THESIS

AUTOMATICALLY DETECTING TASK UNRELATED THOUGHTS DURING CONVERSATIONS USING KEYSTROKE ANALYSIS

Submitted by

Vishal Kiran Kuvar

Department of Computer Science

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Summer 2022

Master's Committee:

Advisor: Nathaniel Blanchard
Co-Advisor: Caitlin Mills

Asa Ben-Hur
Wen Zhou

ABSTRACT

AUTOMATICALLY DETECTING TASK UNRELATED THOUGHTS DURING CONVERSATIONS USING KEYSTROKE ANALYSIS

Task-unrelated thought (TUT), commonly known as the phenomenon of daydreaming or zoning-out, is a mental state where a person's attention moves away from the task-at-hand to self-generated thoughts. This state is extremely common yet not much is known about it during dyadic interactions. We built a model to detect when a person experiences TUTs while talking to another person through a chat platform, by analyzing their keystroke patterns. This model was able to differentiate between task-unrelated thoughts and task-related thoughts with a kappa of 0.343. This serves as a strong indicator that typing behavior is linked with mental states, task-unrelated thoughts in our case.

ACKNOWLEDGEMENTS

I would like to thank Dr. Nathaniel Blanchard and my Dr. Caitlin Mills for their invaluable guidance.

DEDICATION

To my parents,

I could not have been able to do this without your constant love and support.

Thank you.

TABLE OF CONTENTS

ABSTRACT ii
ACKNOWLEDGEMENTS iii
DEDICATION iv
LIST OF TABLES vi
LIST OF FIGURES vii

Chapter 1 Introduction 1

Chapter 2 Related work 3
2.1 Task Unrelated Thoughts 3
2.2 Keystrokes as an indicator of cognitive processing 5
2.3 Current Study 6

Chapter 3 Methods 7
3.1 Participants and Data Collection 7
3.1.1 Participants 7
3.1.2 Chat Session 7
3.1.3 TUT reports 8
3.2 Supervised Machine Learning 8
3.2.1 Data Preprocessing 8
3.2.2 Feature Extraction 9
3.2.3 Validation and Evaluation 13
3.2.4 Model Building 16

Chapter 4 Results 18
4.1 Best Models 18
4.2 Feature Analysis 19
4.3 Generalizing on new data: Chatbot conversations 24

Chapter 5 Discussion 26
5.1 Major Findings 26
5.2 Comparison with other studies 27
5.3 Limitations 27
5.4 Future Work 28

Bibliography 30

LIST OF TABLES

3.1 TUT and TRT distribution. 11
3.2 Description of extracted features. 14

4.1 Kappa values for best models. 18
4.2 Performance metrics for the best models. 18
4.3 Confusion matrices. 19
4.4 Feature Descriptves. 22
4.5 Feature groups. 23
4.6 Results with feature groups. 23
4.7 Results with individual features. 24
4.8 Confusion matrix for chatbot dataset. 25

5.1 Comparison with other studies. 29

LIST OF FIGURES

3.1 Window Creation. 10

4.1 Kappa distribution. 20
4.2 Kappa values against number of reports. 21

Chapter 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 [1]. TUT, which is often used synonymously with the term mind-wandering [2], occurs in virtually every scenario of our lives while we are awake [3] – 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, clinical, and behavioral correlates. For instance, TUT can impair retention of recorded lectures [4] and reading comprehension [5]. At the same time, TUT may be positive in other cases; in minimally demanding tasks, for instance, TUT facilitates creative thought [6] and is associated with less impulsive decision making [7].

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. The contribution of this work is two-fold: First, 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 [8–10]. 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 towards these goals, which is to identify effective low-cost methods for TUT detection in interactive contexts.

Chapter 2

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).

2.1 Task Unrelated Thoughts

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 of 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 processes [11,12]. Many researchers have utilized gaze behaviors for detecting TUT during reading, as eye movements provide a reliable window into cognitive processing [13, 14]. For instance, Faber et al. [15] 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. [16] 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. [17] 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. [18] and Mills et al. [19] 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 [9, 20] narrative films [21, 22], and simulated driving [23]. Hutt et al. [24, 25], for example, detected TUT in more dynamic learning scenarios using gaze as their primary modality. In one study [24], 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 [26] 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 [21] or facial features and body movements [22].

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 in today's society, and even more so with more education and workplaces shifting to dominantly remote, online interactions. 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 [23]. 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.

2.2 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 [27,28]. 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 [27, 29].

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 [27, 28]. In the first study, Bixler & D'Mello [27] 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 [30]. 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. [28] 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.

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 of these 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 do not constitute the same construct. Whereas boredom is inherently defined by its affective components (negative feelings; [31]), TUT does not share a one-to-one mapping with affect [32]. Similarly, the relationship between boredom and TUT is not yet conceptually linked [31, 33, 34]. 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.

