User Modeling and User-Adapted Interaction (2023) 33:617–641
https://doi.org/10.1007/s11257-022-09340-z

Automatically detecting task-unrelated thoughts during conversations using keystroke analysis

Vishal Kuvar[1,3] · Nathaniel Blanchard[2] · Alexander Colby[3] · Laura Allen[1,3] · Caitlin Mills[1,3]

Received: 31 October 2021 / Accepted in revised form: 6 July 2022 / Published online: 19 August 2022

Abstract

Task-unrelated thought (TUT), commonly referred to as mind wandering, is a mental state where a person's attention moves away from the task-at-hand. This state is extremely common, yet not much is known about how to measure it, especially during dyadic interactions. We thus built a model to detect when a person experiences TUTs while talking to another person through a computer-mediated conversation, using their keystroke patterns. The best model was able to differentiate between task-unrelated thoughts and task-related thoughts with a kappa of 0.363, using features extracted from a 15 second window. We also present a feature analysis to provide additional insights into how various typing behaviors can be linked to our ongoing mental states.

Keywords Task-unrelated thought · Keystrokes · Machine learning · Affective computing · Mind wandering

✉ Vishal Kuvar
kuvar001@umn.edu

Nathaniel Blanchard
Nathaniel.Blanchard@colostate.edu

Alexander Colby
alexander.colby@unh.edu

Laura Allen
lallen@umn.edu

Caitlin Mills
cmills@umn.edu

1 University of Minnesota Twin Cities, Minneapolis, USA

2 Colorado State University, Fort Collins, USA

3 University of New Hampshire, Durham, USA

1 Introduction

Imagine a recent conversation with a friend. Despite your intentions to stay engaged with the conversation, your mind may have—and likely did—drift away from the conversation at times. You may have started thinking about what you needed to do later that day, or something they said may have triggered you to think about a memory from the past. These are both examples of what is known as task-unrelated thought (TUT), which is defined here as an internal mental state that is unrelated to one's current task (Smallwood & Schooler, 2015). TUT, often used synonymously with the term mind-wandering (Mills et al., 2018), occurs in virtually every scenario of our lives while we are awake (Killingsworth & Gilbert, 2010)—whether it be driving, reading or watching videos. TUT is not only remarkably frequent, it also has functional implications in our everyday lives, including affective (Killingsworth & Gilbert, 2010; Mills et al., 2021), educational (D'Mello & Mills, 2021; Smallwood et al., 2007), and clinical correlates (Arch et al., 2021), (Marchetti et al., 2016).

Whether positive or negative, the frequency and consequences of TUT highlight the need for reliable methods to detect its occurrence—especially using low cost, unobtrusive metrics in everyday life tasks. The current work takes a step in this direction by presenting the first real-time detector of TUT in the context of computer-mediated conversations using keystroke analyses. Cost-effective, stealth assessments of TUT in real-time offer novel ways to measure TUT without interruptions or task demands. TUT is currently almost exclusively measured via self-reports using online experience sampling or ecological momentary assessments, both of which necessarily interrupt ongoing tasks to obtain real-time assessments. Accurate real-time detection offers unique opportunities for interventions or personalized feedback (S. K. D'Mello et al., 2017; Hutt et al., 2021), (Mills et al., 2020). For example, knowing when and how often someone was off task in conversational contexts may eventually help facilitate more effective conversations, particularly in remote education and health contexts. Here, we take the first steps toward this goal, which is to identify effective low-cost methods for TUT detection in interactive contexts.

1.1 Related work

Below we review two relevant bodies of work. First, we describe previous work on the development of TUT detectors—all of which have existed outside of conversational contexts thus far. Second, we turn to a complementary body of work that has developed detectors of related cognitive-affective states in the context of language-relevant tasks (i.e., writing).

1.1.1 TUT

Detecting TUT in real-time is a growing area of interest, presumably for both measurement purposes and due to the link between TUT and task performance. The goal of any "good" TUT detector is to provide a real-time assessment about one's cognitive state (i.e., are they on vs. off task?) without needing on any prior information about

the person. That is, a "good" detector is both accurate, and it is generalizable to people it has never encountered before. Generalizability is commonly assessed using a cross-validation technique where the algorithm is trained on a set of people and then tested on a new person (or people) that it did not encounter in the testing phase. All the work we review below has used this cross-validation approach, where the training and testing sets have been kept explicitly separate. This is also the approach we use in the current work.

Reading tasks have been the most popular domain for real-time TUT detection to date, likely because of the association between TUTs and comprehension outcomes (Smallwood, 2011; Smallwood et al., 2007). Many researchers have utilized gaze behaviors for detecting TUT during reading, as eye movements provide a reliable window into cognitive processing (Rayner, 1998; Reichle et al., 1998). For instance, Faber et al. (Faber et al., 2018) used eye gaze to measure TUT during computerized reading and found that the proportional distributions of TUT predicted by the model and the self-reported TUT were similar and were significantly correlated, with a Pearson's r value of 0.40. Physiological features have also shown considerable promise as reliable detectors of TUT in reading tasks. Blanchard et al. (Blanchard et al., 2014) used skin temperature and skin conductance to detect TUT and found that these physiological measures could be used to detect TUT at 22% above chance. Bixler et al. (Bixler et al., 2015) improved on this performance by combining both gaze and physiological features to detect TUT during reading with 11% more accuracy than when only one of the two feature sets was used. Finally, other work has used features of the text itself to detect TUT during reading. For instance, Franklin et al. (Franklin et al., 2011) and Mills et al. (Mills & D'Mello, 2015) both used features of the reading itself (i.e., linguistic properties of the words) and participants' reading times to successfully classify TUT during reading.

Following the success of detection in reading tasks, researchers have also begun to examine the potential for developing models to detect TUT during other tasks, including ones that involve more dynamically changing external stimuli, such as video lectures (Hutt et al., 2021; Hutt, Mills, et al., 2016), narrative films (Mills et al., 2016; Stewart et al., 2016), and simulated driving (Baldwin et al., 2017). Hutt et al. (Hutt et al., 2019, 2017), for example, detected TUT in more dynamic learning scenarios using gaze as their primary modality. In one study (Hutt et al., 2017), they collected participants' gaze data while they watched recorded lectures, and were able to detect TUT with an F1 score of 0.47 (chance F1 = 0.30; where F1 represents the harmonic mean between precision and recall). Pham & Wang (Pham & Wang, 2015) also detected TUT during video lectures using heart rate with an accuracy of 0.712 and a kappa of 0.22 (where kappa represents percent above chance prediction levels). Finally, TUT detection is also possible in context like narrative film comprehension (i.e., while watching The Red Balloon movie) with either eye-gaze (Mills et al., 2016) or facial features and body movements (Stewart et al., 2016).

Taken together, these studies highlight the idea that behavior can be a reliable predictor of cognitive states. At the same time, there is a clear gap with respect to predicting TUT in interactive tasks that require coordination from multiple users at once (e.g., conversations, collaborative tasks). For this reason, we focus here on computer-mediated communication, which is increasingly common today, and even more so

with more education and workplaces shifting to dominantly remote, online interactions during the COVID-19 pandemic. Further, it is also important for TUT detectors to use unobtrusive and low-cost features for prediction. This is especially true for tasks that take place outside of the lab where eye-trackers and other physiological measures are not available (Baldwin et al., 2017). Although things like eye-trackers and heart rate monitors are increasingly becoming lower cost, they still present some issues compared to sensor-free features when considering the ability to apply solutions at scale.

