Detecting Off-Task Behavior from Student Dialogue in Game-Based Collaborative Learning

Dan Carpenter[1], Andrew Emerson[1], Bradford W. Mott[1], Asmalina Saleh[2], Krista D. Glazewski[2], Cindy E. Hmelo-Silver[2], and James C. Lester[1]

[1] North Carolina State University, Raleigh, NC 27695, USA
{dcarpen2, ajemerso, bwmott, lester}@ncsu.edu

[2] Indiana University, Bloomington, IN 47405, USA
{asmsaleh, glaze, chmelosi}@indiana.edu

Abstract. Collaborative game-based learning environments integrate game-based learning and collaborative learning. These environments present students with a shared objective and provide them with a means to communicate, which allows them to share information, ask questions, construct explanations, and work together toward their shared goal. A key challenge in collaborative learning is that students may engage in unproductive discourse, which may affect learning activities and outcomes. Collaborative game-based learning environments that can detect this off-task behavior in real-time have the potential to enhance collaboration between students by redirecting the conversation back to more productive topics. This paper investigates the use of dialogue analysis to classify student conversational utterances as either off-task or on-task. Using classroom data collected from 13 groups of four students, we trained off-task dialogue models for text messages from a group chat feature integrated into CRYSTAL ISLAND: ECOJOURNEYS, a collaborative game-based learning environment for middle school ecosystem science. We evaluate the effectiveness of the off-task dialogue models, which use different word embeddings (i.e., word2vec, ELMo, and BERT), as well as predictive off-task dialogue models that capture varying amounts of contextual information from the chat log. Results indicate that predictive off-task dialogue models that incorporate a window of recent context and represent the sequential nature of the chat messages achieve higher predictive performance compared to models that do not leverage this information. These findings suggest that off-task dialogue models for collaborative game-based learning environments can reliably recognize and predict students' off-task behavior, which introduces the opportunity to adaptively scaffold collaborative dialogue.

Keywords: Off-Task Behavior, Computer-Supported Collaborative Learning, Collaborative Game-Based Learning, Game-Based Learning Environments, Dialogue Analysis.

1 Introduction

Computer-supported collaborative learning can create highly effective learning experiences [1, 2]. It has been found that students benefit from learning in groups when given automated support [3], with conversation between students acting as a stimulus for learning [4]. In digital learning environments, collaboration can be achieved by allowing students to contribute to a group chat conversation [5, 6]. However, students can engage in off-task behavior [7], which can manifest as off-task chat messaging.

Off-task behavior has been identified as a significant challenge [8-10]. Because off-task behavior may be linked to boredom, which has been shown to negatively impact learning outcomes [11], it is important to enable learning environments to respond when students go off task. Although it has been found that off-task behavior can sometimes be beneficial for learning, as students may use off-task time to regulate negative affective states such as frustration [12], it is nonetheless important to identify student behaviors as off-task as such behaviors can be frequently associated with ineffective learning.

Determining when a behavior is off-task is challenging because whether a given behavior is on-task or off-task is highly dependent on the context in which the behavior occurs. To be able to provide adaptive scaffolding that responds to off-task behaviors, learning environments must be able to automatically detect off-task behavior in real-time. While there has been progress on characterizing types of off-task behavior [9, 13] and understanding their impacts on learning [12, 14], limited work has investigated automatically identifying off-task behavior. A particularly intriguing area of unexplored work is on identifying off-task behavior during collaborative learning. In this paper, we investigate off-task dialogue models to classify chat messages from interactions in collaborative game-based learning as off-task or on-task to inform the design of conversational agents that can guide groups that have gone off-task toward more productive dialogue.