2.3 Current Study

We attempt to build a model that detects 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. 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.

Chapter 3

Methods

3.1 Participants and Data Collection

3.1.1 Participants

All methods and procedures were approved by the IRB at [removed for blind review] in and accordance with the Declaration of Helsinki. Data was 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 was corrupted, leaving us 120 participants' data to work with.

Participants came to the laboratory to take part in a study involving a 10-minute 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.

3.1.2 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 [35], [36]. 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

3.1.3 TUT reports

We used a commonly employed and validated technique to collect TUT reports, referred to as a "self-caught" method [35, 36] 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 frequently used when attempting TUT detection [22, 23, 37] 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.

3.2 Supervised Machine Learning

3.2.1 Data Preprocessing

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. The absence of TUT may also indicate that participants misunderstood or forgot the instructions. 11 participants out of the 120 did not report any TUT and were dropped. In total, we were left with 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.

3.2.2 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. 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 Figure 3.1, and is similar to approaches used in previous studies [22]. We wrote a program to traverse each CSV file until a TUT report was identified. In order to avoid unintentionally capturing TUT report patterns (i.e., artificial pauses), we adopted a TUT report “buffer” approach, which is commonly used in other detection attempts [14, 22]. Specifically, we discarded 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 (15s, 30s) preceding the buffer. The 2 window sizes were chosen to determine how much keystroke data is needed for TUT detection in conversational contexts. All the keystrokes that fell into a given window were used for feature extraction.

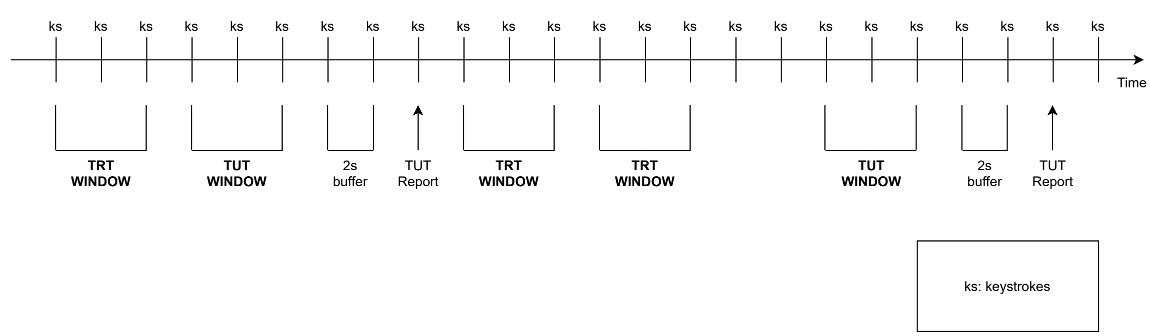


Figure 3.1: Window creation for TUT and TRT.

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., 15s, 30s), we created as many TRT windows as possible. As an example, if participants did not engage in TUT for over 2 minutes, we would be able to create a maximum of 8 TRT windows for the 15s model (provided all the windows are exactly 15s). 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. In case no TRT windows were created for a participant, that participant would be dropped. There was 1 such case for the 15s window and 2 for the 30s window.

It is also important to point out that TRT will be much more common than TUT. This class imbalance issue is true for previous TUT detection attempts as well. 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 distribution of TUT and TRT instances for both 15 and 30s windows is shown in Table 3.1

Table 3.1: Distribution of TUTs and TRTs for 15s and 30s windows

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

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 [27]. 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: 0.25s - 0.75s; 0.75s. - 1.25s; 1.25s - 1.75s; 1.75s - 2.25s; 2.25s - 2.75s.

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. If not, 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 3.2 summarizes the different features that were extracted.

3.2.3 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: 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

Table 3.2: Description of extracted features

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

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 [38] 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 random 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 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. This could become problematic if the dataset is imbalanced, like in our case. The metrics are defined as follows [38, 39]:

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

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

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

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

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

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

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

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

3.2.4 Model Building

As discussed in our Validation section, we used the 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) [40] 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 [15].

We also test performance across two different window sizes: 15s and 30s, 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, 15s may perform the best because the data is more proximal to the TUT report; in contrast 30s 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: 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 [41,42]. Hyperopt implements Bayesian optimization [43] 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 for a set number of times and the model with the highest kappa at the end of this process was selected.

Chapter 4

Results

4.1 Best Models

Table 4.1 shows the best kappa values and its standard deviation for both the windows with normal data as well as SMOTE data. This table shows that the 15s window give better results compared to the 30s window. Within that 15s window, using SMOTE provides us with better results. SMOTE also gives better results for the 30s window as compared to the 30s window without SMOTE.