1.1.2 Keystrokes as an indicator of cognitive processing

From the detectors mentioned above, it seems clear that TUT is reliably related to low-level physiological and behavioral patterns. At the same time, no work has linked TUT to language-production tasks such as conversation, where most robust signals of cognitive processing may come from the production patterns in the task itself. Here, we explore the idea that TUTs may be reliably detected from user's keystroke patterns while engaged in a computer-mediated conversation. This builds on a substantial body of work that has demonstrated a link between features of keystrokes and cognitive processes during text production tasks (Allen et al., 2016; Bixler & D'Mello, 2013). For example, findings from keystroke analyses suggest that text production (i.e., writing) often has bursts with more keyboard presses and pauses where nothing is written/pressed. The alternation between these two phases and the lengths of them can be informative of mental activity; some pauses may indicate thinking about what to say next, whereas others may signal evaluation or even disengagement (Bixler & D'Mello, 2013; Wengelin et al., 2009).

Previous work has attempted to capitalize on the link between keystrokes and cognition to predict cognitive-affective states, such as boredom and engagement, during writing tasks (Bixler & D'Mello, 2013), (Allen et al., 2016). In the first study, Bixler & D'Mello (Bixler & D'Mello, 2013) attempted to classify whether students were experiencing boredom, engagement, or a neutral affective during an essay writing task. They trained their model using keystroke analyses and trait-level variables to predict retrospective reports of affective states. Specifically, participants watched concurrent videos of their essay and their face and make periodic ratings about what affective state they were experiencing while writing (S. D'Mello & Mills, 2014). Their model could differentiate between boredom and high engagement with 87% accuracy and between boredom, neutral engagement, and high engagement with 56.3% accuracy. Similarly, Allen et al. (Allen et al., 2016) explored if overall levels engagement could be measured using keystroke analyses and features of the language itself during essay writing. Results demonstrated that a combination of keystrokes and text features could be used to detect levels of boredom during a writing task with 76.5% accuracy.

Beyond boredom and engagement, keystrokes have been used to predict other affective states as well. For example, Salmeron-Majadas et al. (Salmeron-Majadas et al., 2014) logged keystrokes from participants during a math task where they also answered intermittent probes questions about affective valence and arousal. Correlation analyses revealed that keystroke features like standard deviation of time between consecutive keystrokes, mean duration of diagraph were highly correlated with valence. Although

these features were also correlated with arousal, there were smaller effects than for valence. Another study by Epp et al. (Epp et al., 2011) collected timestamps for when the key was pressed and released while the participants performed free and fixed writing tasks. Participants answered probes from the emotional state questionnaire while performing both writing tasks. The researchers then built a model that was able to accurately classify between two levels (agree vs. disagree) for six out of the fifteen possible emotional states: confidence, hesitance, nervousness, relaxation, sadness, and tiredness. Further, in both aforementioned studies (Epp et al., 2011; Salmeron-Majadas et al., 2014), features were extracted from digraphs and trigraphs, making them reliant on content-based information.

Although these studies lend support to the idea that keystrokes may be a reliable indicator of TUT, it is important to point out some critical differences. First, both studies used features that extend beyond a content-independent keystroke approach. That is, they either took into consideration traits about the participants or what the participant actually wrote—both of which would limit the generalizability and scalability of a detector. Second, although boredom and engagement may be related to TUT, they are not the same construct. Whereas boredom is inherently defined by its affective components (its definition involves negative feelings; (Eastwood et al., 2012)), TUT does not share a one-to-one mapping with affect (Fox et al., 2018). Similarly, the relationship between boredom and TUT is unclear (Critcher & Gilovich, 2010; Eastwood et al., 2012; Raffaelli et al., 2018), but we do know that TUT can be a much more ephemeral state. Finally, it is important to point out that typing during an essay writing task is inherently different from typing in a conversation, where there may be other determinants of attention and engagement, such as the responsiveness of their chat partner or the topic of the conversation. In sum, there is ample theoretical and empirical support for our approach, but it is still an open question whether content-independent keystrokes will be a reliable detection method for TUT in a computer-mediated conversation.

1.2 Current study

We attempt to build a model that can detect TUT in a conversational setting. Participants chatted with one another anonymously through a chat application for ten minutes while their keystrokes were recorded. They self-reported TUT throughout the chat session. Models were trained on content-independent features of each participant's keystrokes using supervised learning algorithms and validated using a leave-one-participant-out method. We then test our model's generalizability using a second dataset where participants chatted with a chatbot instead of another human. To our knowledge, our study is the first to attempt TUT detection in a conversational setting as well as the first to detect TUT using keystrokes. The method proposed here could dramatically reduce the cost and intrusiveness for reliable TUT detection and measurement—requiring nothing more than a device that is capable of running a web browser and a keyboard to type with.

2 Methods

2.1 Participants and data collection

All methods and procedures were approved by the IRB at [removed for blind review] in and accordance with the Declaration of Helsinki. Data were collected from 126 undergraduate students at a public university. We did not obtain demographics data from 13 participants due to program errors but collected complete demographics information from the remaining 113 (75% female; Mage $= 19.179$, SDage $= 2.483$). Out of the 126 participants, chat files for 2 were lost and files for 4 participants were corrupted, leaving us with 120 participants' data.

Participants came to the laboratory to take part in a study involving a 10-min computer-mediated conversation with another participant, during which they self-reported instances of TUT. To disguise the purpose of the study and preserve participant anonymity, we recruited participants from two separate study listings on the university's research participation system. Each listing noted that the purpose of the study was to test a new computer program designed by the researchers. To minimize the risk of participants meeting prior to the session, each study listing was listed on a separate floor of the same building. All study sessions were completed on 13″ laptops.

2.1.1 Chat session

Conversations took place on the browser-based chat application ChatPlat. ChatPlat allows researchers to administer and customize synchronous text conversations between anonymous users and has been used and validated in previous studies (Huang et al., 2017; Wolf et al., 2016). Chat sessions were created with emoticons disabled and were designed to expire exactly ten minutes after the last member of a pair entered the session. Participants were informed that they could discuss any subject matter they preferred with their partner so long as they did not disclose identifiable information including their name, age, residence, or the university they attended. Keystrokes were collected using a Python script that began recording keystrokes as soon as participants entered the chat session.

2.1.2 Ground truth: TUT reports

We used a commonly employed and validated technique to collect TUT reports, referred to as a "self-caught" method (Varao-Sousa & Kingstone, 2019; Weinstein, 2018). This method asks participants to self-report instances of TUT as soon as they realize it occurs. We specifically asked participants to indicate TUT by pressing a designated button on the keyboard that was otherwise irrelevant to the chat session. Despite some limitations of this method, the self-caught method is quite frequently in TUT detection (Baldwin et al., 2017; Kopp et al., 2016; Stewart et al., 2016) and has the advantage of collecting TUT reports at any point in time rather than at specific moments (using probes) or retrospectively once the session ends. We thus consider self-caught TUT to be a suitable method for detecting TUT in the current study.

The TUT report button (left Control key) was covered with a sticker in order to avoid any confusion. This button made a small noise when pressed to notify the participant that their response had been registered. We instructed participants to press the button anytime they found themselves thinking about anything besides the conversation. We also provided participants with an example (thoughts drifting from the conversation to a test coming up in a class) to further ensure that participants were familiar enough with the phenomenon to be able to report it accurately.

2.2 Supervised machine learning

2.2.1 Data processing

Participants' keystrokes were recorded, along with timestamps, as they conversed. We removed all participants who did not register any TUT reports, as no TUT features could be extracted from such participants. Since Kappa was calculated per-participant, Kappa is difficult to quantify in the absence of reports since there is only one 'true' class (i.e., TUT). Further, the absence of TUT may also indicate that participants misunderstood or forgot the instructions—i.e., we assume that although it is possible for someone to not experience TUT in the span of ten minutes, it is highly unlikely for that to happen in this study given that computer-mediated conversations involve a significant period of waiting for the partner to respond. In total, 11 participants out of the 120 did not report any TUT and were dropped, making our final count for model building 109 participants.