Using chat log data collected from middle school students' interactions in CRYSTAL ISLAND: ECOJOURNEYS, a collaborative game-based learning environment for ecosystem science, we investigate off-task dialogue models for classifying students' conversational utterances as off-task or on-task during collaborative game-based learning. We investigate the effects of contextual information by comparing predictive models that only incorporate features derived from the current chat message to models that also include features derived from a context window of previous messages within the chat log. These include both static and sequential modeling techniques that utilize varying amounts of context. Additionally, we compare the use of several word embedding techniques for deriving features. First, we use pre-trained word2vec embeddings [15], which were trained on very large corpora to capture semantic and syntactic features of individual words. Second, we derive embeddings from the ELMo [16] and BERT [17] models, which use sequence-based neural networks to represent lexical semantics. These embeddings also leverage large corpora and augment each word embedding with additional information based on how the word is being used in specific contexts . Results demonstrate that sequential models that incorporate contextual information using both a window of previous dialogue and contextualized word embeddings yield substantial predictive accuracy and precision for detecting off-task student dialogue.

2 Related Work

Computer-supported collaborative learning (CSCL) has been shown to positively impact learning outcomes in a variety of contexts [1, 2]. However, providing students with a means to communicate during learning can potentially lead to off-task conversations. In a study examining discovery learning in a collaborative environment [7], dyads of high school students worked on separate screens in a shared environment and communicated via an integrated chat system. Researchers found that 15.7% of the chat messages were considered to be off-task, which by their definition meant that the messages had nothing to do with the task [7]. And while collaborative game-based learning environments offer the potential to create learning experiences that are engaging on many levels, the combination of collaboration and "seductive details" of game-based learning [8] can potentially exacerbate this issue, leading to off-task behavior.

The majority of previous work investigating off-task behavior in digital learning environments does not seek to automatically detect off-task behaviors. Rather, researchers commonly try to classify the type of off-task behavior and analyze the effects it has on learning [8, 10]. Some work has explored automatically detecting off-task behavior in digital learning environments. Baker [13] sought to detect off-task behavior in an intelligent tutoring system for math education, where off-task behavior was defined as behavior that did not involve the system or the learning task. Field observations of students' behaviors were used as ground truth labels for the machine learning algorithms used by Baker [13] and corresponded to the four categories set forth in Baker et al. [9]. As a baseline, Baker [13] set a threshold for time spent inactive, considering anything above that threshold to be an instance of off-task behavior. Our work extends this line of investigation and focuses on students' textual communication while engaging in collaborative learning.

Little work has analyzed natural language to detect off-task behavior. However, this approach is similar in vein to detecting the topic of students' writing [18-20] and analyzing student dialogue during collaboration [21, 22]. Louis & Higgins [18], Persing & Ng [19] and Rei [20] all used natural language processing methods to determine whether a student's essay is related to a given text prompt. Rei [20] made use of word embeddings for determining if an essay is related to a prompt. Similarly, we use word embeddings to determine if students' dialogue is related to either relevant curricular content or the collaboration process. Focusing more on collaborative learning, Adamson et al. [21] presented a framework for dynamically scaffolding online collaborative learning discussions using conversational agents that analyze students' conversations and respond to certain linguistic triggers. The work by Rodriguez et al. [22] demonstrated that specific characteristics of quality collaboration can be found by examining the contribution of multiple students, which we capture in off-task dialogue models that consider previous messages in the chat log.

3 Off-Task Dialogue Modeling

This work used data collected from CRYSTAL ISLAND: ECOJOURNEYS, a collaborative game-based learning environment on ecosystem science (Figure 1). Students work

together in the game to identify the causes underlying a sudden sickness affecting a fish species on a remote island. Students work at their own computers and share a virtual game environment with the other students in their group. Within each group of students, individual members take on unique roles in the storyline, gathering information that can help them solve the problem along the way. At various points during the story, students gather at an in-game virtual whiteboard to share what they have learned and work together to narrow down the causes of the fishes' sickness. Communication between students is achieved through an in-game chat system (Figure 1), where they can discuss what they have learned, ask their peers for help, or work together to construct explanations.