Table 4.1: Kappa values for the best models for 15s and 30s window

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

We list the remaining performance metrics of the best models in Table 4.2. Here we only include the 15s and 30s window models that used SMOTE. Both the windows use the random forest classifier to train the best model.

Table 4.2: Performance metrics for 15s and 30s windows trained with SMOTE.

Window size	Classifier	Accuracy	ROC-AUC	F1	MCC
15s	Random forest	0.775	0.692	0.508	0.372
30s	Random forest	0.730	0.690	0.558	0.373

The confusion matrices for the best models for each window are shown in Table 4.3. Even though the dataset is imbalanced, the models do not predict one class over the other.

Table 4.3: Confusion matrices for each window.

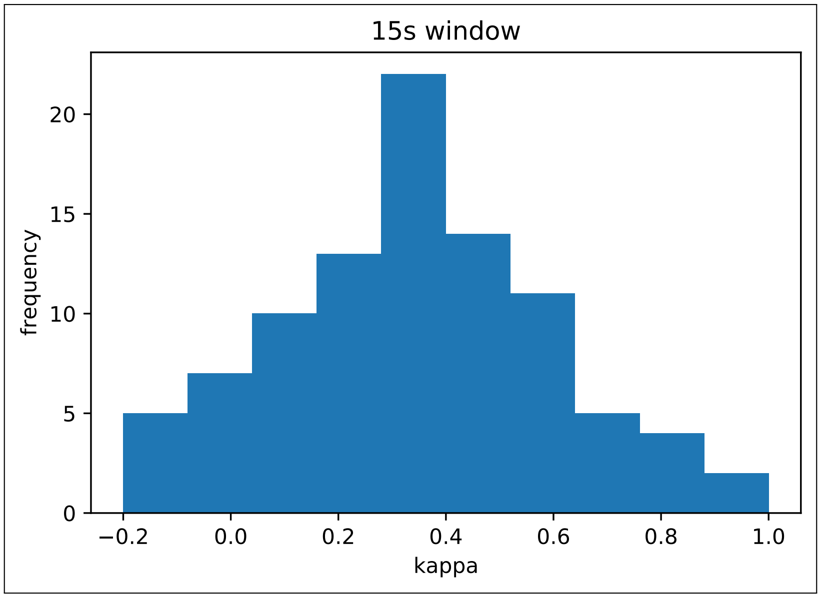
		Predicted Values		
	Actual Values	TUT	TRT	Total
15s	TUT (0.20)	199	175	374
	TRT (0.79)	210	1215	1425
	Total	409	1390	1799
30s	TUT (0.30)	140	103	243
	TRT (0.69)	119	489	608
	Total	259	592	851

We also investigated 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. Such a model would have an overall decent kappa value but would not have generalized well on new data; ideally, the models' kappa values would have a normal distribution. Figure 4.1 shows the distribution of kappa values.

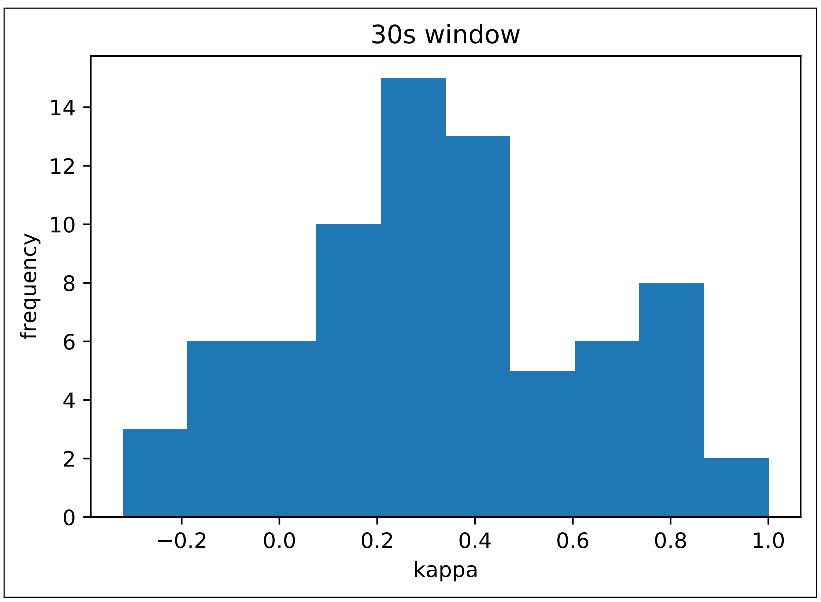
Lastly, we plotted the kappa values against the number of TUT reports. We wanted to inspect the possibility if a high kappa value was a function of the number of reports. Figure 4.2 shows the plot with number of reports on the X axis and kappa values on the Y axis. This plot clearly indicates that the kappa values in independent of the number of TUT reports.