The keylogger that was used to capture the keystrokes of the participants did not register combination keys. One way to type the capital letter 'I' would be pressing the shift key and the 'i' key simultaneously. The logger stored both of these keys separately. To fix this, the letter key pressed after the shift key was pressed was capitalized and shift key was removed from the keystrokes. The timestamp for the capitalized letter would be the timestamp of the latter of the two keystrokes. The same procedure was used to create question marks and exclamation marks.

Since this was a computer-mediated conversation, internet slang and punctuation were to be expected. Repeating punctuation marks for emphasis is a common practice seen in CMCs. Some common examples of these are '???', '!!!' and '…'. In such a case, the set of punctuations are replaced with the last punctuation in that set and its and timestamp.

2.2.2 Feature extraction

Creating windows for TUT vs. TRT We extracted features from 109 participants' data. In order to use a supervised machine learning, we first needed to create our labeled data; that is, instances of time that represent a TUT report vs. instance that represent task-RELATED thought (TRT). Given our self-caught method, we were able to use the TUT reports as our "labeled" TUT data and considered all other times to be TRT. A critical aspect of building a real-time detector is that the features (here, keystroke patterns) used by the algorithm to make predictions needs to occur before the report.

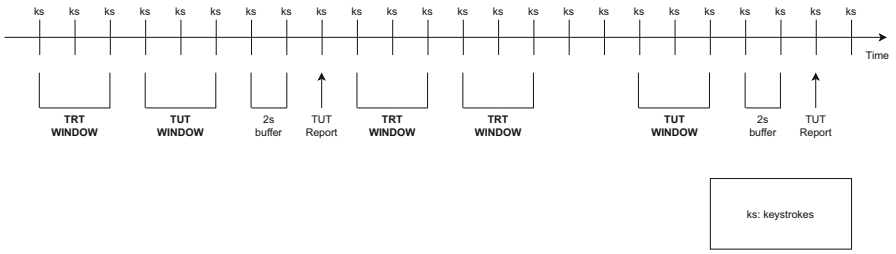

Fig. 1 Window creation for TUT and TRT

This way, the prediction is not based on the report itself, but rather the patterns of data leading up to the report. For this reason, we needed to create "windows" of data leading up to each TUT report as well as windows that represent TRT.

We took the following approach to create such windows. Our feature extraction strategy is graphically represented in Fig. 1 and is similar to approaches used in previous studies (Stewart et al., 2016). We wrote a program to traverse each CSV file until a TUT report was identified. We adopted a TUT report "buffer" approach, which is commonly used in other detection attempts (Faber et al., 2018; Stewart et al., 2016), which involves discarding at least two seconds of data before the TUT report. This "buffer" is intended to accommodate the gap in time when the participants recognized and subsequently reported their TUT state, and to ensure the algorithm does not simply learn this report behavior. Since keystroke timestamps occur at non-fixed intervals, the buffer that was inserted may have been longer than two seconds by nature.

Once a TUT report was identified, features were extracted from the keystrokes using two different a pre-determined window size (15 s, 30 s) preceding the buffer. The 2 window sizes were chosen to determine how much keystroke data are needed for TUT detection in conversational contexts. All the keystrokes that fell into a given window were used for feature extraction.

TRT windows were created under the assumption that participants were likely engaging in task-related thought (TRT) outside of the TUT windows (or else we would have expected a TUT report to occur). TRT windows were created before, in between, and after TUT windows. To create TRT windows in between two TUT windows, the time difference was measured between the end of the first TUT window and the start of the second TUT window. If the difference was greater than the window size (e.g., 15 s, 30 s), we created as many TRT windows as possible. As an example, if participants did not engage in TUT for over 2 min, we would be able to create a maximum of 8 TRT windows for the 15 s model (provided all the windows are exactly 15 s). In case of a single TUT section present in the chat we used the remaining time in the chat to create as many TRT windows as possible. The TRT windows did not require a similar buffer since there is no realization period for the participant as there was during the TUT report. The TRT windows were of the same size as the TUT ones. If no TRT windows were created for a participant, that participant would be dropped from our analyses; there was only 1 such case for the 15-s window and 2 for the 30-s window.

Table 1 Distribution of TUTs and TRTs for 15-s and 30-s windows

	TUT		TRT	
	Total	Mean (SD)	Total	Mean (SD)
15 s	374	4.021 (2.545)	1425	15.322 (7.284)
30 s	243	3.283 (1.675)	608	8.216 (4.171)

It is also important to point out that TRT will be much more common than TUT; this is true for almost all TUT detection papers to date. In real life, we are only off-task a subset of the time and our dataset should reflect this—producing an imbalanced dataset with more TRT windows that TUT windows. The distributions of ground truth TUT and TRT instances for both 15- and 30-s windows are shown in Table 1.

Extracting keystroke patterns After creating the windows for TUT and TRT instances, we extracted features from each of them. A total of 38 features were created, broadly spanning two categories: *non-message* and *message*. Features that required the recreation of the messages sent by the user in the window were categorized as *message features* and others were categorized as *non-message features*. Table 1 provides a summary and definition for all features extracted.

Our non-message features were derived based on the ones use by Bixler & D'Mello (Bixler & D'Mello, 2013). *Verbosity* measures the number of keystrokes that were pressed by the participant in that window, including the idle keystrokes. *Backspace frequency* measures the number of times the participant pressed the backspace key in the feature extraction window. *Latency* was defined as the amount of time between two consecutive keystrokes. We used the mean, median, maximum and the minimum values of all the latencies present in the window. Lastly, we defined the *Number of pauses* as the number of times the user paused in the window for a given period of time. We used five different intervals to calculate these pauses, each as a unique feature: 0.25–0.75 s; 0.75–1.25 s; 1.25–1.75 s; 1.75–2.25 s; 2.25–2.75 s.

For message features, we automatically recreated the actual message by mapping each keystroke in a sequential order. If the actual message within the window was blank, that window was skipped for message features. Such a case could arise if the sequence of keystrokes results in an overall blank message an example of which is- 'h', 'e', 'y', 'backspace', 'backspace', 'backspace'. Message recreated from this set of keystrokes will be blank. If the recreated message was not blank, we extracted the message features from the window. Note that these features were still content-independent, meaning we purposefully ignore the content of the message for generalizability and scalability purposes.

The first message feature was the *Length* of the actual message being sent. Additionally, we included the *Number of words*, *Number of sentences*, and *Number of messages* sent. Words were defined as groups of letters separated by spaces. Sentences were separated with periods, exclamation marks, and question marks, and messages were separated by the press of the enter key. *Inter-word time*, *Inter-sentence time* and

Inter-message time were defined as the average time between consecutive words, sentences, and messages, respectively. In case of a single word, sentence or message, the inter-word, inter-sentence, and inter-message time were set to zero, respectively. We also used *Word length* as a feature. There were two different ways to calculate word length: number of keys pressed, and time taken to type a word, which required us to determine where a word started and ended. We considered the word to start from where the last word ended. For example, if a participant were to hit the following keys in order: 'h', 'i', 'backspace', 'backspace', 'h', 'e', 'y', 'space', the message that would be sent is 'hey'. In this case, the word length will be seven even though the actual length of word sent is three. The indices of the start and end point of the word were stored, and the length of each word was calculated for the entire window. The minimum, maximum, mean and median values of all of the word lengths were included as features. Finally, we calculated the *Length of a sentence* in three ways: keys pressed, time taken to type each sentence and number of words in each sentence. The minimum, maximum, mean and median values of all of the sentence lengths were included as features. Table 2 summarizes the features that were extracted.

