Automatic Detection of Usability Problem Encounters in Think-aloud Sessions

MINGMING FAN, School of Information, Rochester Institute of Technology, NY, USA
YUE LI, Department of Computer Science, University of Toronto, ON, Canada
KHAI N. TRUONG, Department of Computer Science, University of Toronto, ON, Canada

Think-aloud protocols are a highly valued usability testing method for identifying usability problems. Despite the value of conducting think-aloud usability test sessions, analyzing think-aloud sessions is often time-consuming and labor-intensive. Consequently, previous research has urged the community to develop techniques to support fast-paced analysis. In this work, we took the first step to design and evaluate machine learning (ML) models to automatically detect usability problem encounters based on users' verbalization and speech features in think-aloud sessions. Inspired by recent research that shows subtle patterns in users' verbalizations and speech features tend to occur when they encounter problems, we examined whether these patterns can be utilized to improve the automatic detection of usability problems. We first conducted and recorded think-aloud sessions and then examined the effect of different input features, ML models, test products, and users on usability problem encounters detection. Our work uncovers several technical and user interface design challenges and sets a baseline for automating usability problem detection and integrating such automation into UX practitioners' workflow.

CCS Concepts: • **Information systems** • **Human-centered computing**

Additional Key Words and Phrases: Think aloud, usability problem, verbalization, speech features, machine learning, user experience (UX), AI-assisted UX analysis method

ACM Reference format:
Mingming Fan, Yue Li, and Khai N. Truong. 2020. Automatic Detection of Usability Problem Encounters in Think-aloud Sessions. *ACM Trans. Interact. Intell. Syst.* 10, 2, Article 16 (May 2020), 24 pages.
https://doi.org/10.1145/3385732

1 INTRODUCTION

Think-aloud protocols are often used and valued by user experience (UX) practitioners to understand the problems that users encounter while interacting with products via their verbalized thought processes [28, 42, 50]. However, analyzing recorded think-aloud sessions is often arduous and entails scrutinizing users' verbalizations (i.e., utterances) and conducting video/audio analysis to pinpoint the problems that they encountered [35, 39]. Furthermore, UX practitioners often

The first author conducted this work when he was a PhD candidate in the Department of Computer Science at the University of Toronto.

Authors' addresses: M. Fan, School of Information, Rochester Institute of Technology, 152 Lomb Memorial Drive, Rochester, NY, 14623; email: mingming.fan@rit.edu; Y. Li, Department of Computer Science, University of Toronto; email: shurrik.li@mail.utoronto.ca; K. N. Truong, Department of Computer Science, University of Toronto, 40 St. George Street, Room 7268, Toronto, ON, Canada, M5S 2E4; email: khai@cs.toronto.edu.

2160-6455/2020/05-ART16 $15.00
https://doi.org/10.1145/3385732

work under time pressure and tend to perform quick rather than rigorous analyses. As a result, researchers have argued for developing techniques to support fast-paced analysis [24, 43].

To facilitate the analysis of think-aloud sessions, Cooke analyzed what users verbalized during think-aloud sessions and categorized their verbalizations into four categories [14]. These four categories were further confirmed and extended by other studies [21, 29, 32, 57]. Recently, Fan et al. went a step further to examine whether there are patterns in users' verbalization and speech features that are indicative of usability problems that they encounter [23]. They found that when users encounter usability problems, they tend to verbalize utterances of one particular type of verbalization (e.g., comments) more often than other types of verbalizations (e.g., action description) and also tend to verbalize utterances with negative sentiment, abnormally high or low pitch, or low speech rate, among others [23]. This finding offers direct evidence to link users' *verbalization and speech features* to usability problem encounters. We hypothesize that it might be possible to detect usability problem encounters by leveraging these subtle verbalization and speech patterns during think-aloud sessions.

Recently, natural language processing (NLP) and machine learning (ML) technologies have become increasingly powerful and are gradually adopted to tackle challenging problems in qualitative research [15]. For example, researchers have designed ML methods to automate or semi-automate qualitative coding [38, 53, 54] to detect potential disagreements in qualitative coding between coders [10] and to generate human-understandable explanations that reveal AI's internal states [20]. Inspired by this line of research, in this work, we focused on the domain of usability testing and took the first step to *design and evaluate computational methods to automatically detect usability problem encounters in think-aloud usability test sessions*. Furthermore, inspired by the finding that links users' verbalization and speech features to the encounters of usability problems [23], we also sought to *examine whether the verbalization and speech features that tend to occur when users encounter usability problems* [23] *could be used to improve the detection of usability problem encounters*.

To answer the above research question, we first conducted and recorded think-aloud sessions, labeled the encounters of the usability problems, computed the verbalization and speech features, and then trained and evaluated a wide range of ML models using different sets of features. We further examined the effect of test products and users on the ML models' performance. As the first attempt to automate the detection of usability problem encounters in think-aloud usability tests, our work calls out and sets a baseline for a set of technical and interactive intelligent user interface design challenges for forging a symbiotic relationship between UX practitioners and machine intelligence.

2 BACKGROUND AND RELATED WORK

2.1 Conducting Think-aloud Sessions

Think-aloud protocols are a widely used and highly valued usability testing method that is often used in iterative design to help ensure that products work as intended [28, 42, 50]. The protocols enable evaluators to identify usability problems encountered by potential users and gain insights into their thought processes that cannot be obtained from mere observations [14, 49].

When using think-aloud, participants verbalize their thought processes while carrying out a task. Participants' verbalizations provide data to understand their thought processes. To ensure that users' verbalizations reflect their unaltered thought processes, Ericsson and Simon proposed *three guidelines* for conducting think-aloud sessions: (1) use neutral instructions that do not request participants to verbalize any specific type of verbalization (e.g., explanation); (2) administer a practice session to help participants get familiar with verbalizing their thoughts; (3) keep

interaction with participants to a minimum (e.g., only use a neutral "keep talking" token to remind participants to think aloud if they fall into silence for a period of time) [22]. Fox et al. conducted a meta-analysis of 94 studies between 1983 and 2009 that used think-aloud protocols and found that thinking aloud had little or no effect on users' performance if these guidelines were heeded. In contrast, artificial changes in users' behavior can happen if these guidelines were breached [25]. For example, when users were explicitly asked to verbalize a specific type of content during think-aloud sessions, their behavior changed [40]; when users were interrupted by questions from the evaluators during think-aloud sessions, their behaviors were also altered [30]. Furthermore, if users did not have a chance to practice thinking aloud, they often had difficulty to verbalize their thought processes frequently [9]. As a result, we followed the three guidelines to conduct think-aloud sessions to curate our dataset for detecting encounters of usability problems in this work.

2.2 Analyzing Think-aloud Sessions

User-based evaluations, such as using think-aloud protocols, tend to facilitate the identification of a higher number of problems than inspection-based methods, such as walk-through or heuristic evaluation [34]. However, conducting and analyzing think-aloud sessions is often time-consuming and labor-intensive [35, 39]. Furthermore, in practice, usability evaluators often face time pressure [11, 43]. Consequently, they may choose to perform quick, instead of thorough and rigorous, analysis [11, 43]. For example, Nørgaard and Hornbæk found that evaluators often use test notes produced in test sessions instead of performing rigorous analysis on actual session recordings [43]. This can miss potential usability problems. As a result, they urged the community to develop techniques to support fast-paced analysis [43]. Kjeldskov et al. proposed an instant data analysis method to identify problems from think-aloud test sessions [35]. Instead of focusing on identifying as many potential problems as possible, the method emphasizes identifying the most critical problems. Furthermore, the method requires collocated collaboration and brainstorming among multiple evaluators (one as the test moderator, one as the data logger, and one as the moderator for the brainstorming session). However, in practice, few UX practitioners had an opportunity to analyze the same test session with others [24]. Motivated by this line of research, in this work, instead of optimizing the manual analysis workflow, we sought to develop computational methods to automatically detect usability problem encounters in think-aloud sessions.

2.3 Verbalizations and Speech Features and Usability Problems in Think-aloud Sessions

To better understand the verbalizations (i.e., utterances) that users verbalized during think-aloud sessions, Cooke segmented verbalizations in recorded think-aloud sessions into small segments based on pauses between verbalizations as well as their meaning and proposed a coding scheme to categorize *verbalizations* into *four categories*: Reading, Procedure, Observation, and Explanation [14]. The Reading category refers to when *the user reads words, phrases, or sentences directly from the product or its instruction manual.* The Procedure category refers to when *the user describes his/her current/future actions.* The Observation category refers to when *the user makes remarks about the product, its instruction manual, or himself or herself*; and the Explanation category refers to when *the user explains their motivation for their behavior.* The four-category scheme was later examined and confirmed by Elling et al. [21] and Hori et al. [32] on their datasets, respectively. The categories were further extended by other studies [29, 57]. While Hertzum et al. broke down the Observation category further into four fine-grained sub-categories (i.e., system observation, redesign proposal, domain knowledge, and user experience) [29], Zhao et al. broke down the Observation category into three different sub-categories (i.e., expectation, positive experience, and

negative experience) [57]. As a result, different categorization schemes proposed in the literature can be mapped to Cooke's four categories. Therefore, we adopted this scheme to categorize users' verbalizations in this work.