4.2 Feature Analysis

An important question is how each feature contributed to our final TUT classifier. We created a feature descriptive table that calculated the mean and standard deviation for TUT and TRT instances for each feature. This helped us determine if there was an apparent pattern in which the



(a) Kappa distribution for 15s model.



(b) Kappa distribution for 30s model.

Figure 4.1: Kappa distribution values of (a) 15s model (top), (b) 30s model (bottom).

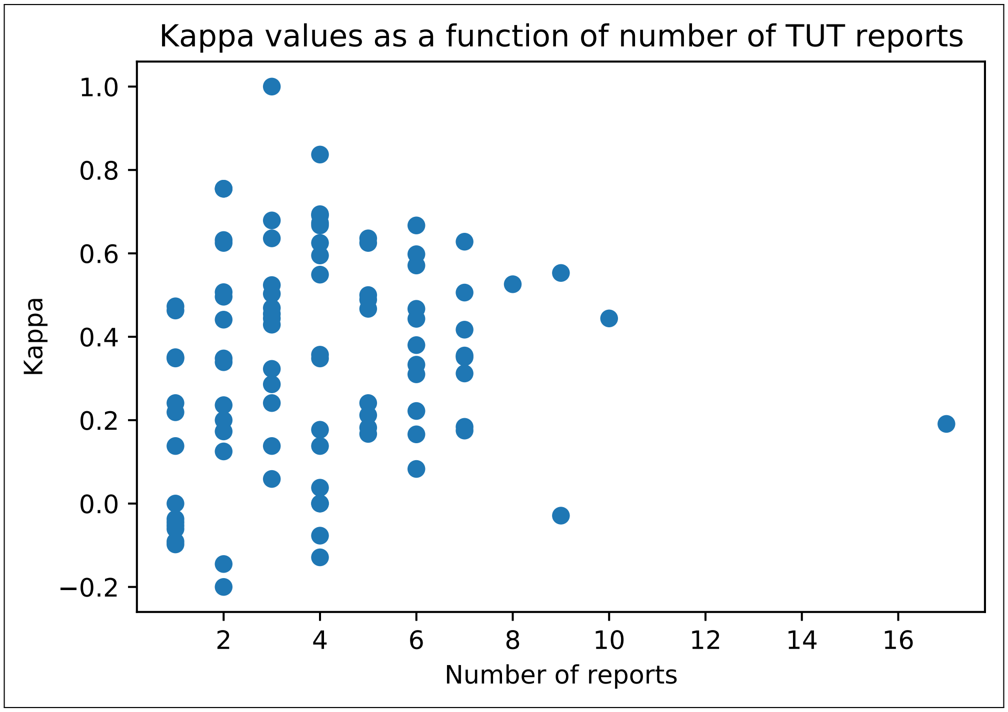


Figure 4.2: Distribution of kappa values against number of TUT reports.

models discriminated between two classes. We also performed a paired samples t-tests and calculated Cohen's d to compare each feature (see Table 4.4). A paired samples t-test is a statistical method that determines if there is any statistical difference between populations, which in our case is the TUT and TRT classes. A Cohen's d is used to see if the difference between the classes in distinguishable by the "naked eye", which means if the difference is immediately apparent. The table consists of features for whom the p value is less than 0.05.

To investigate how each feature group influences the models, we created models that were trained on individual feature group as well as models that were trained my withholding one feature group at time. The features included in each feature group can be seen in Table 4.5.

Matching our best models, the machine learning algorithm used was the random forest classifier. The hyperparameters in this classifier were set to be the same as the best model for the 15s window. The kappa values of these models can be seen in Table 4.6.

The feature group of Sentence length looks like the biggest influence on the model with the highest kappa value of 0.310 when trained on individually and the biggest drop in the kappa when

Table 4.4: Feature descriptive table for TUT and TRT for the 15s window.

