger, K.R., Razzaq, L.: Expanding the model-tracing architecture: a 3rd generation intelligent tutor for algebra symbolization. Int. J. Artif. Intell. Educ. **18**, 153–178 (2008)
3. Lehman, B., Matthews, M., D'Mello, S., Person, N.: What are you feeling? Investigating student affective states during expert human tutoring sessions. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 50–59. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69132-7_10
4. Gaebel, M.: MOOCs: Massive Open Online Courses. EUA, Geneva (2014)
5. Leelawong, K., Biswas, G.: Designing learning by teaching agents: the Betty's brain system. Int. J. Artif. Intell. Educ. **18**, 181–208 (2008)
6. Karumbaiah, S., Lizarralde, R., Allessio, D., Woolf, B., Arroyo, I., Wixon, N.: Addressing student behavior and affect with empathy and growth mindset. In: Proceeding of the International Conference on Educational Data Mining, pp. 96–103, June 2017
7. Benedetto, L., Cremonesi, P., Parenti, M.: A virtual teaching assistant for personalized learning. arXiv preprint arXiv:1902.09289 (2019)
8. Adamson, D., Rosé, C.P.: Coordinating multi-dimensional support in collaborative conversational agents. In: Cerri, S.A., Clancey, W.J., Papadourakis, G., Panourgia, K. (eds.) ITS 2012. LNCS, vol. 7315, pp. 346–351. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30950-2_45
9. Goel, A., Polepeddi, L.: Jill Watson: a virtual teaching assistant for online education. Georgia Institute of Technology. Technical report (2016)

10. Bilgrien, N. et al.: PARQR: augmenting the piazza online forum to better support degree seeking online masters students. In: Proceedings of the Sixth (2019) ACM Conference on Learning at Scale, pp. 1–4 (2019)
11. Feng, D., Shaw, E., Kim, J., Hovy, E.: An intelligent discussion-bot for answering student queries in threaded discussions. In: Proceedings of the 11th International Conference on Intelligent User Interfaces, pp. 171–177 (2006)
12. Wang, C.-C., Hung, J.C., Yang, C.-Y., Shih, T.K.: An application of question answering system for collaborative learning. In: 26th IEEE International Conference on Distributed Computing Systems Workshops. (ICDCSW 2006), p. 49. IEEE (2006)
13. Ramesh, A., Goldwasser, D., Huang, B., Daumé III, H., Getoor, L.: Understanding MOOC discussion forums using seeded LDA. In: Proceeding of the Workshop on Innovative use of NLP for Building Educational Applications, pp. 28–33, June 2014
14. Chiu, T.K., Hew, T.K.: Factors influencing peer learning and performance in MOOC asynchronous online discussion forum. Australas. J. Educ. Technol. **34**(4), 16–28 (2018)
15. Lan, A.S., Spencer, J.C., Chen, Z., Brinton, C.G., Chiang, M.: Personalized thread recommendation for MOOC discussion forums. In: Berlingerio, M., Bonchi, F., Gärtner, T., Hurley, N., Ifrim, G. (eds.) ECML PKDD 2018. LNCS (LNAI), vol. 11052, pp. 725–740. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-10928-8_43
16. Benedetto, L., Cremonesi, P.: *Rexy*, a configurable application for building virtual teaching assistants. In: Lamas, D., Loizides, F., Nacke, L., Petrie, H., Winckler, M., Zaphiris, P. (eds.) INTERACT 2019. LNCS, vol. 11747, pp. 233–241. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29384-0_15
17. Reyes, R., Garza, D., Garrido, L., De la Cueva, V., Ramirez, J.: Methodology for the implementation of virtual assistants for education using Google dialogflow. In: Martínez-Villaseñor, L., Batyrshin, I., Marín-Hernández, A. (eds.) MICAI 2019. LNCS (LNAI), vol. 11835, pp. 440–451. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33749-0_35
18. Niranjan, M., Saipreethy, M., Kumar, T.G.: An intelligent question answering conversational agent using Naïve Bayesian classifier. In: 2012 IEEE International Conference on Technology Enhanced Education (ICTEE), pp. 1–5. IEEE (2012)
19. Chen, D., Fisch, A., Weston, J., Bordes, A.: Reading Wikipedia to answer open-domain questions. In: ACL 2017 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers), vol. 1, pp. 1870–1879. Association for Computational Linguistics (ACL) (2017)
20. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuad: 100,000+ questions for machine comprehension of text. In: EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings, pp. 2383–2392. Association for Computational Linguistics (ACL) (2016)
21. Raamadhurai, S., Baker, R., Poduval, V.: Curio SmartChat: a system for natural language question answering for self-paced K-12 learning. In: Proceeding of the Workshop on Innovative Use of NLP for Building Educational Applications, pp. 336–342, July 2019
22. Rajaraman, A., Ullman, J.D.: Mining of Massive Datasets. Cambridge University Press, Cambridge (2011)
23. Wang, S. et al.: R 3: reinforced ranker-reader for open-domain question answering. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)

24. Lin, Y., Ji, H., Liu, Z., Sun, M.: Denoising distantly supervised open-domain question answering. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1736–1745 (2018)
25. Das, R., Dhuliawala, S., Zaheer, M., McCallum, A.: Multi-step retriever-reader interaction for scalable open-domain question answering. arXiv preprint arXiv:1905.05733 (2019)
26. Halder, K., Kan, M.-Y., Sugiyama, K.: Predicting helpful posts in open-ended discussion forums: a neural architecture. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1 (Long and Short Papers), pp. 3148–3157 (2019)
27. Jenders, M., Krestel, R., Naumann, F.: Which answer is best? Predicting accepted answers in MOOC forums. In: Proceedings of the 25th International Conference Companion on World Wide Web, pp. 679–684 (2016)
28. Hansen, P., et al.: Predicting the timing and quality of responses in online discussion forums. In: Proceeding IEEE International Conference on Distributed Computing Systems, pp. 1931–1940, May 2019
29. Singhal, A., et al.: Modern information retrieval: a brief overview. IEEE Data Eng. Bull. **24**(4), 35–43 (2001)
30. Yin, J., Jiang, X., Lu, Z., Shang, L., Li, H., Li, X.: Neural generative question answering. arXiv preprint arXiv:1512.01337 (2015)
31. He, S., Liu, C., Liu, K., Zhao, J.: Generating natural answers by incorporating copying and retrieving mechanisms in sequence-to-sequence learning. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 199–208 (2017)
32. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
33. Houlsby, N. et al.: Parameter-efficient transfer learning for NLP. arXiv preprint arXiv:1902.00751 (2019)
34. Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: unanswerable questions for SQuAD. In: ACL 2018 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers), vol. 2, pp. 784–789. Association for Computational Linguistics (ACL) (2018)
35. Kwiatkowski, T.: Natural questions: a benchmark for question answering research. Trans. Assoc. Comput. Linguist. **7**, 453–466 (2019)
36. Rennie, J.D., Srebro, N.: Loss functions for preference levels: regression with discrete ordered labels. In: Proceedings of the IJCAI Multidisciplinary Workshop on Advances in Preference Handling, vol. 1. Kluwer Norwell, Norwell (2005)
37. Cohen, J.: A coefficient of agreement for nominal scales. Educ. Psychol. Measur. **20**(1), 37–46 (1960)
38. Wang, Z., Lan, A.S., Nie, W., Waters, A.E., Grimaldi, P.J., Baraniuk, R.G.: QG-net: a data-driven question generation model for educational content. In: Proceedings of the ACM Conference on Learning at Scale, pp. 1–10, June 2018