Recently, Fan et al. took a first step further to examine whether there are patterns in users' verbalization and speech features when they encounter problems. They found that users tend to verbalize utterances of the Observation category more often than other categories or utterances of negative *sentiment* when encountering problems; users also tend to raise *questions* (e.g., why, what, how) or use *negation*s (e.g., no, not) when encountering problems [23]. Furthermore, users also tend to verbalize in an *abnormally high or low pitch* or *abnormally low speech rate* when encountering problems. Inspired by these verbalizations and speech features patterns that are indicative of usability problems, we sought to examine whether these subtle patterns could be leveraged to improve usability problems detection.

2.4 Computational Methods for Qualitative Research

Natural language processing (NLP) and machine learning (ML) methods have become increasingly powerful. Although most of the work is driven primarily within computational fields to design more effective algorithms for classification and clustering tasks, qualitative researchers have recently begun to leverage NLP and ML methods for qualitative analysis. For example, *Coding* is an important step in qualitative analysis, in which researchers comprehend and annotate texts with descriptive labels called *codes* [16, 52]. However, coding is often arduous or even intractable for large amounts of data [5, 33]. Recently, researchers have explored methods to automate or semi-automate the coding process [38, 53, 54, 58]. Besides automating the coding process, Chen et al. recently built ML models for each coder that can automatically code qualitative data and then used these models to highlight potential disagreements/conflicts between coders so coders can better focus on resolving potential disagreements/conflicts [19]. Furthermore, researchers have also leveraged NLP and ML methods to generate explanations to express an AI agent's internal states into a natural language [10, 20]. Following the trend of using NLP and ML to tackle qualitative research problems, we focused on automating the analysis of think-aloud usability testing sessions and answering the following research questions (RQs):

- **RQ1**: Can *users' verbalizations* (i.e., *utterances)* during think-aloud sessions be used to detect usability problem encounters?
- **RQ2**: Can *the subtle verbalization and speech patterns that tend to occur when users encountered problems* [23] be used to improve usability problem detection?

Furthermore, as UX practitioners typically focus on identifying problems for a specific product to improve its user experience, it could be beneficial *to build ML models for a product that can detect usability problems encountered by a new user,* because this would allow UX practitioners to leverage the automatic detection results to speed up their analysis for a new user's test data. Similarly, it is not uncommon for a company to maintain a pool of volunteer testers whom they may contact for testing different products to save the recruitment cost. Therefore, it could also be beneficial *to build an ML model for a user* (e.g., *a volunteer in the company's volunteer testers pool) that can detect the problems that the user may encounter when using a new product,* because this would allow UX practitioners to use the automatic detection results to help them identify usability problems with a new product. As a result, we also sought to answer the following two research questions in this work:

- **RQ3**: Can an ML model be built for *a product* using its existing users' think-aloud data to detect usability problems encountered by a *new* user when using the product?

- **RQ4**: Can an ML model be built for *a user* using the think-aloud data of the products that the user has interacted with to detect usability problems encountered by the same user when using a *new* product?

3 AUTOMATIC DETECTION OF THE ENCOUNTERS OF USABILITY PROBLEMS

To answer the RQs, we first needed to curate a dataset of recorded think-aloud sessions. To do so, we conducted and recorded think-aloud sessions, in which users used different types of products (i.e., physical and digital products) in a controlled lab study. We then labeled the encounters of usability problems as ground truth. We also labeled or computed verbalization and speech features as input features for training a set of ML models. To better understand how different ML methods fared in detecting the encounters of usability problems, we implemented and evaluated a wide range of ML methods.

3.1 Think-aloud Data Collection

We recruited eight participants (four females and four males, aged 19–26), all of whom were native English speakers, to participate in the think-aloud study. We recruited participants with diverse educational backgrounds (e.g., biology, creative writing, neuroscience) to reduce the potential bias of any particular educational background. The study was conducted in a quiet usability testing room with no external noises. To ensure the recording quality of sound, we used a clip-on audio recorder and asked participants to clip it on their clothes and position it close to their mouths. During the study, participants were first given a device (i.e., an alarm clock) and a task (i.e., set the alarm at a specific time) to practice thinking aloud while working on the task. After the practice session, each participant was given two *physical devices* (i.e., a coffee machine and a universal remote control) and two *digital websites* (i.e., a national history museum website (HM) and a national science and technology museum website (STM)) in a random order to work on while thinking aloud. All participants had not used these specific products before the study. For each physical device, each participant worked on tasks related to its main functions and had access to each device's instruction manual. For each website, each participant worked on three tasks related to the websites' primary functions. The order of the four test products was randomized. The details of the tasks are listed in Table 1.

During the study, the moderator did not probe or interact with participants except reminding them to keep talking if they fell into silence longer than 15 seconds. All think-aloud sessions were audio-recorded. In total, 64 think-aloud sessions were recorded (each participant performed eight sessions: one task for each physical device and three tasks for each website). The length of the sessions ranged from 62 seconds to 1,255 seconds ($M=360$, $SD=279$). These think-aloud sessions were used as the dataset for training and evaluating ML models.

3.2 Usability Problems Labelling

The think-aloud sessions were first manually transcribed into text to ensure their accuracy. Then, two coders followed a similar approach used in previous work [14, 21] to divide each think-aloud session recording into smaller segments to facilitate further annotation. The beginning and the end of a segment was determined by pauses between verbalizations and the verbalization content [14, 21]. Each segment could include single words, but also clauses, phrases, and sentences. For each segment, two coders labeled independently whether the user experienced a problem (e.g., being frustrated, confused, or having trouble). Upon completion, they discussed to consolidate the *problem* label (0 or 1) for each verbalization segment in the dataset, *which* was used as the *ground truth*.

Table 1. The Tasks for the Test Products (Two Physical Products and Two Digital Products)

Products	Tasks
Coffee machine	Program the coffee machine to make 2 cups of strong-flavored drip coffee at 7:30 a.m.
Universal Remote Control	Program the universal remote control to control a DVD player.
STM	Your friend is an 8th-grade science teacher. She asks you to check if there are any available school programs in April at the Science museum. Your task is to find out whether any programs may be suitable for 8th-grade students in April.
STM	Your uncle has an 11-year-old child. One day, the child asks you a question, "what is it like to be a scientist or an engineer?" You have heard that the museum offers interactive presentations during which children can interact with speakers, who are scientists. Thus, your task is to find out if there is any such program in March for an 11-year-old child.
STM	You are a college student and working on an assignment about early telescopes. Your task is to obtain a photo of an instruction manual, which is for an early telescope.
HM	Your friend is a 7th-grade teacher. She is organizing a trip for 30 7th-grade students to the history museum. Your task is to help your friend find an available program in March for 30 7th-grade students.
HM	Your friend has a 4-year-old child and is planning to take him to the history museum. Please help your friend find out the number of activities that are appropriate for a 4-year-old child in March.
HM	You are a graduate student and currently researching the topic of first *peoples* in Canada. Your task is to search for an essay on the topic.

In total, there are 4,111 segments, including 483 problem segments and 3,628 non-problem segments. Here are a few excerpts from think-aloud sessions in which users used the coffee machine to show what problem segments look like: "*it's no longer flashing, hmm, okay*"; "*Oh I see, so I was actually trying to open a wrong thing*"; "*place the carafe [user was confused about the word 'carafe' and mispronounced it with a rising tone] on the warming plate with the lid on*"; and "*it's showing the same screen again.*" Note that while the first two examples might be associated with negative sentiments based on the words used, the last two were not clearly associated with any negative sentiments. In contrast, in the third example, the user stuttered on the word "carafe" and raised her tone toward the end of the word as if she were asking a question. This indicates that she was confused about which component of the coffee machine the word referred to. In the last example, the user slowed her speech down dramatically while verbalizing her thoughts and therefore there were long pauses between words.

However, a verbalization segment might not be a problem segment even if it contains negative words. For example, here is a non-problem segment: "*Um length of warming time. Um, I wasn't really…Do I need a length of warming time? no. okay.*" The user was reading, thinking, and verbalizing and quickly figured out that he did not need to set the warming time for the task. Note that the punctuation marks were added to help with comprehension. In sum, inferring whether a user encounters a problem needs to take many factors into consideration; keyword matching or sentiment analysis alone would be insufficient.

3.3 Basic Transcript-based Feature Extraction

We computed basic text features from the transcript of each verbalization segment in our dataset. Specifically, for the transcript of each segment, we computed the *TF-IDF* (i.e., term frequency-inverse document frequency) feature vector using the Scikit-learn [45] and computed the *word embedding* vector using Tensorflow [1]. These vector representations and the ground-truth labels of the usability problems were used together to train the ML models later.

3.4 Verbalization and Speech Features Extraction

Recent research has shown that users tend to verbalize the content of the *Observation category, negations, questions*, and *negative sentiment* using *abnormal pitches* or *speech rates* when they encounter usability problems in think-aloud sessions [23]. Inspired by this finding, we aimed to evaluate whether *these verbalization and speech features* can be used to train ML models to better detect usability problem encounters. Next, we describe how we manually labeled or automatically computed the features as follows.