	TRT (N=1452)		TUT (N=374)			
Features	Mean	SD	Mean	SD	p	d
Verbosity	38.1	9.14	44.8	14.6	< 0.001	0.466
Maximum latency	15.8	8.1	11.3	12.4	< 0.001	0.353
Minimum latency	2.81	6.01	0.577	2.42	< 0.001	0.338
Mean latency	4.22	6.24	1.56	3.39	< 0.001	0.383
Median latency	3.38	6.29	0.936	3.08	< 0.001	0.358
0.25s – 0.75s pause	6.97	1.82	8.15	3.42	< 0.001	0.371
Length of message	27.0	7.53	34.2	14.0	< 0.001	0.541
Number of words	4.03	1.18	5.10	2.12	< 0.001	0.506
Number of sentences	1.16	0.296	1.40	0.480	< 0.001	0.444
Number of messages	1.44	0.325	1.20	0.416	< 0.001	0.616
Inter-sentence time	0.974	1.68	0.39	1.03	< 0.001	0.345
Minimum word length (keystroke)	3.13	2.62	2.42	1.96	0.024	0.220
Mean word length (keystroke)	6.80	2.94	5.95	2.25	0.015	0.238
Maximum sentence length (keystroke)	33.1	8.03	37.2	13.2	0.002	0.302
Minimum sentence length (keystroke)	20.9	8.81	28.2	14.7	< 0.001	0.440
Mean sentence length (keystroke)	26.9	7.84	32.6	12.8	< 0.001	0.417
Median sentence length (keystroke)	26.6	7.85	32.5	12.8	< 0.001	0.429
Maximum sentence length (timestamp)	13.4	6.25	16.6	10.9	0.012	0.246
Minimum sentence length (timestamp)	9.55	6.66	13.3	11.8	0.004	0.282
Maximum sentence length (word)	5.01	1.34	5.83	2.52	< 0.001	0.339
Minimum sentence length (word)	3.28	1.29	4.56	2.47	< 0.001	0.494
Mean sentence length (word)	4.12	1.23	5.19	2.32	< 0.001	0.467
Median sentence length (word)	4.08	1.23	5.17	2.32	< 0.001	0.483

Table 4.5: Feature groups.

Feature Group	Features included
Length	Verbosity, length of actual message
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
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 4.6: 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	-

it is removed. On the opposite side of this spectrum lies the feature group of Pauses which seems to have the lowest influence on the overall model. One feature group that stands out in particular is Word length. While it has the second highest kappa of the models trained on individual feature group, the overall kappa of the model goes up when this feature is removed.

To look at each individual feature, we repeated this same procedure on training on one and removing one on each feature. The kappa values of the top 10 models are reported in Table 4.7 (Each feature can be interpreted as follows: feature group, measured as, statistic).

Table 4.7: Results of models trained on one feature and by removing one feature.

Features	Trained on one feature	Removed one feature
Sentence length, word, 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	-

Based on the results shown in Table 4.6 and Table 4.7, we can infer that the model is driven by 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.

4.3 Generalizing on new data: Chatbot conversations

We wanted to know if a model that can detect TUT in a two-person conversation can also detect the same for a participant who conversed with a chatbot. In order to do so, we created two separate datasets. The first one, titled human dataset, containing all the keystroke data files for participants who talked to other participants and the second one, titled the chatbot dataset, containing files for participants who conversed with a chatbot. Data was cleaned and extracted from both of these datasets. The data from the human dataset was used for training and the data from the chatbot dataset was used for testing. We used the best classifier and hyperparameters from the two-person conversation model to train this data. The 15s window was used since it was the one that gave the best results. SMOTE was used on the training set to create a balanced dataset.

We used data that was collected from the same set-up. That is, in addition to collecting data from participant-participant pairs, we also collected data in which participants conversed with a chatbot. Participants conversed with ALICE when only one of two sign up slots for the study was filled, or one of the two participants neglected to show up for their timeslot. We collected data from 65 participants who chatted with the well-known chatbot ALICE [44]. Besides the differences in conversation participants, the procedure for participant pairs and participant-chatbot pairs was identical.

The model created had a kappa value of 0.333 (0.206), with an accuracy of 0.753. The AUC, F1 score and MCC for this model are 0.666, 0.497 and 0.333, respectively. While this model has a high accuracy, it did not perform very well in differentiating between the TUT. As seen in the table below, it is able to detect TRT with a high accuracy but does is only able to detect TUT 41 out of 83 times. Hence, this model won't be able to detect TUT with a very high accuracy.

Table 4.8: Confusion matrix for the model trained on human dataset and tested on chatbot dataset.

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

Chapter 5

Discussion

5.1 Major Findings

Task-unrelated thought in a state of mind where a person's attention is focused towards internally generated thoughts rather that task-at-hand. This is a very common pheonomenon but is difficult to study because of its elusive nature. A number of studies have been performed to detect TUT states and all of them have been focused on single user activities like driving and reading. Ours is the first of its kind study that tries to detect TUT in a multi-person setting, specifically computer-mediated conversation. The hypothesis for this experiment that the keystroke pattern of a person when they are on-task is significantly different from when they are off-task.

Keystroke data, along with the timestamp, was gathered from participants who were asked to chat with another participant using an online chat platform. Task-unrelated thought and task-related thought windows were created on the keystroke data. Two window sizes of 15s and 30s were used for this experiment. A number of features were extracted in each window and models were trained on the extracted features. The results from these models seem to support our hypothesis and indicate that people exhibit different keystroke patterns while they are off-task versus when they are not. The developed models generalize well on unseen data, 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). The confusion matrix generated indicated that the model did not tend to over-predict one class over other. This is most likely because SMOTE was used on the training data, creating a balanced dataset.