In this work, we utilized 4,074 chat messages collected from 13 groups of students. On average, each group sent 313.4 chat messages (min = 118, max = 617, *SD* = 155.6). Groups consist of four students and a facilitator, who observes students' problem solving and dialogue and guides their discussions. The researcher's role is to keep students on track and to occasionally ask leading questions to nudge them in the right direction. Within each group, students sent an average of 242.3 messages (min = 83, max = 553, SD = 141.9) and the researcher sent an average of 70.1 messages (min = 30, max = 125, SD = 30.1). Individually, students sent an average of 61.8 messages over the course of the study (min = 10, max = 203, SD = 47.7). Messages sent by the researcher were used as context for student messages but were not used as training or testing samples. As a result, the total number of messages available for training and testing was 3,150.

Fig. 1. (Left) CRYSTAL ISLAND: ECOJOURNEYS' gameplay.
(Right) CRYSTAL ISLAND: ECOJOURNEYS' in-game chat system.

3.1 Off-Task Message Annotation

We formulate off-task dialogue modeling as a supervised binary classification task. Thus, each message in the chat data is annotated as off-task or on-task. The annotation scheme builds on a classic dialogue act modeling framework [23] as well as dialogue act frameworks related to collaborative learning [22]. Like previous work [24], we label messages as on-task if they address relevant curricular content, foster collaboration, address affective states, or pose relevant questions. These messages are either related to the game's learning goals, self-regulation, or collaborative processes, so we consider them to be on-task. Some examples of chat messages and the labels assigned to them can be seen in Table 1.

Table 1. On-task and off-task chat messages.

	Definition	**Examples**
On-Task (0)	Productive text: any message that deals with the game's scientific content, fosters collaboration, addresses relevant affective states, or poses a relevant question.	"Water temp is warm needs to go in the water cold column" "What do I do I am at the house and have a map"; "Hi" (if the students are introducing themselves)
Off-Task (1)	Text that is not productive.	"I notice it seems I am the only one using capital letters around here"; "Nancy and I switched mice and switched back"

To label the chat messages, we first organized the messages by gameplay sessions, which were determined by the day that the students played CRYSTAL ISLAND: ECOJOURNEYS and the group to which they were assigned. This was done so that the sequences of chat messages used to create contextual features were all from the same group and occurred on the same day. The dataset contains 4,074 messages from 13 groups of students, which are split into 69 gameplay sessions. On average, each session includes approximately 59 messages (min = 1, max = 280, SD = 55.8). Each session, students sent approximately 45.7 messages on average (min = 1, max = 214, SD = 44.9) and the researcher sent approximately 17.1 messages (min = 0, max = 66, SD = 14.4). The data was labeled by two researchers using a rubric that was developed for this task (Table 1). Both researchers labeled 60% of the data, with an overlapping 20% to allow for calculation of inter-rater reliability. The raters achieved a Cohen's kappa of 0.751, indicating substantial agreement. For the messages that the raters did not agree on, labels were reconciled through discussion, and messages that appeared to contain both on-task and off-task dialogue were considered to be on-task. The final message labels contain 1,960 on-task (0) labels and 1,190 off-task labels (37.7% off-task), representing an imbalance. This is significantly higher than the rate of off-task conversation found in some other work [7], which may be because the learning environment combines collaboration and game-related elements.

3.2 Feature Extraction

To evaluate if the context in which a message occurs affects its classification as off-task or on-task, we generated context-based features as well as features that only used information from the current message. The message-specific features were the number of times the student had previously contributed to the group conversation, a score representing the polarity of the message's sentiment, the number of characters in the message, the Jaccard similarity of the message with the game's text content, and the average word embedding for the message [25].

Table 2. An example of 21 consecutive chat messages. A window containing a subset of the 20 preceding messages is used as context for predicting whether the last message is on- or off-task.