2.2.3 Validation and evaluation

Cross-validation We used the leave-one-participant-out validation method for this study. This means that all the participants except one (k-1) were used to train a model, and that model's performance was then measured by testing it on the one participant that had been held out. This validation method is performed k times, where k refers to the number of participants in the dataset—each time, a different participant is left out to test the model. Using the leave-one-out validation method ensures that the model is evaluated on new and unseen participants; this ensures the models will generalize to real-world conditions. Overall model performance is then assessed by taking the average performance across the k iterations. Here we provide average performance as well as histograms for how the models perform across all individuals.

Evaluation metrics Supervised classifiers can be evaluated multiple ways, with much debate about which metric is the gold standard. Even within the TUT detection literature, different papers report different metrics. For this reason, we report multiple performance metrics to provide easier facilitation with past and future work: accuracy, kappa, Matthew's correlation coefficient, ROC AUC and F1 score. We also report the confusion matrix for the models. Accuracy is defined as the ratio of correct predictions to the total number of predictions; however, we encourage readers to ignore accuracy given our highly imbalanced dataset (i.e., predicting the majority class would still lead to a very high accuracy). F1 score is the harmonic mean of precision and recall, where precision is the ratio of total positive to total predicted positive and recall is the ratio of total positive to total actual positive. Kappa (McHugh, 2012) measures the agreement of predicted values and the actual values while taking the chance agreement out of the picture. In other words, it tells us how much better our model is performing over the performance of a model that simply guesses at base rate based on the class distribution. Kappa values can range from -1, which indicates opposite agreement, to 1 which indicates complete agreement. A kappa value of 0 would indicate no agreement, chance

Table 2 Description of extracted features

Type	Feature name	Number of features	Feature description
Non-message features	Verbosity	1	The number of keystrokes in the window
	Backspace Frequency	1	The number of times the backspace key was hit in the window
	Latency	4	The difference between two successive keystrokes in the window (min, max, mean, median)
	Pause	5	Number of pauses for each interval (0.25 s—0.75 s, 0.75 s—1.25 s, 1.25 s—1.75 s, 1.75 s—2.25 s, 2.25 s—2.75)
Message features	Length of message	1	Length of the recreated message in the window
	Number of words	1	Number of words in the recreated message (separated by the space key)
	Number of sentences	1	Number of sentences in the recreated message (separated by period, exclamation mark, question mark)
	Number of messages	1	Number of messages sent in the window (separated by enter key)
	Inter-word time	1	Mean time between consecutive words in the recreated message
	Inter-sentence time	1	Mean time between consecutive sentences in the recreated message
	Inter-message time	1	Mean time between consecutive messages in the recreated message
	Word length (keystrokes)	4	Number of keystrokes logged to type a word (min, max, mean, median)
	Word length (time)	4	Time taken to type a word (min, max, mean, median)
	Sentence length (keystrokes)	4	Number of keystrokes logged to type a sentence (min, max, mean, median)
	Sentence length (timestamp)	4	Time taken to type a sentence (min, max, mean, median)
	Sentence length (words)	4	Number of words logged to type a sentence (min, max, mean, median)

performance. The final metric we include is the Matthew's correlation coefficient. MCC is reported as an alternative to the F1 score, which can be asymmetric. The F1 score ignores the true negative cell of a confusion matrix, which at times might be problematic if the dataset is imbalanced, like in our case. The metrics are defined as follows (Ferreira, 2018; McHugh, 2012):

TP, TN, FP, FN refer to the true positive, true negative, false positive, and false negative in a confusion matrix, respectively.

$$accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

$$precision = \frac{(TP)}{(TP + FP)}$$

$$recall = \frac{(TP)}{(TP + FN)}$$

$$f1 = 2 * \frac{precision * recall}{precision + recall}$$

$$randomAccuracy = \frac{(TN + FP) * (TN + FN) + (FN + TP) * (FP + TP)}{(TP + TN + FP + FN)^2}$$

$$kappa = \frac{accuracy - randomAccuracy}{1 - randomAccuracy}$$

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}}$$

2.2.4 Model building

As discussed in our Validation section, we used a leave-one-participant out validation method. We created training sets which included all the participants but one and that one left out participant was used as the test set. We also used a commonly applied technique called SMOTE (synthetic minority over-sampling technique) (Chawla et al., 2002) to deal with our highly imbalanced training set. SMOTE creates create synthetic data points in the training data, balancing TUT and TRT as the model "learns" the patterns; however, the test set is kept as-is. SMOTE is an increasingly common technique in supervised machine learning and is often used TUT detection as well (Faber et al., 2018).

We also test performance across two different window sizes: 15 s and 30 s, for the calculation of the keystroke features. This is another common approach in TUT detection and will be of particular interest here to know how much text data is required to make accurate predictions. For example, 15 s may perform the best because the data are more proximal to the TUT report; in contrast, 30-s windows may simply provide more data for the calculation of each feature. We are interested in assessing this trade-off in the computer-mediated conversation context.

This is the first work to attempt detection of TUT in the context of conversations, so it was unclear what classification algorithms and hyperparameters would perform best. We thus trained and evaluated a series of supervised classification algorithms and hyperparameters on the detection of TUTs. Specifically, we used seven commonly applied different classifiers and implemented them using the Scikit-learn library in Python (Pedregosa et al., 2011): decision tree, linear support vector machine, random forest, K-nearest neighbor, Gaussian process, Adaboost, and quadratic discriminant analysis. Each of these has optional hyperparameters, classifier-specific conditions or values that are set before training. Due to the high number of possible hyperparameters, and classifier combinations, we automated the training and evaluation of all of these

conditions using the Hyperopt package (Bergstra et al., 2015). Hyperopt implements Bayesian optimization (Snoek et al., 2012), using Python, to automatically find the best hyperparameters for detecting TUTs. Bayesian optimization evaluates the results from the previous models to decide which combinations to evaluate next. By default, the first 20 model conditions are random to initialize the Bayesian optimization. A model would be considered "better" than the best model if its Kappa value was higher. This process of creating new models and comparing its kappa to the best model was repeated an additional 130 times, creating 150 models in total, and the model with the highest kappa at the end of this process was selected.

We note that having such a high number of possible combinations of hyperparameters made the hyperparameter tuning incredibly complex. Compared to these combinations, the number of data points available was relatively small and thus introduces a possibility of overfitting in the search itself. Accounting for this, we test our final model on a completely separate dataset—something that it has never encountered before and using a different chatting context (chatbot vs. human). Using this additional validation method, we would expect to see a large drop off in performance if the model has learnt the initial dataset itself instead of learning from it.

3 Results

3.1 Best models

Table 3 shows the best kappa values and its standard deviation for both the windows with normal data as well as SMOTE data.

The 15-s window appears to have better performance overall compared to the 30-s window. Although SMOTE improves model performance in both windows, the 15-s window model is the best overall. The full performance metrics of the best models for 15 s and 30 s can be found in Table 4. Both the windows use the random forest classifier to train the best model.

The confusion matrices for the best models for each window are shown in Table 5. Even though the dataset is imbalanced, the models do not selectively predict one class over the other, which is ideal given that our main goal is to predict the minority class.