We wanted to investigate the contribution of each individual feature group. For this we created models that were trained on one feature group and models that were created by eliminating one feature group. We found that sentence length is the biggest contributor in the model with its individual model having a kappa of 0.310, and kappa dropping to 0.269 when that feature group was removed. A sentence can be considered to be the end of a thought. This gives the participant's mind an opportunity either to go into the TUT state or continue the thought in the next sentence. This could be a reason as to why sentence length is a clear indicator between on and off-task thoughts.

We did the same experiment with individual features next. As a result of this experiment, we found that individual features provide little contribution to the model's performance. But when these features come together to form a feature group, their contribution becomes significant.

Lastly, we wanted to see if the model that detects TUT in a participant talking to another human could also detect TUT when a person is talking to a chatbot. For this we used the human dataset and the training set and the chatbot dataset as testing. This model had a kappa of 0.333. But delving into the confusion matrix, we realized that the model was not able to predict TUT very well. It might be possible that the participants soon realized that they were talking to a bot, which made them lose interest. This could have made it so that the TUT when talking to a human was very different from the TUT when talking to a bot.

5.2 Comparison with other studies

Here, we compare the results of our study with other studies detecting TUT. The other studies use a number of different modalities and settings. All studies listed below use the leave-one-participant-out or the leave-multiple-participant-out validation methods similar to our own study. Table 5.1 shows the best results obtained in each study.

5.3 Limitations

We note a few different limitations to the current study. First, we note limitations in the amount of data we extracted features from. 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. Asking participants to self-report TUT forces participants to remain meta-aware of their own mental states, which has caused researchers to question their accuracy [47]. Our results thus rely on the assumption that participants were honest and accurate with their reports. Unlike probe-caught measures of TUT, the self-report does not provide us with instances of TRT.

5.4 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 should consider TUT not as dichotomous, but on a continuum. After all it is possible to be 'focused' on a less demanding task while still not having thoughts that are completely task-relevant. Future work should measure TUT on a continuum using a 7-point Likert scale instead of a binary one. Third, the actual content of the text could be looked at and it could be a good indication of the affective state of the participant. It would be interesting to see if the way a person types out their thoughts could be an indicator of TUT. Finally, future work should consider combining keystrokes with other modalities that detect TUT. As was demonstrated by Bixler et al. [17], 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.

Table 5.1: Comparison of results to other studies.

Authors	Setting	Modality	Kappa	F1-score
Bixler et al. (2015) [17]	Reading	Gaze and physiological features	0.19	-
Mills et al. (2016) [21]	Film viewing	Eye gaze	-	0.59
Zhang et al. (2018) [45]	Driving	Driving performance	0.384	-
Hutt et al. (2019) [25]	Learning	Eye gaze	-	0.61
Liu et al. (2020) [46]	Sustained Attention to Response Task	fNIRS	-	0.74
Kuvar et al.	Conversation	Keystrokes	0.343	0.508

Bibliography

[1] Jonathan Smallwood and Jonathan W. Schooler. The Science of Mind Wandering: Empirically Navigating the Stream of Consciousness. *Annual Review of Psychology*, 66(1):487–518, January 2015.

[2] Caitlin Mills, Quentin Raffaelli, Zachary C. Irving, Dylan Stan, and Kalina Christoff. Is an off-task mind a freely-moving mind? Examining the relationship between different dimensions of thought. *Consciousness and Cognition*, 58:20–33, February 2018.

[3] Matthew A. Killingsworth and Daniel T. Gilbert. A Wandering Mind Is an Unhappy Mind. *Science*, 330(6006):932–932, November 2010. Publisher: American Association for the Advancement of Science Section: Brevia.

[4] Evan F. Risko, Nicola Anderson, Amara Sarwal, Megan Engelhardt, and Alan Kingstone. Everyday attention: Variation in mind wandering and memory in a lecture. *Applied Cognitive Psychology*, 26(2):234–242, 2012. Place: US Publisher: John Wiley & Sons.

[5] Shi Feng, Sidney D'Mello, and Arthur C. Graesser. Mind wandering while reading easy and difficult texts. *Psychonomic Bulletin & Review*, 20(3):586–592, June 2013.

[6] Benjamin Baird, Jonathan Smallwood, Michael D. Mrazek, Julia W. Y. Kam, Michael S. Franklin, and Jonathan W. Schooler. Inspired by Distraction: Mind Wandering Facilitates Creative Incubation. *Psychological Science*, 23(10):1117–1122, October 2012. Publisher: SAGE Publications Inc.

[7] Jonathan Smallwood, Florence J.M. Ruby, and Tania Singer. Letting go of the present: Mind-wandering is associated with reduced delay discounting. *Consciousness and Cognition*, 22(1):1–7, March 2013.