***Verbalization Category*:** For each segment in the recorded think-aloud sessions, two coders independently labeled it with one of the four verbalization categories (i.e., *Reading, Procedure, Observation, and Explanation*) [14]. Upon completion, they discussed their labels to resolve any conflicts. In the end, each segment was assigned a label to indicate its verbalization category.

To help readers understand the categories, we provide example excerpts of each category from the think-aloud sessions in which participants used the coffee machine: "*It says 'you can program the appliance to prepare drip coffee automatically'" (Reading); "I'm just going to put two spoons of coffee" (Procedure); "I assume strong coffee just has a lot of scoops of coffee in it" (Observation); and "Oh it flashed for a second, so I guess I'm supposed to hold it" (Explanation).*

***Negations*:** We designed a keyword-matching algorithm to determine whether users verbalized a negation. The keywords were chosen based on a recent study [23] and contained the following words: *no, not, don't, doesn't, didn't*, and *never*. Thus, each segment was assigned a binary label to indicate whether the user used a negation.

***Questions*:** We also designed another keyword-matching algorithm to determine whether users asked a question in each segment. The question keywords were chosen based on a recent study [23] and contained the following words: *what, which, why, how*, and *where*. When transcribing the think-aloud sessions, we did not consistently add or verify the accuracy of punctuation marks and their positions. Unlike verbalizations (i.e., utterances), which were deterministic, punctuation marks and their positions in verbalizations were dependent on contextual information and interpretations of the transcribers. Consequently, the keyword-matching did not use punctuation marks or require the question keywords to be the first word of a sentence, which would depend on accurate interpretation of punctuation marks and their positions. After this process, each segment was assigned a binary label to indicate whether the user asked a question.

***Sentiment*:** For each segment in the recorded think-aloud sessions, two coders independently assigned it with one of the three sentiment values (1 for positive sentiment; 0 for neutral sentiment; and -1 for negative sentiment) by referring to the text transcription of the segment and listening to the corresponding audio segment if deemed necessary by the coders. Afterward, they discussed their labels to consolidate the sentiment labels for all segments.

***Pitch*:** For the corresponding audio of each segment, we computed the participant's fundamental frequency F0 (Hz) at the sampling rate of 100 Hz using *praatUtil.calculateF0* function in the *praatUtil* library [59], which interfaces with speech process toolkit *Praat 6.0.13* [60] and uses Praat's *Sound to Pitch* function [61]. We set the frequency range to be 50–400 Hz to cover typical male and female frequencies.

***Speech Rate*:** For the corresponding audio of each segment, we computed the speech rate by dividing the number of words spoken in a segment by its duration. The number of words spoken in a segment was counted based on the text transcription of the segment.

***Abnormal Pitch and Speech Rate*:** To determine whether a segment contains *abnormal* pitch or speech rate, we computed the mean and the standard deviation of the pitch and the speech rate over the entire session recording and automatically labeled a segment as having *abnormally* high or low pitch or speech rate if any value in the segment was two standard deviations higher or lower than the mean value. As a result, each segment would have two labels to indicate whether it has an *abnormally high pitch* or *abnormally low pitch,* respectively, and one label to indicate whether it has an *abnormally low speech rate.*

3.5 ML Models

Recent research has shown the promise of ML in solving qualitative research problems. For example, Support Vector Machines (SVM) have shown to be effective in helping qualitative researchers code qualitative data [53, 54, 58], and Random Forests (RF) have demonstrated to be effective in detecting segments of question-answer from classroom conversations [4] or classifying activities in classroom discourse [51]. Therefore, we employed these two methods (i.e., SVM and RF) to detect the usability problem encounters in this work. Additionally, the convolutional neural network (CNN) and recurrent neural network (RNN) have shown to be promising on generic text classification tasks [12, 37]. Thus, we also included CNN and RNN to understand how they fare in detecting usability problem encounters on our dataset.

Specifically, we computed and used the TF-IDF vectors and the ground-truth labels of all the segments in our dataset to train the RF and the SVM models and used the word-embedding vectors and the ground-truth labels of all verbalization segments to train the CNN and the RNN models. We referred these models trained on these generic text features (i.e., TF-IDF vectors, word-embedding vectors) as the *baseline.*

To evaluate whether the six verbalization and speech features (i.e., category, sentiment, negation, question, pitch, and speech rate) can be used to improve the performance of these ML models, we appended these six features to the end of the TF-IDF vector or the word-embedding vector of each segment to construct the updated feature vectors. We then used these updated feature vectors and the ground-truth labels of all segments to train the same set of ML models. By comparing the performance of these updated models to the baseline, we were able to assess whether these verbalization and speech features helped to improve the models' performance.

We used the Scikit-learn library to implement the RF and SVM models and Tensorflow to implement the CNN and RNN models. We used *RandomForestClassifier* in Scikit-learn with default parameters for RF and *LinearSVC* in Scilit-learn with default parameters for SVM. We also used default parameters in Tensorflow for the CNN and RNN models. Our CNN model had an embedding layer followed by a convolution layer, a ReLU activation function, a max pooling layer, and then a softmax layer. Our RNN model had an embedding layer followed by a dynamic RNN with GRU cell, and a softmax layer. For both CNN and RNN models, we followed common practice and examples provided in prior work [1, 26] and initialized the embedding variable with random values of uniform distribution in the range of [-1,1) using Tensorflow (tf) function *tf.random_uniform* [62].

4 EVALUATION AND RESULTS

We performed cross-validation on the whole dataset (Section 4.1) to answer RQ1 and RQ2. We then performed leave-one-user-out evaluations for each product (Section 4.2) to answer RQ3 and leave-one-product-out evaluations for each user (Section 4.3) to answer RQ4.

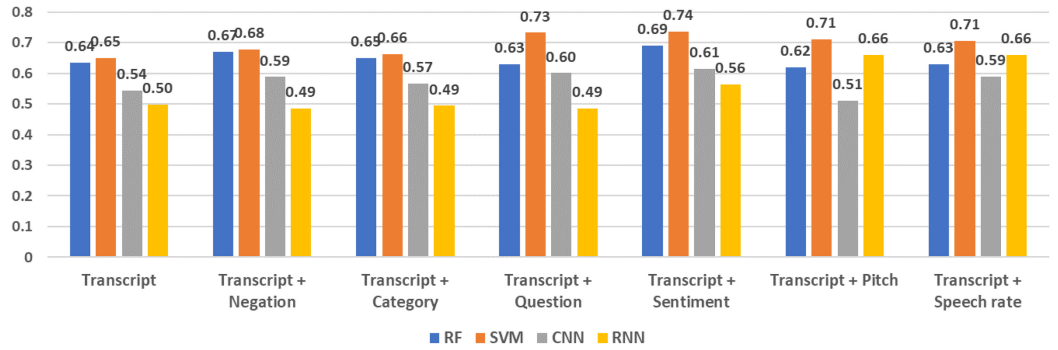

Fig. 1. The average F1-score of the ML models trained using the transcript feature only as the input or using the transcript and one additional verbalization & speech feature (i.e., negation, category, question, sentiment, pitch, and speech rate) together as the input and evaluated using 10-fold cross-validation on the entire dataset.

4.1 The Effect of *Verbalization and Speech Features* and *ML models* on Usability Problem Encounters Detection

Evaluations in this section aimed to answer RQ1 and RQ2. We trained ML models using TF-IDF or word-embedding vector extracted from the transcript (i.e., users' verbalizations) as the input feature, which was referred to as *the transcript feature*. Furthermore, we trained the same set of ML models using both the *transcript features* and one of the *verbalization & speech features* (Section 3.5) as the input. We performed 10-fold cross-validation to evaluate the models and used F1-score as the overall performance measure on the entire dataset. Figure 1 shows the result.

The average F1-score of the four ML models trained on the *transcript feature* was *.58 (SD=.07)*. In contrast, the average F1-score of the four ML models trained with the *transcript feature* and *one additional verbalization and speech feature* was *.62 (SD=.02)*. The increased performance indicates that the verbalization and speech features helped to improve the performance. The SVM models performed the best among all with the average F1-score of *.70*.

As each verbalization and speech feature was shown to improve the performance of the ML models (Figure 1), we tested whether using all the *verbalization and speech features* together as input would improve the performance of the models even further. We trained the ML models using *all the verbalization and speech features* and *the transcript feature* (i.e., *TF-IDF or word embedding vector)* together as the input and performed a 10-fold cross-validation on the entire dataset again. Figure 2 shows the precision, recall, and F1-score of the ML models trained on *the transcript feature* only and also the *transcript feature and all the verbalization and speech features* together, respectively.

The average F1-score of the four ML models was *.67 (SD=.06)*, which was higher than that of the models trained with *the transcript only* (.58) or with *the transcript feature and any one of the verbalization & speech features* together (.62). This finding suggests that the verbalization and speech features *complement* each other as the input feature for training ML models. Moreover, the absolute difference between the precision and recall for RF, SVM, CNN, and RNN models when using the *transcript feature* (i.e., *TF-IDF) and all the verbalization & speech features together* as the input feature was .16, .06, .31, and .22, respectively. This result suggests that the SVM model had the most balanced precision and recall compared to the other three models (i.e., RF, CNN, and RNN).