We also examined the frequency distribution of the kappa values in the leave-one-participant-out validation method. This was necessary because the model could have been predicting TUT with a high level of confidence for certain participants and with extremely low confidence for the rest, creating a reasonable "average" performance even though many people may be completely unpredictable (i.e., kappa close to zero). Such a model may have a decent kappa value overall but would not have generalized

Table 3 Kappa values for the best models for 15-s and 30-s windows

	No SMOTE	SMOTE
15 s	0.323 (0.326)	**0.343 (0.258)**
30 s	0.304 (0.304)	0.331 (0.296)

Table 4 Performance metrics for 15-s and 30-s windows trained with SMOTE

Window size	Classifier	Accuracy	ROC-AUC	F1 (TUT)	F1 (TUT) chance	F1 (TRT)	MCC
15 s	Random forest	0.775	0.692	0.507	0.333	0.862	0.372
30 s	Random forest	0.730	0.690	0.557	0.461	0.814	0.373

Table 5 Confusion matrices for each window

	Predicted values			
	Actual Values	TUT	TRT	Total
15 s	TUT (0.20)	199	175	374
	TRT (0.79)	210	1215	1425
	Total	409	1390	1799
30 s	TUT (0.30)	140	103	243
	TRT (0.69)	119	489	608
	Total	259	592	851

well to new data; ideally, the models' kappa values would have a normal distribution. Figure 2 shows the distribution of kappa values, which indeed looks relatively normal.

Lastly, we plotted the kappa values against the number of TUT reports (note one participant with 17 TUT reports and a kappa of 0.2 was removed from the figure for better visualization of the linear pattern). We inspected the possibility that kappa values were merely a function of the number of reports. Figure 3 indicates that the kappa values are relatively independent of the number of TUT reports; a Pearson's

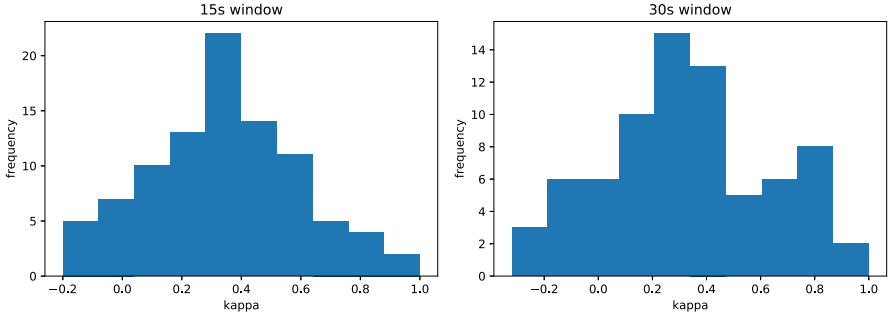

Fig. 2 Kappa distribution values of 15 s model (left), 30 s model (right)

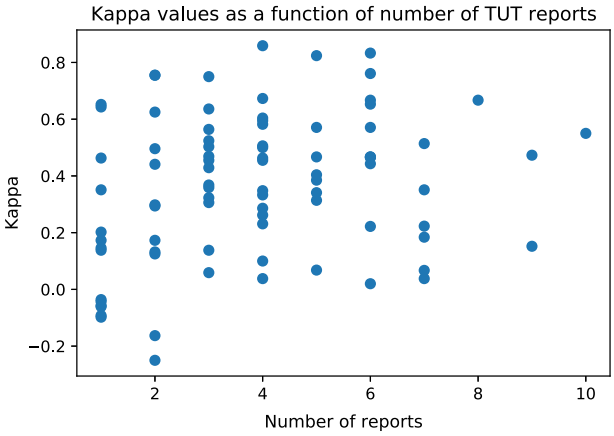

Fig. 3 Distribution of kappa values against number of TUT reports

correlation of 0.252 suggests that the number of reports only explained about 5% of the variance in the kappa value.

3.2 Feature analysis

We also examined how each feature group influenced model performance: message length, latency, pauses, count, inter-element, word length, sentence length. We thus created models that were: (1) trained on an individual feature group alone; and (2) trained by withholding an individual feature group. A description of each feature group can be seen in Table 6.

We trained each feature group using the same algorithm (random forest classifier) and hyperparameters that was used in our "best" model. The kappa values of these models can be seen in Table 7. The feature group of *Sentence length* appears to have the biggest influence; this feature group alone had the highest kappa value of 0.310,

Table 6 Feature groups

Feature group	Features included
Latency	All four of the latency features
Pauses	All five of the pauses' features
Count	Number of words, sentences, and messages in the reconstructed window
Inter-element	The average time between consecutive words, sentences, and messages in the reconstructed window
Message length	Verbosity, length of actual message
Word length	All eight features that were extracted for the word length
Sentence length	All twelve features that were extracted for the sentence length

Table 7 Results of models trained on one feature group and by removing one feature group

Feature groups	Train with feature group	Train with all except feature group
Sentence length	0.310	0.269
Word length	0.213	0.353
Inter-element	0.174	0.324
Count	0.148	0.336
Length	0.121	0.330
Latency	0.050	0.320
Pauses	0.018	0.338
All features	0.343	–

and largest drop in the kappa occurred when it was removed. In contrast, the feature group of *Pauses* had the smallest influence on the overall model. It is also interesting to observe that *Word length* had the second highest kappa of the models trained on individual feature groups, but the overall kappa of the model also improved when this feature is removed, suggesting its presence along with other features somehow confounds the model, but is strong enough itself to drive the model. This feature group's interaction with the rest of the features is something to be examined.

We then repeated this same procedure on individual features (rather than groups) by training on one feature individually and removing one feature at a time. The kappa values of the top 10 models are reported in Table 8. (Each feature can be interpreted as follows: *feature group, measured as, statistic.*)

Based on the results shown in Tables 7 and 8, we can infer that the model is likely reliant on feature groups rather than individual features. Models trained on individual features have a lower kappa, whereas models with one feature removed have a kappa value very close to the actual model.

3.3 Feature selection

A machine learning model sometimes performs better on a subset of features as opposed to all of them. Although we had no theoretical basis to remove features at the beginning, there are other issues to consider including selecting a more parsimonious model. We examined the number of features that would be ideal given our entire set of features by re-training models using a feature selection algorithm. The first model was trained on the best feature selected, the second one with two features. This continued till the final model was trained with all the features included, giving us a total of 38 models, each with one extra feature than the previously trained model. We used the select-k-best feature selection algorithm. The algorithm and hyperparameters were the same as the best model we trained before. Figure 4 shows how each of the selection performed. The best model had a kappa value of 0.362 with a standard deviation of 0.255 using 24 features, offering a slight improvement over the model with no feature selection (kappa = 0.343).

Table 8 Results of models trained on one feature and by removing one feature

Features	Trained on one feature by itself	Removed one feature from best model
Sentence length, num. words, minimum	0.192	0.344
Sentence length, keystrokes, minimum	0.190	0.328
Sentence length, timestamp, mean	0.179	0.304
Sentence length, timestamp, minimum	0.178	0.334
Word length, keystrokes, mean	0.161	0.320
Word length, keystrokes, minimum	0.160	0.355
Sentence length, timestamp, median	0.158	0.319
Sentence length, word, mean	0.153	0.347
Sentence length, word, median	0.151	0.343
Sentence length, timestamp, maximum	0.148	0.324
All features	0.343	–

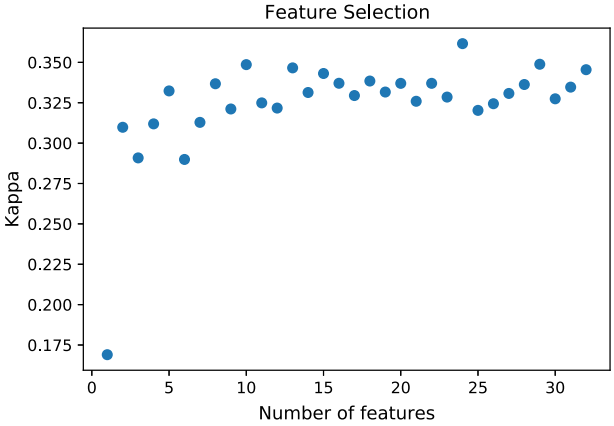

Fig. 4 Kappa values based on the number of features selected

3.4 Feature importance

Out of the 24 features selected in this model, we then examined how each feature contributed to the model. We calculated the feature importance of each feature in the model, with the top 10 features including: maximum word length (keystrokes), maximum sentence length (keystrokes), maximum sentence length (timestamp), length of message, maximum sentence length (word), mean word length (timestamp), minimum latency, inter-word time, 0.75–1.25 s pause, maximum word length (keystroke). Feature importance was determined by the Gini impurity (i.e., the probability of misclassifying the observation). Specifically, feature order was determined by identifying which feature minimizes the measure.