[8] Sidney K D'Mello, Caitlin Mills, Robert Bixler, and Nigel Bosch. Zone out no more: Mitigating mind wandering during computerized reading. page 8.

[9] Stephen Hutt, Kristina Krasich, James R. Brockmole, and Sidney K. D'Mello. Breaking out of the Lab: Mitigating Mind Wandering with Gaze-Based Attention-Aware Technology in Classrooms. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–14, Yokohama Japan, May 2021. ACM.

[10] Caitlin Mills, Julie Gregg, Robert Bixler, and Sidney K. D'Mello. Eye-mind reader: An intelligent reading interface that promotes long-term comprehension by detecting and responding to mind wandering. *Human-Computer Interaction*, pages No Pagination Specified–No Pagination Specified, 2020. Place: United Kingdom Publisher: Taylor & Francis.

[11] Jonathan Smallwood. Mind-wandering While Reading: Attentional Decoupling, Mindless Reading and the Cascade Model of Inattention. *Language and Linguistics Compass*, 5(2):63–77, 2011. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1749-818X.2010.00263.x.

[12] Jonathan Smallwood, Daniel J. Fishman, and Jonathan W. Schooler. Counting the cost of an absent mind: Mind wandering as an underrecognized influence on educational performance. *Psychonomic Bulletin & Review*, 14(2):230–236, April 2007.

[13] Erik D. Reichle, Alexander Pollatsek, Donald L. Fisher, and Keith Rayner. Toward a model of eye movement control in reading. *Psychological Review*, 105(1):125–157, 1998.

[14] Keith Rayner. Eye Movements in Reading and Information Processing: 20 Years of Research. *EYE MOVEMENTS IN READING*, page 51.

[15] Myrthe Faber, Robert Bixler, and Sidney K. D'Mello. An automated behavioral measure of mind wandering during computerized reading. *Behavior Research Methods*, 50(1):134–150, February 2018.

[16] Nathaniel Blanchard, Robert Bixler, Tera Joyce, and Sidney D'Mello. Automated Physiological-Based Detection of Mind Wandering during Learning. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Alfred Kobsa, Friedemann Mattern, John C.

Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Demetri Terzopoulos, Doug Tygar, Gerhard Weikum, Stefan Trausan-Matu, Kristy Elizabeth Boyer, Martha Crosby, and Kitty Panourgia, editors, *Intelligent Tutoring Systems*, volume 8474, pages 55–60. Springer International Publishing, Cham, 2014. Series Title: Lecture Notes in Computer Science.

[17] Robert Bixler, Nathaniel Blanchard, Luke Garrison, and Sidney D'Mello. Automatic Detection of Mind Wandering During Reading Using Gaze and Physiology. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction - ICMI '15*, pages 299–306, Seattle, Washington, USA, 2015. ACM Press.

[18] Michael S. Franklin, Jonathan Smallwood, and Jonathan W. Schooler. Catching the mind in flight: Using behavioral indices to detect mindless reading in real time. *Psychonomic Bulletin & Review*, 18(5):992–997, October 2011.

[19] Caitlin Mills and Sidney D'Mello. Toward a Real-time (Day) Dreamcatcher: Sensor-Free Detection of Mind Wandering During Online Reading. page 8.

[20] Stephen Hutt, Caitlin Mills, Shelby White, Patrick J Donnelly, and Sidney K D'Mello. The Eyes Have It: Gaze-based Detection of Mind Wandering during Learning with an Intelligent Tutoring System. page 8.

[21] Caitlin Mills, Robert Bixler, Xinyi Wang, and Sidney K D'Mello. Automatic Gaze-Based Detection of Mind Wandering during Narrative Film Comprehension. page 8.

[22] Angela Stewart, Nigel Bosch, Huili Chen, Patrick J. Donnelly, and Sidney K. D'Mello. Where's Your Mind At?: Video-Based Mind Wandering Detection During Film Viewing. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization - UMAP '16*, pages 295–296, Halifax, Nova Scotia, Canada, 2016. ACM Press.

[23] Carryl L. Baldwin, Daniel M. Roberts, Daniela Barragan, John D. Lee, Neil Lerner, and James S. Higgins. Detecting and Quantifying Mind Wandering during Simulated Driving. *Frontiers in Human Neuroscience*, 11, 2017. Publisher: Frontiers.

[24] Stephen Hutt, Jessica Hardey, Robert Bixler, Angela Stewart, Evan Risko, and Sidney K D'Mello. Gaze-based Detection of Mind Wandering during Lecture Viewing. page 6.