We used the *transcript feature* (i.e., TF-IDF or word-embedding) as the *entire* or *part* of the input to train ML models so far. To further understand whether *the verbalization and speech features alone* are sufficient to train effective ML models, we used only *the six verbalization and speech features*

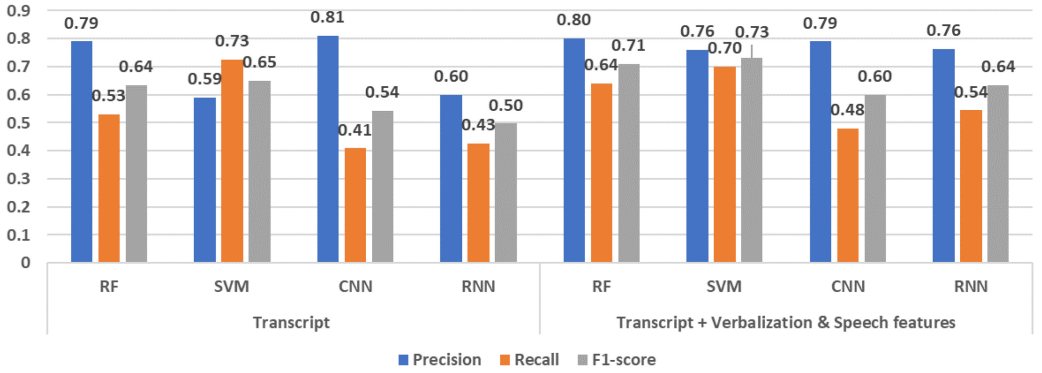

Fig. 2. The precision, recall, and F1-score of the ML models trained using the *transcript* feature only as the input (the left half) and using the *transcript + all the verbalization & speech features* together as the input (the right half) and evaluated using 10-fold cross-validation on the entire dataset.

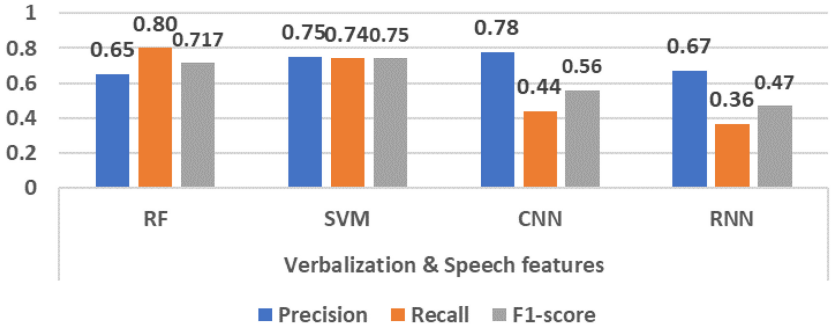

Fig. 3. The average precision, recall, and F1-score of the ML models trained using only the *verbalization & speech features* as the input and evaluated using 10-fold cross-validation on the entire dataset.

as the input vector to train the ML models and performed 10-fold cross-validation on the entire dataset. Figure 3 shows the result, which suggests that the performance of the ML models trained on *only the verbalization and speech features* was comparable to that of the same models trained on both the *transcript feature* and the *verbalization and speech features* together (Figure 2, right).

To further understand how each verbalization or speech feature contributed to the precision, recall, and F1-score of an ML model, we used each verbalization or speech feature as the input feature, respectively, to train SVM models and performed 10-fold cross-validation on the entire dataset. We used SVM for this evaluation, because it was the best-performed ML model based on the evaluations so far. Figure 4 shows the results, which compare the performance of these SVM models trained on *each verbalization or speech feature, respectively,* with that of the SVM model trained with *all the verbalization and speech features together*. The differences in precision and recall values demonstrate that each verbalization or speech feature had different precision and recall trade-off for locating usability problems. For example, the *sentiment* and the *negation* features had a relatively higher precision, while the *category, pitch*, and *speech rate* features had a relatively higher recall. Furthermore, the overall performance (i.e., F1-score) of the SVM model performed the best when trained on *all* the verbalization and speech features together. This suggests that the verbalization and speech features are *complementary* to each other for detecting usability problem encounters.

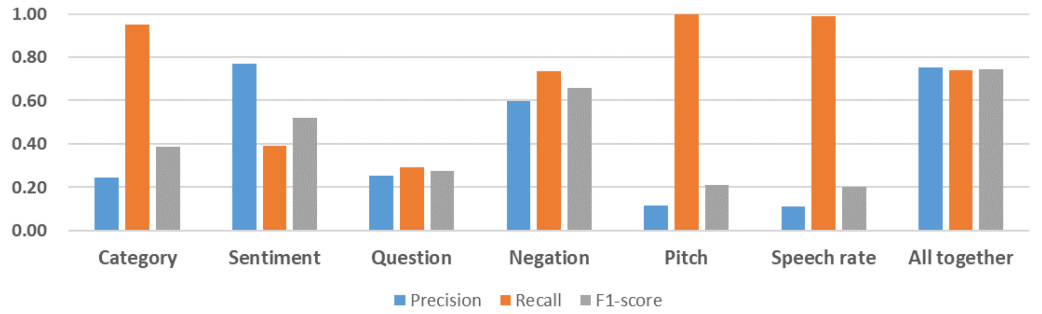

Fig. 4. The precision, recall, and F1-score of the SVM models trained with each verbalization or speech feature, *respectively*, and *together* and evaluated using the 10-fold cross-validation on the entire dataset.

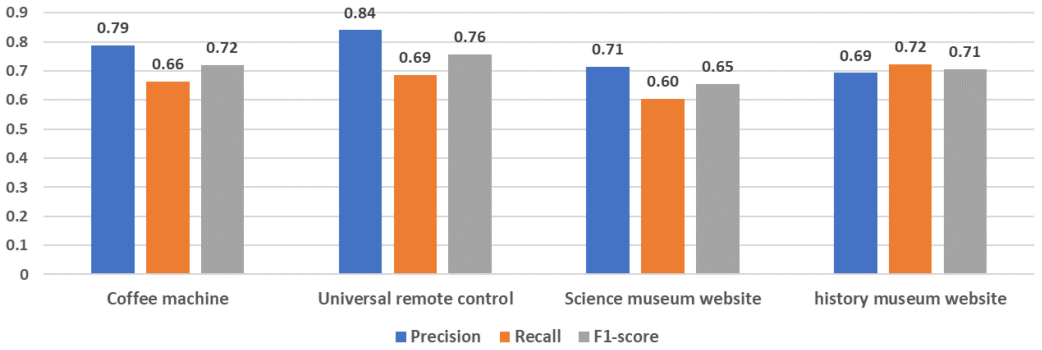

Fig. 5. The average precision, recall, and F-1 score of the SVM model trained on any seven users' data using the *transcript* (i.e., *TF-IDF) + all the verbalization & speech features together* as the input and evaluated on the rest one user's data for each *product,* respectively (i.e., *leave-one-user-out* scheme).

4.2 The Effect of *Products* on Usability Problem Encounters Detection

To reduce the workload of analyzing large amounts of users' think-aloud usability test sessions for *a product*, it is valuable to explore whether an ML model trained on existing users' test sessions can predict the usability problems of new users' test sessions.

To answer this question, we adopted the *leave-one-user-out* scheme to train and evaluate an SVM model for each *product*. We used SVM for the evaluation, because it performed best among all the models in terms of F1-score and had balanced precision and recall values. For each of the four products, we trained an SVM model using *the transcript* (i.e., *TF-IDF) + verbalization & speech features together* as input on any seven users' data and then tested the model using the rest one user's data. The rest one user was used to simulate the new user whose data the SVM model did not use in the training. As each product was used by eight users in our dataset, we repeated this evaluation process eight times so each user was treated as the new user once for each product. Finally, we averaged the three measures (i.e., precision, recall, F1-score) of the SVM models across all eight users for each product.

Figure 5 shows the average precision, recall, and F1-score of the SVM models for each *product* when using *the transcript feature* (i.e., *TF-IDF) and the verbalization and speech features together* as the input. The result shows that it is possible to detect the usability problems encountered by a new user for a product with reasonable precision, recall, and F1-score. In addition, the average F1-score of the SVM models for two physical devices and two digital websites were .74 and .68, respectively,

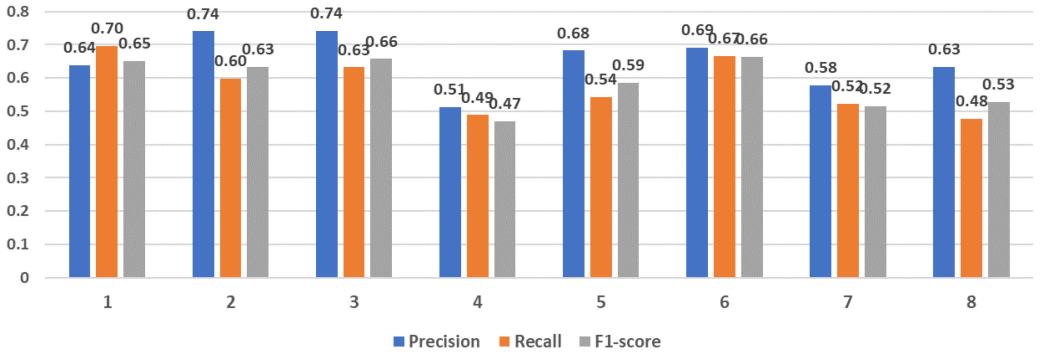

Fig. 6. The average precision, recall, and F-1 score of the SVM model trained on any three products' data using *the transcript* (i.e., *TF-IDF) + all verbalization & speech features together* as input and evaluated on the rest one product's data for each *user,* respectively (i.e., *leave-one-product-out* scheme).

which indicates that the models performed relatively better for the two physical devices than for the two digital websites.