We then sought compared the top 10 features across the TUT and TRT in order to glean some insight into the actual keystroke behaviors associated with each class. We performed paired samples t-tests and calculated Cohen's d values to compare each feature (see Table 9). Five features were significantly different across the two classes: Maximum sentence length number of keystrokes; Maximum sentence length based on time, Maximum sentence length based on number of words, Length of message, and Minimum latency. The sentence length features all suggest that sentence lengths are longer during TUT compared to TRT. There also appears to be shorter pauses in TUT as well.

At first glance, these patterns may seem counterintuitive. One may expect that TUT would have shorter sentences and longer pauses. However, recall that we are making

Table 9 Feature descriptive table for TUT and TRT for the 15-s window

Features	Task-related thought (N = 1425)		Task-unrelated thought (N = 374)		*p*	*d*
	Mean	SD	Mean	SD		
Maximum word length (timestamp)	8.67	6.29	10.6	11.2	0.135	0.144
Maximum sentence length (keystroke)	33.1	8.03	37.2	13.2	**0.002**	0.302
Maximum sentence length (timestamp)	13.4	6.25	16.6	10.9	**0.012**	0.246
Length of message	27.0	7.53	34.2	14.0	**< 0.001**	0.541
Maximum sentence length (word)	5.01	1.34	5.83	2.52	**< 0.001**	0.339
Mean word length (timestamp)	5.06	5.95	4.08	8.74	0.340	0.092
Minimum latency	2.81	6.01	0.577	2.42	**< 0.001**	0.338
Inter-word time	0.516	0.248	0.524	0.309	0.806	0.023
0.75 s – 1.25 s pause	1.19	0.488	1.82	3.42	0.341	0.092
Maximum word length (keystroke)	13.8	4.95	12.7	4.84	0.078	0.171

predictions across the entire chat session, including both generating and receiving chats. TRT windows (the much more common class label) are often created between a user's conversational turns—for example, when a participant is waiting for their partners' reply. Thus, even though the average number of messages is greater for TRT compared to TUT, there is more idle time during the TRT windows, possibly driving the overall sentence features down. This is further evidenced by that fact that the majority of the TUT reports came during the user's own conversational turns (74.12%) compared to when they were awaiting a reply. It is also important to note that none of these features performed well individually—rather multiple features were needed for accurate classifications.

3.5 Generalizing to new data: chatbot conversations

We next tested the generalizability of our model in a different context; namely when users were chatting with a chatbot instead of a human. The human dataset used above was the training dataset, and a chatbot dataset was the test set. We used the best classifier and hyperparameters from the two-person conversation model to train these data. The 15-s window was used since it was the one that gave the best results.

Data that were collected from the same set-up with the only exception being that participants spoke with a chatbot instead of another human. Participants conversed with a free online chatbot but were not explicitly informed that the other agent was not a human. To increase the ecological validity of chatting with a chatbot, a human transcribed the chatbot's responses in real time so the message cadence felt more realistic In total, we collected data from 65 participants who chatted with the well-known chatbot ALICE (Wallace, 2009). Besides the differences in conversation participants, the procedure for participant pairs and participant–chatbot pairs was identical.

Our final model applied to the chatbot data had a kappa value of 0.333 (0.206), with an accuracy of 0.753. The AUC, combined F1 score, and MCC for this model are 0.666, 0.497, and 0.333, respectively. The individual F1 scores for TUT and TRT for the model are 0.496 (0.396 chance) and 0.855. These metrics generally indicate reasonable generalizability of our original model, with comparable performance to other TUT detectors in the literature. At the same time, we do see a drop off in terms of precision/recall compared to our original human–human chat model. As seen in the (Table 10), TRT is predicted with higher accuracy, whereas TUT is predicted 41 out of 83 times (49%).

Table 10 Confusion matrix for the model trained on human dataset and tested on chatbot dataset

	Predicted Values			
	Actual Values	TUT	TRT	Total
15 s	TUT	41	42	83
	TRT	41	212	253
	Total	82	254	336

4 Discussion

4.1 Major findings

Task-unrelated thought occurs when a person's attention is focused toward internally generated thoughts rather that task-at-hand. This is a very common phenomenon in everyday life, but it can be difficult to study unobtrusively because of its elusive nature. A number of studies have attempted to detect TUT with some success, but to date all of them have been focused on single user activities like driving and reading. Ours is the first build a detector of TUT in a multi-person interactive setting, specifically in computer-mediated conversations.

Although a few studies to date have uses keystroke patterns to detective affective states like boredom and engagement (Allen et al., 2016; Bixler & D'Mello, 2013), it was unclear if this modality would be amenable for detecting TUT. TUT is a highly covert and ephemeral state with few overt behavioral markers but mounting evidence that suggests our movement patterns may be indicative of our cognitive states. We thus used keystroke data, along with their associated timestamps, to build models to detect TUT while two participants chatted with one another participant using an online chat platform. Results from these models indicate that people indeed exhibit different keystroke patterns while they are off task versus when they are not, leading to reliable detectors using a random forest algorithm.

Our model also generalized quite well to unseen participants (leave-one-out cross-validation), which is important because each individual has a different typing style, dexterity, and speed. The best model that was created had a kappa value of 0.343 (0.258), which was further improved with feature selection (0.362 SD = 0.255). Feature selection analyses revealed that sentence length features may be a helpful indicator of whether someone is off task; however, it is important to note that no single feature performed well on its own, suggesting that the random forest algorithm relied on a combination of features for classification.

Finally, our model was also tested on an entirely new dataset where users were talking to a chatbot instead of human. This model had a kappa of 0.333, indicating generalizability, but with a drop in both overall performance and recall. Some of the drop in performance may be due to participants realizing that they were talking to a bot, which made them lose interest and/or change their overall chatting behaviors. This above chance result also alleviates concerns about overfitting in search of the original model. Future work will therefore explore the conditions for why and when generalizability is degraded.

We also point out that our model was content independent, meaning we did not use any features to capture the sentiment or meaning of the chats. We view this as a strength of our model, as it can be applied in many different contexts (i.e., education, work, experimental studies tracking TUT in conversations). However, future work should explore content dimensions to improve model fit, especially if the domain is well-defined.

4.2 Limitations

We note a few limitations to the current work. First, we note limitations in the amount of data. Since the performance and generalizability of a model is a function of the data available, more data may have helped the model to yield more accurate results. We also note that data was collected from undergraduate students at a public university in a laboratory setting. The lack of diversity in our sample prevents us from generalizing our results to a larger population. Further, being in an unfamiliar laboratory environment during the conversation may have influenced the dynamics of the conversation and rates of TUT.

Finally, we note the decision to use the self-report method to detect TUT as a limitation to our study. There are two gold-standard ways to assess TUT: self-caught (used here) and probe-caught (Weinstein, 2018). Asking participants to self-report TUT forces participants to remain meta-aware of their own mental state (Smallwood & Schooler, 2006), which is a trade-off to limiting the amount of times TUT can be reported in general (i.e., probes are only periodically inserted). Our results rely on the assumption that participants were honest and accurate with their reports. Unlike probe-caught measures of TUT, the self-report also does not provide us with instances of TRT; testing our models on probe-caught data in the future would thus be beneficial and might further address the issues surrounding a small number of participants making zero TUT reports. This limitation, however, is also one of the reasons why it is important to create "stealth" assessments of TUT in the first place. Future research should prioritize the use of developed detectors to determine whether or not a person is off-task without relying on methods that either place additional task demands on participants (cognitive monitoring in self-caught) or necessitate task interruptions (probes appearing in probe caught). TUT detectors such as the one developed here can assist in this advancement, particularly during computer-mediated communications.