[25] Stephen Hutt, Kristina Krasich, Caitlin Mills, Nigel Bosch, Shelby White, James R. Brockmole, and Sidney K. D'Mello. Automated gaze-based mind wandering detection during computerized learning in classrooms. *User Modeling and User-Adapted Interaction*, 29(4):821–867, September 2019.

[26] Phuong Pham and Jingtao Wang. AttentiveLearner: Improving Mobile MOOC Learning via Implicit Heart Rate Tracking. In Cristina Conati, Neil Heffernan, Antonija Mitrovic, and M. Felisa Verdejo, editors, *Artificial Intelligence in Education*, volume 9112, pages 367–376. Springer International Publishing, Cham, 2015. Series Title: Lecture Notes in Computer Science.

[27] Robert Bixler and Sidney D'Mello. Detecting boredom and engagement during writing with keystroke analysis, task appraisals, and stable traits. In *Proceedings of the 2013 international conference on Intelligent user interfaces - IUI '13*, page 225, Santa Monica, California, USA, 2013. ACM Press.

[28] Laura K. Allen, Caitlin Mills, Matthew E. Jacovina, Scott Crossley, Sidney D'Mello, and Danielle S. McNamara. Investigating boredom and engagement during writing using multiple sources of information: the essay, the writer, and keystrokes. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge - LAK '16*, pages 114–123, Edinburgh, United Kingdom, 2016. ACM Press.

[29] Åsa Wengelin, Mark Torrance, Kenneth Holmqvist, Sol Simpson, David Galbraith, Victoria Johansson, and Roger Johansson. Combined eyetracking and keystroke-logging methods for

studying cognitive processes in text production. *Behavior Research Methods*, 41(2):337–351, May 2009.

[30] Sidney D'Mello and Caitlin Mills. Emotions while writing about emotional and non-emotional topics. *Motivation and Emotion*, 38(1):140–156, February 2014.

[31] John D. Eastwood, Alexandra Frischen, Mark J. Fenske, and Daniel Smilek. The Unengaged Mind: Defining Boredom in Terms of Attention. *Perspectives on Psychological Science*, 7(5):482–495, September 2012.

[32] Kieran C.R. Fox, Jessica R. Andrews-Hanna, Caitlin Mills, Matthew L. Dixon, Jelena Markovic, Evan Thompson, and Kalina Christoff. Affective neuroscience of self-generated thought: Affective neuroscience of self-generated thought. *Annals of the New York Academy of Sciences*, 1426(1):25–51, August 2018.

[33] Quentin Raffaelli, Caitlin Mills, and Kalina Christoff. The knowns and unknowns of boredom: a review of the literature. *Experimental Brain Research*, 236(9):2451–2462, September 2018.

[34] Clayton R. Critcher and Thomas Gilovich. Inferring Attitudes From Mindwandering. *Personality and Social Psychology Bulletin*, 36(9):1255–1266, September 2010.

[35] Yana Weinstein. Mind-wandering, how do I measure thee with probes? Let me count the ways. *Behavior Research Methods*, 50(2):642–661, April 2018.

[36] Trish L. Varao-Sousa and Alan Kingstone. 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. *Behavior Research Methods*, 51(1):235–242, February 2019.

[37] Kristopher Kopp, Caitlin Mills, and Sidney D'Mello. Mind wandering during film comprehension: The role of prior knowledge and situational interest. *Psychonomic Bulletin & Review*, 23(3):842–848, June 2016.

[38] Amrinder Arora. Confusion Matrix – Another Single Value Metric – Kappa Statistic | Software Journal.

[39] Hugo Ferreira. Confusion matrix and other metrics in machine learning, April 2018.

[40] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, June 2002. arXiv: 1106.1813.

[41] J Bergstra, D Yamins, and D D Cox. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. page 9.

[42] James Bergstra, Dan Yamins, and David D. Cox. *Hyperopt: A Python Library for Optimizing the Hyperparameters of Machine Learning Algorithms*.

[43] Wei Wang. Bayesian Optimization Concept Explained in Layman Terms, April 2020.

[44] Richard S. Wallace. The Anatomy of A.L.I.C.E. In Robert Epstein, Gary Roberts, and Grace Beber, editors, *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*, pages 181–210. Springer Netherlands, Dordrecht, 2009.

[45] Yuyu Zhang and Takatsune Kumada. Automatic detection of mind wandering in a simulated driving task with behavioral measures. *PLOS ONE*, 13(11):e0207092, November 2018.

[46] Ruixue Liu, Erin Walker, Leah Friedman, Catherine M. Arrington, and Erin T. Solovey. fNIRS-based classification of mind-wandering with personalized window selection for multimodal learning interfaces. *Journal on Multimodal User Interfaces*, June 2020.

[47] Jonathan Smallwood and Jonathan Schooler. The Restless Mind. *Psychological bulletin*, 132:946–58, December 2006.