4.3 The Effect of *Users* on Usability Problem Encounters Detection

It is not uncommon for companies to maintain a pool of participants whom they could contact over time for usability testing to reduce the recruitment cost. Thus, it is possible for companies to accumulate a dataset of the *same* user interacting with *various* products. If an ML model could be built to predict when *a user* would encounter problems while interacting with *a new product* using the thinking-aloud data of this user when she interacted with *existing* products with reasonable accuracy, then this ML model could potentially help UX evaluators speed up their analysis of think-aloud sessions for the new product. With a dataset of a user interacting with four different products, we were curious and able to explore whether it is possible to build an ML model for *a user* using the data of the products that the user has already interacted with to detect potential usability problems that the same user might encounter when she uses *a new product.*

To answer this question, we adopted the *leave-one-product-out* scheme to train and evaluate the ML model for each user, respectively. We used SVM again, because it performed the best among all ML models in terms of F1-score and had balanced precision and recall values. For each user in our dataset, we trained an SVM model for the user using *the transcript* (i.e., *TF-IDF) + verbalization & speech features together* as input on any three of the four products' data and tested the model on the rest one product's data. The rest one product was used to simulate the new product that the user used but the ML model did not use for training. As each user used four products in our dataset, we repeated this process four times so each product was used as the new product once for each user. Finally, we averaged the measures across all the products for each user. Figure 6 shows the result. The F1-scores of the SVM models for the eight users ranged between .47 and .66. The performance was relatively better for some users (e.g., users 6, 3, 1, and 2) than others (e.g., users 4 and 7).

4.4 The Effect of *Product & User Combinations* on Usability Problem Encounters Detection

It is also valuable to understand how well an ML model trained on an existing set of products and users would perform on the data of a *new* user when she uses a *new* product. In other words, the

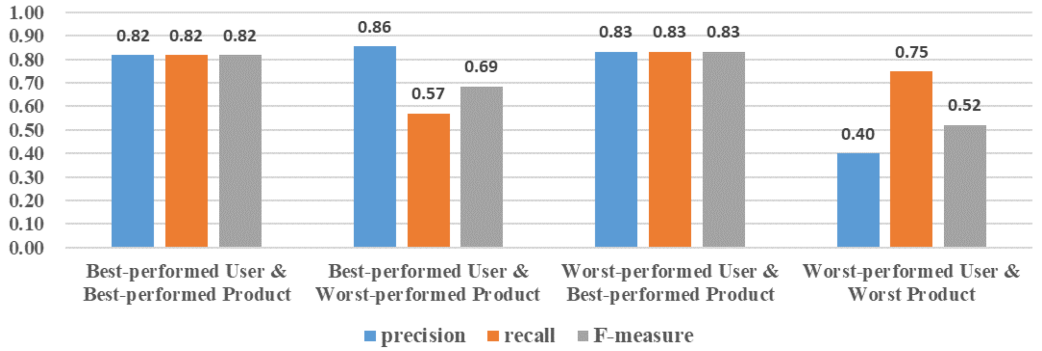

Fig. 7. The precision, recall, and F1-score of the SVM model trained on seven users and three products' data using the *transcript* (i.e., *TF-IDF) + all the verbalization & speech features together* as the input and evaluated on the combination of the remaining one user and the remaining one product's data (i.e., *leave-one-user-one-product-out scheme*).

research question (RQ5) is: *Can a pre-trained ML model using an existing dataset be used to predict usability problems that would be encountered by a new user when she uses a new product?*

To answer this question, we adopted the *leave-one-user-and-one-product-out* scheme to train and evaluate the ML model. Specifically, we trained SVM models using the dataset of seven users and three products and used the remaining one user's remaining one product data for testing. There are 32 (4 * 8) combinations of products and users. As many of these combinations have similar setups and would likely result in similar performance, we decided to evaluate only four potential boundary conditions: the best and worst performed products as determined in Section 4.2 (i.e., universal remote control and science museum website) combined with the best and worst performed users as determined in Section 4.3 (i.e., user 6 and user 4). Specifically, for each of the four user and product combinations, we used the data of the remaining users and products to train the models and test on the data of the combination. Figure 7 shows the result.

5 DISCUSSION

5.1 The Effect of the *Verbalization and Speech Features* on the Detection of Usability Problem Encounters

The results in Figure 1 and Figure 2 confirm that the *verbalization and speech features* that were found to be indicative of usability problems in Fan et al.'s studies [23] can be used to improve the detection of usability problem encounters when used with the transcript features (i.e., TF-IDF or word embedding). The potential reason might be that users tend to verbalize their thoughts in similar ways when they encounter problems in think-aloud sessions, and these similarities can be reasonably captured by these verbalization and speech features.

Furthermore, the result in Figure 3 demonstrates that the *verbalization and speech features* can be used to train effective ML models to detect usability problem encounters without needing to be used with the generic transcript features (i.e., TF-IDF or word-embedding). It implies that the verbalization and speech features that were found in Fan et al.'s studies [23] are informative and comprehensive to capture the key characteristics of usability problem encounters in think-aloud sessions.

In addition, the performance of SVM models trained with each verbalization or speech feature, respectively, (Figure 4) suggests that different verbalization and speech features helped differently in terms of improving precision and recall when detecting usability problem encounters. This

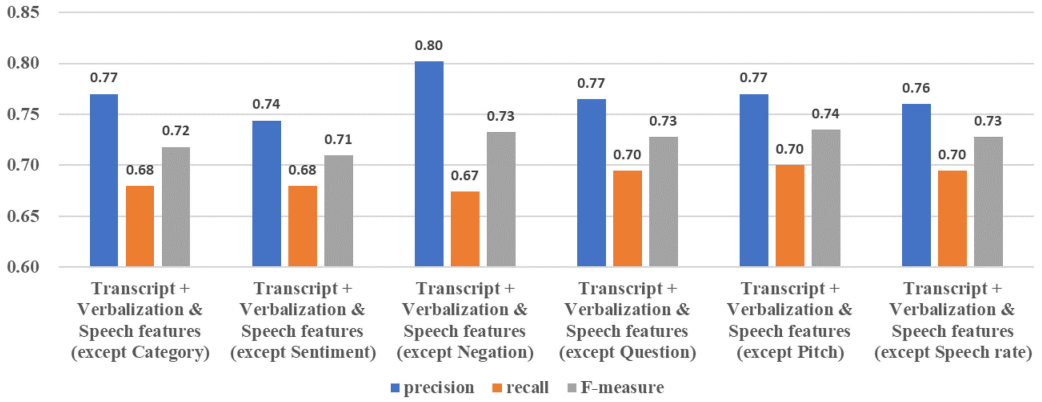

Fig. 8. The precision, recall, and F1-score of the SVM models trained using all the transcript + verbalization & speech features *except one* as the input and evaluated using 10-fold cross-validation on the entire dataset.

result meets our expectation, because users tend to show a rich set of honest signals ranging from using negative words, raising their tones, to slowing down their speech rates when encountering problems, as the example problem segments shown in Section 3.2. Such a rich set of signals could be better captured by different verbalization and speech features together than any individual one. This is evident by the fact that the SVM model performs the best when trained with all the verbalization and speech features together than trained with any individual verbalization or speech feature (Figure 4).

Although verbalization and speech features are most effective when they are used together to train an ML model, getting some of these features, such as Category, required manual annotation at the current stage. To better understand whether some manual annotation effort can be saved, it is worth exploring whether removing one feature while training an ML model would affect its performance. For each verbalization or speech feature, we trained an SVM model with all transcript and verbalization and speech features *except that feature* and performed a 10-fold cross-validation on the entire dataset. The result (Figure 8) shows that removing one feature from the transcript + all verbalization and speech features does not seem to affect the performance much. This is encouraging, because it suggests that it might be possible to save effort by not labeling one feature if all other features are available when building an ML model to detect usability encounters. However, further research is needed to examine whether this finding still holds with a more diverse set of products and a larger group of think-aloud participants.

5.2 Trade-offs between Precision and Recall of the Verbalization and Speech Features

We have used F1-score to compare the performance of ML models trained with different features, because the F1-score is often used to estimate the overall performance of a classifier. However, the results in Figure 2, Figure 3, and Figure 4 suggest that different ML models and even the same ML model trained with different features perform differently in terms of precision, recall, and F1-score. No single method performs the best for all measures. The implication for future automatic detection of usability problems is that instead of hoping to build a single ML model that performs the best in all measures (i.e., precision, recall, and F1-score), it might be more practical to develop ensemble methods (e.g., References [7, 17]), which can combine different ML models that have an edge in different measures together to achieve better performance.