4.3 Future work

As noted above, self-caught TUT can force participants to constantly monitor their thought and mental states. This issue can be circumvented in future work by performing this experiment using the probe-caught method to detect off-task thought. This would give concrete instances of what keystrokes look like when a person is completely focused on a task. Second, future work may consider TUT not as dichotomous, but on a continuum (Kam et al., 2021; Kane et al., 2021; Mills et al., 2018). After all it might be possible to be 'focused' on a less demanding task while still not having thoughts that are completely task-relevant, particularly during conversations. Future work might therefore consider measuring TUT on a continuum using a 7-point Likert scale instead of a binary one. However, it is important to note that Kane et al. (Kane et al., 2021) found that a content-based binary scale is currently the most reliable way to asses TUT (or mind wandering), perhaps owing to the individual differences that influence reporting on a continuous scale. Third, the actual content of the messages could be examined such the affective state of the participant is monitored in tandem with TUT. Finally, future work should consider combining keystrokes with other modalities that

detect TUT. As was demonstrated by Bixler et al. (Bixler et al., 2015), combining multimodal features to detect TUT can result in detection better than one modality alone. Future work thus should consider combining keystrokes with other modalities such as EEG, gaze, and facial features to determine whether this results in more accurate models than those built on keystrokes alone.

References

Allen, L.K., Mills, C., Jacovina, M. E., Crossley, S., D'Mello, S., McNamara, D. S.: Investigating boredom and engagement during writing using multiple sources of information: The essay, the writer, and keystrokes. *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge - LAK '16*. (2016). https://doi.org/10.1145/2883851.2883939

Arch, J.J., Wilcox, R.R., Ives, L.T., Sroloff, A., Andrews-Hanna, J.R.: Off-task thinking among adults with and without social anxiety disorder: An ecological momentary assessment study. Cogn. Emot. **35**(2), 269–281 (2021). https://doi.org/10.1080/02699931.2020.1830751

Baldwin, C.L., Roberts, D.M., Barragan, D., Lee, J.D., Lerner, N., Higgins, J.S.: Detecting and quantifying mind wandering during simulated driving. Front. Hum. Neurosci. (2017). https://doi.org/10.3389/fnhum.2017.00406

Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., Cox, D.D.: Hyperopt: a Python library for model selection and hyperparameter optimization. Comput. Sci. Discov. **8**(1), 014008 (2015). https://doi.org/10.1088/1749-4699/8/1/014008

Bixler, R., D'Mello, S. Detecting boredom and engagement during writing with keystroke analysis, task appraisals, and stable traits. *Proceedings of the 2013 International Conference on Intelligent User Interfaces - IUI '13* (2013). https://doi.org/10.1145/2449396.2449426

Bixler, R., Blanchard, N., Garrison, L., D'Mello, S. Automatic Detection of Mind Wandering During Reading Using Gaze and Physiology. *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction - ICMI '15* (2015). https://doi.org/10.1145/2818346.2820742

Blanchard, N., Bixler, R., Joyce, T., D'Mello, S.: Automated Physiological-Based Detection of Mind Wandering during Learning. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) Intelligent Tutoring Systems, vol. 8474, pp. 55–60. Springer International Publishing, Cham (2014)

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002). https://doi.org/10.1613/jair.953

Critcher, C.R., Gilovich, T.: Inferring attitudes from mindwandering. Pers. Soc. Psychol. Bull. **36**(9), 1255–1266 (2010). https://doi.org/10.1177/0146167210375434

D'Mello, S.K., Mills, C., Bixler, R., Bosch, N. *Zone out no more: Mitigating mind wandering during computerized reading*. 8 (2017).

D'Mello, S., Mills, C.: Emotions while writing about emotional and non-emotional topics. Motiv. Emot. **38**(1), 140–156 (2014). https://doi.org/10.1007/s11031-013-9358-1

D'Mello, S.K., Mills, C.S.: Mind wandering during reading: An interdisciplinary and integrative review of psychological, computing, and intervention research and theory. Lang. Linguist. Compass **15**(4), e12412 (2021). https://doi.org/10.1111/lnc3.12412

Eastwood, J.D., Frischen, A., Fenske, M.J., Smilek, D.: The unengaged mind: defining boredom in terms of attention. Perspect. Psychol. Sci. **7**(5), 482–495 (2012). https://doi.org/10.1177/1745691612456044

Epp, C., Lippold, M., Mandryk, R.L. Identifying emotional states using keystroke dynamics. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 715–724 (2011).https://doi.org/10.1145/1978942.1979046

Faber, M., Bixler, R., D'Mello, S.K.: An automated behavioral measure of mind wandering during computerized reading. Behav. Res. Methods **50**(1), 134–150 (2018). https://doi.org/10.3758/s13428-017-0857-y

Ferreira, H. *Confusion matrix and other metrics in machine learning*. Medium (2018). https://medium.com/hugo-ferreiras-blog/confusion-matrix-and-other-metrics-in-machine-learning-894688cb1c0a

Fox, K.C.R., Andrews-Hanna, J.R., Mills, C., Dixon, M.L., Markovic, J., Thompson, E., Christoff, K.: Affective neuroscience of self-generated thought: affective neuroscience of self-generated thought. Ann. N. Y. Acad. Sci. **1426**(1), 25–51 (2018). https://doi.org/10.1111/nyas.13740

Franklin, M.S., Smallwood, J., Schooler, J.W.: Catching the mind in flight: Using behavioral indices to detect mindless reading in real time. Psychon. Bull. Rev. **18**(5), 992–997 (2011). https://doi.org/10.3758/s13423-011-0109-6

Huang, K., Yeomans, M., Brooks, A.W., Minson, J., Gino, F.: It doesn't hurt to ask: question-asking increases liking. J. Pers. Soc. Psychol. **113**(3), 430–452 (2017). https://doi.org/10.1037/pspi0000097

Hutt, S., Krasich, K., Mills, C., Bosch, N., White, S., Brockmole, J.R., D'Mello, S.K.: Automated gaze-based mind wandering detection during computerized learning in classrooms. User Model. User-Adap. Inter. **29**(4), 821–867 (2019). https://doi.org/10.1007/s11257-019-09228-5

Hutt, S., Mills, C., White, S., Donnelly, P.J., D'Mello, S.K. *The eyes have it: gaze-based detection of mind wandering during learning with an intelligent tutoring system* 8 (2016).

Hutt, S., Hardey, J., Bixler, R., Stewart, A., Risko, E., D'Mello, S.K. *Gaze-based Detection of Mind Wandering during Lecture Viewing* 6 (2017).