Number	Group Member	Message
1	Wizard (Facilitator)	How are you all doing? It would be great if you could go in and vote once you are done putting your evidence in.
2	Student A	We have voted
3	Student B	I am doing very well. I voted for every one and I am also ready for the next chapter. Game on!
4	Student C	And I believe we are done with entering our evidence
5	Wizard	I see that you are all very agreeable!
6	Student B	Great job!
7	Student C	:)
8	Wizard	But we also need to see if we can rule any of our hypotheses out to move on. Let's try to quickly see if we can go through the board. Scientists often have disagreements as they advance their ideas. They will look for evidence both for and against ideas. Let's start on the right with the unsorted ideas. Any suggestions where that might go?
9	Student B	Why thank you kind wizard :)
10	Student B	Ok
11	Student C	Not enough space
12	Student B	Not enough space
13	Wizard	And would that support or not support it? Let's talk about that.
14	Student A	If we put that in not enough space then it would kind of be going against it
15	Wizard	What do the rest of you think? How are we then on the 'not enough space' hypothesis?
16	Student B	Yes
17	Student C	Well I think that it should be even though it goes against it it still fits
18	Student A	It has no point in being there because it doesn't affect their health
19	Student A	For not enough space
20	Wizard	[Student A] and [Student B], what do you think? Why would we keep this hypothesis or remove it?
21	Student B	We should actually remove it. It doesn't fit in anything. I thought it over more.

Message sentiment was calculated using NLTK's [26] Vader sentiment analyzer. Because the game is dialogue-driven, information is presented through text-based conversations with in-game characters. We extracted this text from the game and removed stop words, as defined by NLTK's [26] list of English stop words. Then, the complete corpus of game text was compared against each message to calculate Jaccard similarity, which quantifies the similarity between the chat message and the game's text content [27]. If a message is very similar to the game's text content, then the student is likely talking about something that is relevant to the game and is therefore on-task. Jaccard similarity, which is the size of the intersection of two sets divided by the size of the union, was preferred over other text similarity metrics like the cosine similarity of tf-idf vectors, because Jaccard similarity only looks at the unique words that are common between two sources of text. This was preferable because

many words that are highly related to the game's educational content appear several times in the game's text, and tf-idf would discount these words because they are so common. For the message's average word embedding, we compared word2vec to ELMo and BERT embeddings to evaluate the effects of contextualized embeddings. We used word2vec embeddings with dimensionality 300, ELMo with dimensionality 256, and BERT with dimensionality 768. We used the ELMo embeddings generated from the second LSTM layer (i.e., layer 3 out of 3) to achieve the representation adding contextual information. For the BERT embeddings, we used the average of the token outputs across the 11th layer, which is the last hidden layer. Using these layers for both BERT and ELMo incorporates the richest representation produced by these embedding techniques, allowing for the most contextual information to be used.

For the context-based features, we defined a message's context as a sliding window containing the k previous messages in the chat log. Please see Table 2 for an example of chat dialogue. From these messages, we extracted the number of unique users who contributed to the conversation, the average length of messages in the context, the average time between messages, the number of times the learning facilitator sent a message, the cosine similarity between the current message's average word embedding and the word embedding of the most recent message from the researcher, the cosine similarity between the average word embedding of the current message and the average word embedding for all messages in the context, and the average Jaccard similarity between each previous message and the game's text content. During annotation, researchers noticed that off-task behavior often does not include every student in the team, so keeping track of the number of unique users during this chat window might be an indicator of off-task behavior. That is, if a small number of students are contributing heavily to the chat, it is likely that the messages they are sending are either consistently on-task or consistently off-task. Similarly, message length and time between messages could indicate off-task behavior, since short messages sent in rapid succession likely were not thoughtfully generated and could be off-task. Features related to the researcher's contributions to the chat could indicate off-task behavior, since more messages from the researcher could indicate that they needed to try harder to keep students on-task. Also, given that the facilitator's messages are examples of on-task dialogue, messages that were similar would likely be on-task. Since word embeddings allow words to be represented as real-valued vectors in a high-dimensional space, the cosine similarity between average word embeddings can be used to quantify the similarity of two messages.