Moreover, Precision and Recall emphasize different aspects of the performance of an ML model. A recent study that aimed to understand whether users would accept an imperfect Artificial

Intelligence (AI) [36] suggests that users indeed value precision and recall of an AI agent differently. Consequently, the precision and recall measures should be weighted differently when determining which verbalization and speech features to use for training the desired ML model. For example, if usability evaluators wish to train an ML model that captures as many potential problems as possible for them to review and have a higher tolerance for false positives, then the recall measure should be weighted more than the precision measure. As a result, the verbalization and speech features that could lead to high recall should be preferred. In our case, Figure 4 shows that the pitch, speech rate, and category features should be prioritized over other features. In contrast, if usability evaluators wish to train an ML model that captures potential problems as accurately as possible and can care less about missing a potential problem, then the precision measure should be weighted more than the recall measure. In our case, Figure 4 shows that the sentiment and negation features should be prioritized over other features.

This recommendation, however, would require UX practitioners to make trade-offs between Precision and Recall when choosing an ML model to assist them with identifying usability problems. However, this can be challenging, because UX practitioners are often not ML experts and lack the knowledge needed to make an informed decision. Therefore, it remains unknown how to assist UX practitioners to choose an ML model that balances between Precision and Recall. Perhaps, one potential approach is to design an interactive user interface that allows UX practitioners to tune their desired Precision and Recall values/ranges, for example, by dragging sliders, and provides visual feedback of the potential problems that the underlying ML detects.

5.3 The Effect of Different Types of *ML Models* on the Detection of Usability Problem Encounters

Given the overwhelming evidence of the advantage of deep neural networks (DNNs) over the traditional ML methods, one might expect that the CNN and RNN models would outperform the SVM and RF models. The results in Figure 1 and Figure 2, however, show that SVM models worked better than the other three models, including the two deep learning models, in terms of the F1-score. This could potentially be because the DNNs had more parameters to optimize than the two simple models (i.e., SVM, RF) but our dataset was relatively small and insufficient for the DNNs to learn their optimal parameters.

To better leverage the power of DNNs, one potential approach is to develop methods to effectively curate a larger dataset that would allow the DNNs to learn their optimal parameters. However, curating large datasets in the usability testing domain is challenging in practice, because scheduling and conducting usability studies (e.g., think-aloud sessions) with participants in a controlled lab environment is often labor-intensive and time-consuming. However, one potential opportunity to gain large amounts of usability testing sessions is through remote usability testing, in which users can participate remotely in their convenient environment without the burden of scheduling and coming to the lab. Remote usability testing is promising also because it has shown to be cost-benefit effective (e.g., References [6, 8]). For example, Andreasen et al. showed that remote synchronous usability testing is virtually equivalent to the conventional lab-based controlled user studies [3]. However, one limitation with remote usability testing is that it would be challenging if users need to access and interact with a physical product. Another challenge with curating a large dataset lies in *transcribing* and *annotating* the test sessions and consolidating the *ground-truth labels* for usability problems. In this work, researchers took the burden of completing these steps. However, future research should develop better tools and methods that either facilitate UX professionals to label the dataset efficiently or automate or semi-automate the labeling process, such as via crowdsourcing.

5.4 The Effect of *Products* on the Detection of Usability Problem Encounters

Our leave-one-user-out evaluations for each product (Section 4.2) demonstrate that it is possible to build an ML model for a product to detect problems that a *new user* encountered when using the product. The implication is that companies could utilize recorded think-aloud test sessions that they have collected so far for a product to train an ML model to process the think-aloud sessions of a new user to pinpoint where in the sessions the new user encounters problems.

The leave-one-user-out evaluation results also show that the performance of the models was relatively better for the physical products (with the average F1-score of .74) than for the digital websites (with the average F1-score of .68). One potential reason for the difference might be the natural difference in the utterance of the users when using physical and digital products. Another possibility might be related to *the type of tasks* that users worked during the tests. For the physical products, users worked on *guided tasks,* because they had access to the instruction manuals, which offer a prescribed set of steps to complete the tasks. In contrast, for digital websites, users worked on *unguided tasks,* because they had no access to any prescribed steps. Although users can deviate from the prescribed steps when working on *guided tasks*, the availability of the guided steps could have influenced users' usage patterns and caused them to verbalize more similar utterances than when they were working on the *unguided tasks* with digital products, for which they must freely explore to complete the *unguided tasks.* However, further research should examine whether and how *the type of tasks* that users work on during think-aloud sessions influence their verbalizations and its implications on the design of effective ML models for detecting usability problems.

5.5 The Effect of *Users* Models on the Detection of Usability Problem Encounters

Our leave-one-product-out evaluation for each user (Section 4.3) demonstrates that it is possible to build an ML model for each user using the data of the user interacting with existing products to detect problems that the user might encounter when using a *new product.* However, the result also shows a large variation in models' performance for different users. One potential reason for the variation in users' performance could be that different users may have verbalized the usability problems they encountered to different extents. Some users' verbalizations reflected the problems that they encountered more *explicitly in their utterances* than other users. Another potential reason for the variation in the performance could be that some users may have verbalized their thought processes *more consistently* across products than other users. This consistency in their verbalizations may have helped the ML model learn and generalize. However, further research is needed to fully understand what affects the performance of such user-dependent models.

5.6 Integrating Machine Intelligence into UX Practitioners' Workflow

Although some recent research has shown that UX practitioners often struggle to understand the capabilities and limitations of ML [18, 56], some also suggest UX practitioners can design ML-enhanced products without knowing about ML thoroughly [55]. This is encouraging, because it suggests that it is possible to integrate machine intelligence, such as the automatically detected encounters of usability problems as described in this work, into UX practitioners' workflow to improve their analysis efficiency. Toward this goal, we highlight two possible directions to forge a successful symbiosis relationship between UX practitioners and machine intelligence.

First, it is beneficial for UX practitioners to have a second perspective on their analysis to reduce the potential "evaluator effect" [31]. Unfortunately, in practice, fewer than 30% of the UX practitioners had a chance to work with another practitioner to analyze the same usability test session due to many practical constraints [24]. As this work shows that ML models can achieve reasonable accuracy in detecting usability problem encounters, it might be possible to use the ML models as

Table 2. The Accuracy, Precision, Recall, and F1-score of the Two-class (i.e., Observation and Non-Observation) and Four-class (i.e., Reading, Procedure, Observation, and Explanation) Category Classifiers

	Accuracy	Precision	Recall	F1-score
Two-class classifier	.83	.86	.71	.78
Four-class classifier	.75	.78	.64	.68

"*virtual evaluators*" to provide a second perspective to UX practitioners so they could spot problems that they would have missed otherwise. To fulfill this vision, future research must overcome many intelligent interactive user interface design challenges, such as *how should the user interface be designed so UX practitioners could trust the "virtual evaluator" and better leverage its detection results into their analysis*?

Second, previous research has argued and demonstrated that machine intelligence can benefit from users' input over time [2, 27, 41]. Similarly, we hypothesize that ML models could detect the encounters of usability problems more accurately with input from UX practitioners, such as correcting the ML's detection results that are deemed to be wrong. However, such mixed-initiative human-in-the-loop designs must be carefully considered. For example, if ML models keep getting corrected by a UX practitioner, these models might be biased by this user's input and behave more and more like the UX practitioner, which may not be able to help the UX practitioner overcome her confirmation bias anymore. Further research should examine *how to design an interactive interface that allows UX practitioners to not only leverage the ML's detected encounters of usability problems but also conveniently provide their feedback to the ML without biasing it in the long run.*

5.7 Automatic Verbalization Category Labelling

We aimed to understand the effect of users' verbalization and speech features in think-aloud sessions on the detection of usability problem encounters. As a result, we chose to label the verbalization category for each segment in the dataset *manually* to ensure the label's accuracy. Since our evaluations show that the verbalization category is useful in improving the performance of the ML models, we took a step further to answer the following research question (RQ6): *Can the verbalization category for a segment be determined automatically based on the verbalized text content* (i.e., *the words that users uttered)?*

Informed by the findings of the recent research that the *Observation* category is most indicative of the usability problems among all the four categories [23], we sought to build a binary classifier to *detect whether a segment should be labeled as the Observation category or the non-Observation category*. To answer this question, we went through the category labels for all the verbalization segments and grouped the *Reading*, *Procedure*, and *Explanation* categories into the *non-Observation* category but kept the *Observation* label unchanged. We then followed the same procedure as described in Section 3.4 to compute the TF-IDF feature for each segment as the input feature and used the binary verbalization category labels as the ground truth to train an SVM classifier to classify whether a segment should be labeled as *Observation* or *Non-Observation*.

We performed 10-fold cross-validation on the SVM classifier using the entire dataset. The accuracy, precision, recall, and F1-score of the binary classifier were .83, .86, .71, and .78, respectively (i.e., the first row in Table 2). The result implies that it is possible to reduce the effort of manually labeling the verbalization category, especially for large amounts of think-aloud sessions, by building a binary-class category classifier. Of course, to create such a classifier, UX practitioners still need to label a small portion of their data to curate the training data.