Hutt, S., Krasich, K., R. Brockmole, J., K. D'Mello, S. Breaking out of the Lab: Mitigating Mind Wandering with Gaze-Based Attention-Aware Technology in Classrooms. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–14 (2021). https://doi.org/10.1145/3411764.3445269

Kam, J.W.Y., Irving, Z.C., Mills, C., Patel, S., Gopnik, A., Knight, R.T.: Distinct electrophysiological signatures of task-unrelated and dynamic thoughts. Proc. Natl. Acad. Sci. **118**(4), e2011796118 (2021). https://doi.org/10.1073/pnas.2011796118

Kane, M.J., Smeekens, B.A., Meier, M.E., Welhaf, M.S., Phillips, N.E.: Testing the construct validity of competing measurement approaches to probed mind-wandering reports. Behav. Res. Methods (2021). https://doi.org/10.3758/s13428-021-01557-x

Killingsworth, M.A., Gilbert, D.T.: A Wandering mind is an unhappy mind. Science **330**(6006), 932–932 (2010). https://doi.org/10.1126/science.1192439

Kopp, K., Mills, C., D'Mello, S.: Mind wandering during film comprehension: the role of prior knowledge and situational interest. Psychon. Bull. Rev. **23**(3), 842–848 (2016). https://doi.org/10.3758/s13423-015-0936-y

Marchetti, I., Koster, E.H.W., Klinger, E., Alloy, L.B.: Spontaneous thought and vulnerability to mood disorders: the dark side of the wandering mind. Clin. Psychol. Sci. **4**(5), 835–857 (2016). https://doi.org/10.1177/2167702615622383

McHugh, M.L. Interrater reliability: The kappa statistic. *Biochemia Medica*, 276–282 (2012). https://doi.org/10.11613/BM.2012.031

Mills, C., Raffaelli, Q., Irving, Z.C., Stan, D., Christoff, K.: Is an off-task mind a freely-moving mind? Examining the relationship between different dimensions of thought. Conscious. Cogn. **58**, 20–33 (2018). https://doi.org/10.1016/j.concog.2017.10.003

Mills, C., Gregg, J., Bixler, R., D'Mello, S.K.: Eye-mind reader: An intelligent reading interface that promotes long-term comprehension by detecting and responding to mind wandering. Human-Comput. Interact. (2020). https://doi.org/10.1080/07370024.2020.1716762

Mills, C., Porter, A.R., Andrews-Hanna, J.R., Christoff, K., Colby, A.: How task-unrelated and freely moving thought relate to affect: evidence for dissociable patterns in everyday life. Emotion (2021). https://doi.org/10.1037/emo0000849

Mills, C., D'Mello, S. *Toward a real-time (Day) Dreamcatcher: sensor-free detection of mind wandering during online reading*. 8 (2015).

Mills, C., Bixler, R., Wang, X., D'Mello, S.K. *Automatic gaze-based detection of mind wandering during narrative film comprehension*. 8 (2016).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**(85), 2825–2830 (2011)

Pham, P., Wang, J.: AttentiveLearner: Improving Mobile MOOC Learning via Implicit Heart Rate Tracking. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) Artificial Intelligence in Education, vol. 9112, pp. 367–376. Springer International Publishing, Cham (2015)

Raffaelli, Q., Mills, C., Christoff, K.: The knowns and unknowns of boredom: a review of the literature. Exp. Brain Res. **236**(9), 2451–2462 (2018). https://doi.org/10.1007/s00221-017-4922-7

Rayner, K.: Eye movements in reading and information processing: 20 years of research. Psychol. Bull. **124**(3), 372–422 (1998). https://doi.org/10.1037/0033-2909.124.3.372

Reichle, E.D., Pollatsek, A., Fisher, D.L., Rayner, K.: Toward a model of eye movement control in reading. Psychol. Rev. **105**(1), 125–157 (1998). https://doi.org/10.1037/0033-295X.105.1.125

Salmeron-Majadas, S., Santos, O.C., Boticario, J.G.: An evaluation of mouse and keyboard interaction indicators towards non-intrusive and low cost affective modeling in an educational context. Procedia Computer Science **35**, 691–700 (2014). https://doi.org/10.1016/j.procs.2014.08.151

Smallwood, J.: Mind-wandering while reading: attentional decoupling, mindless reading and the cascade model of inattention. Lang. Linguist. Compass **5**(2), 63–77 (2011). https://doi.org/10.1111/j.1749-818X.2010.00263.x

Smallwood, J., Schooler, J.W.: The restless mind. Psychol. Bull. **132**(6), 946–958 (2006). https://doi.org/10.1037/0033-2909.132.6.946

Smallwood, J., Schooler, J.W.: The Science of mind wandering: empirically navigating the stream of consciousness. Annu. Rev. Psychol. **66**(1), 487–518 (2015). https://doi.org/10.1146/annurev-psych-010814-015331

Smallwood, J., Fishman, D.J., Schooler, J.W.: Counting the cost of an absent mind: mind wandering as an underrecognized influence on educational performance. Psychon. Bull. Rev. **14**(2), 230–236 (2007). https://doi.org/10.3758/BF03194057

Snoek, J., Larochelle, H., Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 2951–2959). Curran Associates, Inc (2012). http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf

Stewart, A., Bosch, N., Chen, H., Donnelly, P.J., D'Mello, S.K. Where's Your Mind At?: Video-based mind wandering detection during film viewing. *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization - UMAP '16*, 295–296 (2016). https://doi.org/10.1145/2930238.2930266

Varao-Sousa, T.L., Kingstone, A.: Are mind wandering rates an artifact of the probe-caught method? Using self-caught mind wandering in the classroom to test, and reject, this possibility. Behav. Res. Methods **51**(1), 235–242 (2019). https://doi.org/10.3758/s13428-018-1073-0

Wallace, R.S.: The Anatomy of A.L.I.C.E. In: Epstein, R., Roberts, G., Beber, G. (eds.) Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer, pp. 181–210. Springer, Netherlands, Dordrecht (2009)

Weinstein, Y.: Mind-wandering, how do I measure thee with probes? Let me count the ways. Behav. Res. Methods **50**(2), 642–661 (2018). https://doi.org/10.3758/s13428-017-0891-9

Wengelin, Å., Torrance, M., Holmqvist, K., Simpson, S., Galbraith, D., Johansson, V., Johansson, R.: Combined eyetracking and keystroke-logging methods for studying cognitive processes in text production. Behav. Res. Methods **41**(2), 337–351 (2009). https://doi.org/10.3758/BRM.41.2.337

Wolf, E.B., Lee, J.J., Sah, S., Brooks, A.W.: Managing perceptions of distress at work: reframing emotion as passion. Organ. Behav. Hum. Decis. Process. **137**, 1–12 (2016). https://doi.org/10.1016/j.obhdp.2016.07.003

Vishal Kuvar is a Ph.D. student in the Educational Psychology department at the University of Minnesota. His research focuses on understanding the shift of attention in various contexts like conversation, and education. His research interests also include eye tracking, and virtual and augmented reality.

Nathaniel Blanchard is an Assistant Professor of Computer Science at Colorado State University. He specializes in machine learning for human-AI collaboration. His lab has produced state-of-the-art machine learning models across a multitude of domains including vision, acoustic, and natural language. Ultimately, his research focuses on the application of these models in real-world contexts. Recently, he received best student paper at the Winter conference for Applications of Computer Vision (WACV) 2021.

Alexander Colby is a Research Analyst at the Massachusetts Cannabis Control Commission. His research uses available data to assess consumer and industry behaviors in the context of ever-evolving state and federal cannabis policies. His current project uses industry data to understand how heterogeneity in states' cannabis policies contribute to variability in patterns of purchasing and use.

Laura Allen is an Assistant Professor of Educational Psychology at the University of Minnesota. The principal aim of her research has been to theoretically and empirically investigate the higher-level cognitive skills that are required for successful text comprehension and production, as well as the ways in which performance in these domains can be enhanced through strategy instruction and training. This research has been accompanied by a second line of work that explores how technologies can be developed and leveraged to facilitate learning and training. The overall goal of this research is to develop technologies and computational methodologies that will have a broad impact on current practices in research and instruction across multiple dimensions.

Caitlin Mills (PhD in Cognitive Psychology) is an Assistant Professor in Educational Psychology at the University of Minnesota. Her research primarily focuses on constructs related to mind wandering, boredom, and engagement. She combines ecologically relevant paradigms with computational methods to characterize when mind wandering occurs and how it influences learning. Other ongoing research interests include how the dynamics of our internal attention influence functional aspects of our everyday lives, such as affective valence, boredom, and creativity.