3.3 Modeling

We first compared the performance of static models that incorporate contextual information to those that do not. The contextual models include features extracted from the previous 5, 10, 15 or 20 messages within the gameplay session. If there were fewer previous messages than the size of the window, we utilized the most messages available for calculating the features. Additionally, we evaluated the effects of different word embedding techniques (i.e., word2vec, ELMo, and BERT) on the performance of these models. We used logistic regression to perform this binary classifica-

tion. To ensure a fair feature set comparison, we performed principal component analysis (PCA) on the features for each representation to reduce the feature set to the first 50 principal components. We used standardization of the features before applying PCA, transforming both the training and testing data utilizing the training data's means and standard deviations.

We also investigated the performance of sequential models on this task. We built models that took in different window lengths (i.e., 5, 10, 15, 20) of previous messages, where each message was represented by the set of message-specific features described earlier. Sequences that were shorter than the length of the window were front-padded with zeros. Again, models were evaluated across each word embedding technique. For the sequential modeling task, we adopted LSTM-based sequential models with a single hidden layer. Hyperparameter tuning was performed across the number of nodes in the hidden layer (50, 100, 200, or 300), the activation function (sigmoid, hyperbolic tangent, or rectified linear unit), and the amount of dropout used (0.2, 0.3, 0.4, and 0.5). The optimal configuration was one hidden layer with 50 nodes, sigmoid activation function, and 30% dropout. These models were trained for up to 100 epochs, stopping early if validation loss did not decrease for 15 epochs. Models were trained using group-level 10-fold cross-validation.

4 Results

Results for the off-task prediction task can be found in Table 3. Among the static off-task dialogue models, we found that the most accurate feature configuration used the word2vec embeddings with a context window of size 5 (accuracy = 0.786). We also note that the majority class baseline accuracy for this data is 62.3%, which is the percentage of on-task messages. The improvement over the baseline indicates that the language-based representation of the chat messages does help with determining off-task labels. This same configuration also achieved the highest precision and F1 scores (precision = 0.710, F1 = 0.678). In general, we notice that all three scores tend to be highly related. We also note that, for all embeddings, a context window size of 5 performed the best for these models. Incorporating some amount of contextual information into the model improves performance over relying solely on features derived from the current message, confirming our hypothesis that context can help classify off-task behavior in collaborative game-based learning chat logs.

For the sequential models, the most accurate configuration was the BERT embedding with a window size of 20 (accuracy = 0.791). Both contextual embeddings (i.e., ELMo and BERT) outperformed word2vec across most window sizes. Moreover, these contextual embeddings benefit from longer window sizes, while word2vec still performed best with a window of size 5. While accuracy and F1 score were still correlated, accuracy and precision were less correlated than in the static models, with the most precise configuration being BERT with a window of size 5 (precision = 0.759).

Table 3. Results across embedding type, context window length, and model.

		Logistic Regression			*LSTM*		
EMBEDDING	**Context Length**	**Accuracy**	**Precision**	**F1**	**Accuracy**	**Precision**	**F1**
Word2vec	0	0.769	0.691	0.642	-	-	-
	5	**0.786**	**0.710**	**0.678**	0.774	0.710	0.636
	10	0.783	**0.710**	0.676	0.751	0.680	0.609
	15	0.781	0.707	0.670	0.744	0.659	0.604
	20	0.776	0.702	0.660	0.723	0.628	0.591
ELMo	0	0.754	0.662	0.615	-	-	-
	5	0.778	0.696	0.661	0.772	0.693	0.660
	10	0.775	0.701	0.654	0.781	0.707	0.667
	15	0.767	0.687	0.645	0.788	0.714	0.676
	20	0.766	0.681	0.643	0.789	0.720	0.678
BERT	0	0.745	0.664	0.635	-	-	-
	5	0.763	0.684	0.653	0.787	**0.759**	0.660
	10	0.768	0.696	0.659	0.787	0.731	0.674
	15	0.767	0.692	0.657	0.778	0.744	0.670
	20	0.763	0.687	0.651	**0.791**	0.714	**0.686**

Comparing static and sequential models, we find that the sequential models achieve the best overall performance, both in terms of accuracy and precision. This confirms our hypothesis that sequential techniques for modeling off-task behavior in student conversations outperform static techniques. While the static models performed best with short context windows, the sequential models make better use of longer context.