Table 3. The F1-score of the Four ML Models for Detecting Usability Problem Encounters When Trained with *Transcript + Automatically Generated Category Labels* (Bottom Row) and Trained with *Transcript + Manually Generated Category Labels* (Upper Row)

	RF	SVM	CNN	RNN
Transcript + manual category labels	.65	.66	.57	.49
Transcript + automatic category labels	.70	.65	.43	.55

Although the binary verbalization classifier might be enough for usability problem encounter detection, the other three verbalization categories (i.e., Reading, Procedure, Explanation) could be useful in terms of providing contextual information to understand the issues that may ultimately be verbalized in an Observation segment [23]. Thus, it is also valuable to distinguish the four verbalization categories (i.e., Reading, Procedure, Observation, and Explanation).

We further trained an SVM classifier to detect the four verbalization categories and performed a 10-fold cross-validation on the entire dataset. The average accuracy, precision, recall, and F1-score of the four-class classifier were .75, .78, .64, and .68, respectively (Table 2). Although the measures for four-category classification, as expected, are lower than those of the binary verbalization category classifier, the measures are nevertheless still promising. Future work may explore more effective methods to improve the performance of the verbalization category detection, for example, by creating more effective input features or ML models.

One natural follow-up research question (RQ7) is: *How well could the automatically detected category labels be used to detect usability problem encounters compared to the manually labeled category labels?* To answer this question, we used the transcript and the automatically generated category labels (with four categories) as input features to train the four types of ML models to detect usability problem encounters and performed 10-fold cross-validations on the entire dataset. Table 3 shows the result. For easy comparison, Table 3 also includes the corresponding result (i.e., transcript + manually labeled category labels) taken from Figure 1. The result suggests that the performance of the ML models was not always worse or better when using automatic category labels than manual category labels. Specifically, the performance of the ML models using automatic category labels was better than manual category labels in some models (e.g., RF, RNN) but worse in others (e.g., SVM, CNN). This result implies that it is possible to further automate the entire process of usability problem detection, such as automating the verbalization category labeling.

5.8 Summary of the Key Findings

Our evaluations have identified the following key findings: First, ML models trained on the generic *transcript feature* (i.e., TF-IDF or word embedding) can detect usability problem encounters. Second, the *verbalization and speech features* that were found to be indicative of usability problems in Fan et al.'s research [23] were able to improve the ML models' performance. Furthermore, the ML models achieved the best performance when using *all* the verbalization and speech features together. Third, compared to the other three ML methods, SVM performed the best in terms of F1-score and had the most balanced precision and recall values. However, the other three ML methods outperformed SVM in precision or recall measure separately. Fourth, ML models trained on the existing users' data for a product can detect the encounters of usability problems by a *new* user when the user uses the product. Fifth, ML models trained on the data of the products that a user has interacted with can detect the encounters of usability problems when the user uses a *new* product. Last, ML models can be built to label the verbalization Category feature with reasonable accuracy.

5.9 Limitations and Future Work

As a first step toward automating the detection of usability problem encounters in think-aloud usability testing, our work sets a *baseline.* Future work could explore more advanced computational methods to improve the detection accuracy. Besides, we have identified the following directions for future work.

First, our work focused on detecting the encounters of usability problems, the period when users encountered problems. The ML models we built and evaluated in this work, however, are not able to provide the details about the problems. For example, the ML models do not know what problems users encountered, the causes of the problems, the severity of the problems, and the potential design solutions to the problems. Therefore, the ML models in this work could act as an assistant to UX evaluators to locate the problems effectively but would still rely on UX evaluators to interpret the problems. Future work should explore ways to help UX evaluators assess the aforementioned details about the automatically detected usability problems.

Second, our dataset contained 64 think-aloud sessions in which eight participants used four products. Although the dataset included multiple users and multiple products, it was still relatively small. This could be a reason why the data-intensive models, such as CNN and RNN, did not outperform shallow-learning methods, such as SVM and RF. Future work should curate a larger think-aloud dataset, which includes a larger number of participants and a more diverse set of products, to reassess the performance of ML models and understand whether deep neural networks can achieve better performance. However, conducting think-aloud sessions in a controlled lab environment is labor-intensive and time-consuming. One potential solution is to conduct remote usability testing, which does not require users to be physically present in a lab and therefore allows for recruiting a more diverse set of participants around the world.

Third, our evaluations show that it is possible to build an ML model for *a specific product* using its existing users' data to detect the problems encountered by *a new user*. The performance of such models, however, still has room to improve. Future work should examine how to build more effective ML models to detect usability problem encounters.

Fourth, our evaluations suggest that the *types of tasks* that users perform in think-aloud sessions might have influenced the models' performance. For example, the *guided tasks*, which were used for physical devices and provided instruction steps for users, might have resulted in higher levels of similarity in users' verbalizations than the *unguided tasks*, which were used for digital websites and provided no instructions about how to complete the tasks. Future work should further examine whether *the type of tasks* indeed affects the detection of usability problem encounters.

Fifth, our evaluations also suggest that although ML models can be built for each user to determine the potential problems that the user might encounter when using a new product, the performance of these user-dependent models varied across users. Future work should examine what causes this difference and design methods that can work better for each user.

Sixth, different languages have different pronunciations and grammars to organize and communicate thoughts and are influenced by different cultures. For example, a field study of think-aloud testing in seven companies in three different countries (i.e., Denmark, China, and India) suggested that the way usability problems are experienced by test participants can be different [13]. Similarly, Shi conducted a field study with companies located in the industrial areas in China and found that Chinese participants tend to have difficulty verbalizing their higher levels of thinking, which might be due to the Chinese holistic thinking style [47]. Thus, if these subtle patterns were to be used to build ML models to detect usability problems for other languages, it is necessary to *examine whether the subtle verbalization and speech patterns that tend to occur when users encounter problems are affected by the languages and the cultures in which the think-aloud participants live.*

Seventh, all our participants were young adults. Recent research has suggested that age might have an influence on think-aloud usability testing in terms of task performance and efficiency [44, 48]. As a result, one interesting research question is to study *whether the verbalization and speech features extracted from older adults' think-aloud sessions can still be used to build effective ML models to detect when older adults encounter usability problems.*

Eighth, because the primary goal of this research was to understand if the verbalization and speech features identified in recent research [23] can be used to improve the automatic detection of usability problems, we did not optimize the parameters for each ML model (i.e., using default parameters) but rather focused on comparing the performance of the same type of ML models trained with and without the verbalization and speech features. However, the performance of each model can likely be improved by fine-tuning its parameters. For example, we initialized word embedding layer variables with random values from a uniform distribution for CNN and RNN models following common practice and examples from prior work [1, 26]. However, pre-trained word embedding, such as word2vec or GloVe embedding, and customized embedding trained on a similar dataset to ours might be able to improve the performance of CNN and RNN models.

Last, our approaches to determining verbalization and speech features can be further improved as well. For example, when determining if a user raises a question, we did not use punctuation marks. Further work might examine how to accurately determine punctuation marks, such as using automatic sentence boundary detection (e.g., Reference [46]) and speech-related features (e.g., raising tone), to better detect whether users ask questions. Additionally, we did not include words (e.g., do, does, will, shall) that are often used to raise polar questions in the question keywords list, because these words are not always used to raise questions and would require additional contextual information to determine whether they are indeed used to raise a question. Part-Of-Speech (POS) tags can be useful contextual information to help determine if such words are used to raise polar questions in conjunction with the keyword matching approach. In sum, *it is worth exploring more sophisticated methods to better detect verbalization and speech features.*

6 CONCLUSION

Fast-paced methods for analyzing recorded think-aloud sessions are needed to help UX evaluators leverage the benefit of large amounts of usability test sessions, which can be collected via remote usability testing. In this work, we took the first step to design and evaluate computational methods to automate the detection of the usability problem encounters in think-aloud test sessions. Our evaluations show that when using *the verbalization and speech features* (i.e., category, sentiment, question, negation, abnormal pitch, and abnormal speech rate) that are shown to be indicative of usability problems in recent research [23] as the input, the four types of ML models performed better compared to only using the generic text feature (i.e., TF-IDF or word embedding) as the input.

Furthermore, our evaluations show that it is possible to build an ML model for a *product* using its existing users' data to detect the usability problems encountered by a *new user*; it is also possible to build a *user-dependent* ML model for a *user* to detect usability problems encountered by the user when she uses a *new product*. As ML models achieved reasonable accuracy in detecting usability problem encounters, it is possible to examine how to leverage such models as "*virtual evaluators*" to provide a second perspective to UX practitioners in the future. Our evaluations also suggest that the *types of tasks* that users perform during think-aloud sessions may affect the models' performance. Future work should examine whether the type of tasks (i.e., guided and unguided tasks) used in think-aloud sessions and the difference in user's verbalization behavior affect the detection of usability problem encounters.

As the first step toward automating usability problems detection, our work focused on leveraging users' verbalization and speech features to detect the encounters of usability problems and

set a baseline for several technical and user interface design challenges that aim to integrate machine intelligence into UX practitioners' workflow to forge a sustainable symbiotic relationship. Future work should examine whether *other streams of data*, such as users' gaze-tracking data, facial expressions, actions on the interface, or physiological measures (e.g., galvanic skin response, heartbeat) are informative sources for detecting the encounters of usability problems and whether these streams of data can be used together with verbalization and speech features to further improve machine intelligence. Additionally, future work should consider how to design interactive intelligent user interfaces (IUIs) that leverage ML models that can detect usability problem encounters with reasonable but imperfect accuracy to support UX practitioners to analyze usability test sessions more effectively and also the IUIs that would allow UX practitioners to correct errors made by the machine intelligence interactively to improve its performance. Last, this work focused on detecting the encounter of usability problems but not the details about the problems. Future work could further investigate methods to infer the details about the problems, such as the description of the problems, the severity of the problems, and the potential solutions to the problems.