4.1 Discussion

For the static models, a short window of context yielded the best performance. A window of size 5 performed better than no context at all, and performance tended to decrease with longer windows. This may be because using too much context relies too heavily on information from the past, whereas information that is more recent can indicate components of the conversation's flow. Longer context windows likely include more information from irrelevant messages, and since the static models summarize previous chat messages by averaging features, relevant and irrelevant information are treated the same. However, the sequential models made better use of more context. The performance of the word2vec embeddings decreased as window size increased, but the contextual embeddings (i.e., ELMo and BERT) performed best with windows of size 20. We speculate that this may be due to the fact that ELMo and BERT create embeddings that, in addition to the syntactic and semantic information transferred from pre-training on large corpora, also encode some information that is related to the specific context in which words were used. Thus, while longer sequences accrue more noise from the solely pre-trained embeddings, the sequential models

may be able to focus on context-specific information captured by the contextualized embeddings.

We found that the simpler logistic regression models performed nearly as well as the LSTM models. While we might expect the gap between the static and sequential models to widen given more training data, since the LSTM may be able to pick up on more complex relationships than logistic regression, the static models performed well in this study. This may be due to the set of features that were used to represent the chat's context. In particular, we expect that the cosine similarity with the facilitator's most recent message and the average Jaccard similarity between each previous message and the game's text content could be very helpful in identifying messages as off-task. Since the facilitator's messages are examples of on-task dialogue, messages that are similar will likely be on-task as well. For instance, if a student is responding to the facilitator's question or talking about a similar topic, their messages would likely be similar. In much the same way, if the average Jaccard similarity between the messages in the context window and the game's text content is high, this is an indicator that students are likely talking about things that are related to the game and are thus on-task.

5 Conclusion and Future Work

Collaborative game-based learning environments create learning experiences that feature rich collaborative problem solving. However, students interacting with one another may at times engage in off-task behavior, which can manifest in off-task chat messages. If a collaborative game-based learning environment could utilize an off-task dialogue model to reliably recognize and even predict when students go off-task, it could facilitate more productive conversation. In this work, we have presented predictive off-task dialogue models that analyze students' chat conversations and detect off-task behavior. In particular, LSTM models that use contextualized BERT word embeddings achieve substantial accuracy for detecting off-task messages. These models perform best when provided with a context window of 20 previous messages, since they are able to effectively identify features of the previous messages that may be followed by instances of off-task behavior.

In future work, it will be instructive to investigate additional conversational modeling that considers participant role to determine the most relevant message to send to the students to get them back on task. Additionally, it may be possible to increase the predictive accuracy of models with word-by-word sequential modeling and sentence embedding. Together, these may significantly increase the ability of off-task dialogue models to recognize and predict off-task behavior, which opens the door to real-time adaptive facilitation that supports robust collaborative learning.

Acknowledgements. This research was supported by the National Science Foundation under Grants DRL-1561486, DRL-1561655, SES-1840120, and IIS-1839966. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. Chen, J., Wang, M., Kirschner, P. A., & Tsai, C. C. The role of collaboration, computer use, learning environments, and supporting strategies in CSCL: A meta-analysis. Review of Educational Research, 88(6), pp. 799-843 (2018).
2. Jeong, H., Hmelo-Silver, C. E., & Jo, K. Ten years of Computer-Supported Collaborative Learning: A meta-analysis of CSCL in STEM education during 2005–2014. Educational Research Review, 28, 100284 (2019).
3. Hmelo-Silver, C. E. Analyzing collaborative knowledge construction: Multiple methods for integrated understanding. Computers & Education, 41(4), pp. 397-420 (2003).
4. Rosé, C. P., & Ferschke, O. Technology support for discussion based learning: From computer supported collaborative learning to the future of massive open online courses. International Journal of Artificial Intelligence in Education, 26(2), pp. 660-678 (2016).
5. Jeong, H., & Hmelo-Silver, C. E. Technology supports in CSCL. In The Future of Learning: Proceedings of the 10th International Conference of the Learning Sciences (ICLS 2012), 1, pp. 339-346 (2012).
6. Jeong, H., & Hmelo-Silver, C. E. Seven affordances of computer-supported collaborative learning: How to support collaborative learning? How can technologies help?. Educational Psychologist, 51(2), pp. 247-265 (2016).
7. Saab, N., van Joolingen, W. R., & van Hout-Wolters, B. H. Communication in collaborative discovery learning. British Journal of Educational Psychology, 75(4), pp. 603-621 (2005).
8. Rowe, J. R., McQuiggan, S. W., Robison, J. L., & Lester, J. Off-Task Behavior in Narrative-Centered Learning Environments. In: Proceedings of the International Conference on Artificial Intelligence in Education, pp. 99-106 (2009).
9. Baker, R. S., Corbett, A. T., Koedinger, K. R., & Wagner, A. Z. Off-task behavior in the cognitive tutor classroom: when students "game the system". In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 383-390 (2004).
10. Beserra, V., Nussbaum, M., & Oteo, M. On-task and off-task behavior in the classroom: A study on mathematics learning with educational video games. Journal of Educational Computing Research, 56(8), pp. 1361-1383 (2019).
11. Baker, R. S., D'Mello, S. K., Rodrigo, M. M. T., & Graesser, A. C. Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive–affective states during interactions with three different computer-based learning environments. International Journal of Human-Computer Studies, 68(4), pp. 223-241 (2010).
12. Sabourin, J. L., Rowe, J. P., Mott, B. W., & Lester, J. C. Considering alternate futures to classify off-task behavior as emotion self-regulation: A supervised learning approach. Journal of Educational Data Mining, 5(1), pp. 9-38 (2013).
13. Baker, R. S. Modeling and understanding students' off-task behavior in intelligent tutoring systems. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1059-1068 (2007).
14. Cocea, M., Hershkovitz, A., & Baker, R. S. The impact of off-task and gaming behaviors on learning: immediate or aggregate?. In: Proceeding of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling, pp. 507-514. IOS Press (2009).
15. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111-3119 (2013).
16. Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018).

17. 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).
18. Louis, A., & Higgins, D. Off-topic essay detection using short prompt texts. In: Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics, pp. 92-95 (2010).
19. Persing, I., & Ng, V. Modeling prompt adherence in student essays. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1534-1543 (2014).
20. Rei, M. Detecting off-topic responses to visual prompts. In: Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, pp. 188-197 (2017).
21. Adamson, D., Dyke, G., Jang, H., & Rosé, C. P. Towards an agile approach to adapting dynamic collaboration support to student needs. International Journal of Artificial Intelligence in Education, 24(1), pp. 92-124 (2014).
22. Rodriguez, F. J., Price, K. M., Boyer, K. E. Exploring the pair programming process: Characteristics of effective collaboration. In: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education. ACM (2017).
23. Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., ... & Meteer, M. Dialogue act modeling for automatic tagging and recognition of conversational speech. Computational Linguistics, 26(3), pp. 339-373 (2000).
24. Mercier, E. M., Higgins, S. E., & Joyce-Gibbons, A. The effects of room design on computer-supported collaborative learning in a multi-touch classroom. Interactive Learning Environments, 24(3), pp. 504-522 (2016).
25. Sultan, M. A., Bethard, S., & Sumner, T. DLS@CU: Sentence similarity from word alignment and semantic vector composition. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 148-153 (2015).
26. Bird, Steven, Edward Loper and Ewan Klein, Natural Language Processing with Python. O'Reilly Media Inc (2009).
27. Niwattanakul, S., Singthongchai, J., Naenudorn, E., & Wanapu, S. Using of Jaccard coefficient for keywords similarity. In: Proceedings of the International Multiconference of Engineers and Computer Scientists, pp. 380-384 (2013).