ACKNOWLEDGMENTS

We thank our reviewers and editors for their thoughtful and detailed feedback.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard et al. 2016. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI'16)*, 265–283.

[2] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Mag.* 35, 4 (2014), 105. DOI: https://doi.org/10.1609/aimag.v35i4.2513

[3] Morten Sieker Andreasen, Henrik Villemann Nielsen, Simon Ormholt Schrøder, and Jan Stage. 2007. What happened to remote usability testing?: An empirical study of three methods. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1405–1414. DOI: https://doi.org/10.1145/1240624.1240838

[4] Martin Blanchard, Nathaniel |D'Mello, Sidney |Olney, Andrew M.|Nystrand. 2015. Automatic classification of question & answer discourse segments from teacher's speech in classrooms. *Int. Educ. Data Min. Soc.* (2015). Retrieved from https://eric.ed.gov/?id=ED560555.

[5] Liora Bresler, Judy Davidson Wasser, Nancy B. Hertzog, and Mary Lemons. 1996. Beyond the lone ranger researcher: Team work in qualitative research. *Res. Stud. Music Educ.* 7, 1 (1996), 13–27. DOI: https://doi.org/10.1177/1321103X9600700102

[6] Anders Bruun, Peter Gull, Lene Hofmeister, and Jan Stage. 2009. Let your users do the testing: A comparison of three remote asynchronous usability testing methods. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI'09)*. 1619–1628. DOI: https://doi.org/10.1145/1518701.1518948

[7] Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. 2004. Ensemble selection from libraries of models. In *Proceedings of the 21st International Conference on Machine Learning (ICML'04)*. 18. DOI: https://doi.org/10.1145/1015330.1015432

[8] Kapil Chalil Madathil and Joel S. Greenstein. 2011. Synchronous remote usability testing: A new approach facilitated by virtual worlds. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'11)*. 2225–2234. DOI: https://doi.org/10.1145/1978942.1979267

[9] Elizabeth Charters. 2003. The use of think-aloud methods in qualitative research an introduction to think-aloud methods. *Brock Educ. J.* 12, 2 (2003), 68–82. DOI: https://doi.org/10.26522/brocked.v12i2.38

[10] Nan-Chen Chen, Margaret Drouhard, Rafal Kocielnik, Jina Suh, and Cecilia R. Aragon. 2018. Using machine learning to support qualitative coding in social science: Shifting the focus to ambiguity. *ACM Trans. Interact. Intell. Syst.* 8, 2 (2018). 9:1–9:20. DOI: https://doi.org/10.1145/3185515

[11] Parmit K. Chilana, Jacob O. Wobbrock, and Andrew J. Ko. 2010. Understanding usability practices in complex domains. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2337–2346. DOI: https://doi.org/10.1145/1753326.1753678

[12] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. (2014). Retrieved from http://arxiv.org/abs/1406.1078.

[13] Torkil Clemmensen, Qingxin Shi, Jyoti Kumar, Huiyang Li, Xianghong Sun, and Pradeep Yammiyavar. 2007. Cultural usability tests—How usability tests are not the same all over the world. In *Usability and Internationalization. HCI and Culture*. Springer Berlin, 281–290. DOI: https://doi.org/10.1007/978-3-540-73287-7_35

[14] Lynne Cooke. 2010. Assessing concurrent think-aloud protocol as a usability test method: A technical communication approach. *IEEE Trans. Prof. Commun.* 53, 3 (2010), 202–215. DOI: https://doi.org/10.1109/TPC.2010.2052859

[15] Kevin Crowston, Eileen E. Allen, and Robert Heckman. 2012. Using natural language processing technology for qualitative data analysis. *Int. J. Soc. Res. Methodol.* 15, 6 (2012), 523–543. DOI: https://doi.org/10.1080/13645579.2011.625764

[16] I. Dey. 1993. *Qualitative Data Analysis: A User-Friendly Guide for Social Scientists*. Routledge. DOI: https://doi.org/10.4324/9780203879276

[17] Thomas G. Dietterich. 2000. Ensemble methods in machine learning. Springer, Berlin, 1–15. DOI: https://doi.org/10.1007/3-540-45014-9_1

[18] Graham Dove, Kim Halskov, Jodi Forlizzi, and John Zimmerman. 2017. UX design innovation: Challenges for working with machine learning as a design material. In *Proceedings of the Chi Conference on Human Factors in Computing Systems*. 278–288.

[19] Margaret Drouhard, Nan Chen Chen, Jina Suh, Rafal Kocielnik, Vanessa Pena-Araya, Keting Cen, Xiangyi Zheng, and Cecilia R. Aragon. 2017. Aeonium: Visual analytics to support collaborative qualitative coding. In *Proceedings of the IEEE Pacific Visualization Symposium*. 220–229. DOI: https://doi.org/10.1109/PACIFICVIS.2017.8031598

[20] Upol Ehsan, Pradyumna Tambwekar, Larry Chan, Brent Harrison, and Mark O. Riedl. 2019. Automated rationale generation: A technique for explainable AI and its effects on human perceptions. In *Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI'19)*. 263–274. DOI: https://doi.org/10.1145/3301275.3302316

[21] Elling Sanne, Lentz Leo, and Menno De Jong. 2012. Combining concurrent think-aloud protocols and eye-tracking observations: An analysis of verbalizations. *IEEE Trans. Prof. Commun.* 55, 3 (2012), 206–220. DOI: https://doi.org/10.1109/TPC.2012.2206190

[22] K. Anders Ericsson and Herbert A. Simon. 1984. *Protocol Analysis: Verbal Reports as Data*. The MIT Press, Cambridge, MA.

[23] Mingming Fan, Jinglan Lin, Christina Chung, and Khai N. Truong. 2019. Concurrent think-aloud verbalizations and usability problems. *ACM Trans. Comput. Interact.* 26, 5 (2019), 1–35. DOI: https://doi.org/10.1145/3325281

[24] Asbjørn Følstad, Effie Law, and Kasper Hornbæk. 2012. Analysis in practical usability evaluation: A survey study. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2127–2136. DOI: https://doi.org/10.1145/2207676.2208365

[25] Mark C. Fox, K. Anders Ericsson, and Ryan Best. 2011. Do procedures for verbal reporting of thinking have to be reactive? A meta-analysis and recommendations for best reporting methods. *Psychol. Bull.* 137, 2 (2011), 316.

[26] Palash Goyal, Sumit Pandey, Karan Jain, Palash Goyal, Sumit Pandey, and Karan Jain. 2018. Research paper implementation: Sentiment classification. In *Deep Learning for Natural Language Processing*. Apress, 231–268. DOI: https://doi.org/10.1007/978-1-4842-3685-7_5

[27] Jonathan Grizou, I. Iturrate, Luis Montesano, Pierre-Yves Oudeyer, and Manuel Lopes. 2014. Interactive learning from unlabeled instructions. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI'14)*. Retrieved from http://auai.org/uai2014/proceedings/individuals/198.pdf.

[28] Jan Gulliksen, Inger Boivie, Jenny Persson, Anders Hektor, and Lena Herulf. 2004. Making a difference: A survey of the usability profession in Sweden. In *Proceedings of the 3rd Nordic Conference on Human-Computer Interaction (NordiCHI '04)*. 207–215. DOI: https://doi.org/10.1145/1028014.1028046

[29] Morten Hertzum, Pia Borlund, and Kristina B. Kristoffersen. 2015. What do thinking-aloud participants say? A comparison of moderated and unmoderated usability sessions. *Int. J. Hum. Comput. Interact.* 31, 9 (2015), 557–570. DOI: https://doi.org/10.1080/10447318.2015.1065691

[30] Morten Hertzum and Kristin Due Holmegaard. 2013. Thinking aloud in the presence of interruptions and time constraints. *Int. J. Hum. Comput. Interact.* 29, 5 (2013), 351–364. DOI: https://doi.org/10.1080/10447318.2012.711705

[31] Morten Hertzum and Niels Ebbe Jacobsen. 2001. The evaluator effect: A chilling fact about usability evaluation methods. *Int. J. Hum. Comput. Interact.* 13, 4 (2001), 421–443. DOI: https://doi.org/10.1207/S15327590IJHC1304_05

[32] Masahiro Hori, Yasunori Kihara, and Takashi Kato. 2011. Investigation of indirect oral operation method for think aloud usability testing. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 38–46. DOI: https://doi.org/10.1007/978-3-642-21753-1_5

[33] Paula Jarzabkowski, Rebecca Bednarek, and Laure Cabantous. 2015. Conducting global team-based ethnography: Methodological challenges and practical methods. *Hum. Relations* 68, 1 (2015), 3–33. DOI: https://doi.org/10.1177/0018726714535449

[34] Claire-Marie Karat, Robert Campbell, and Tarra Fiegel. 1992. Comparison of empirical testing and walkthrough methods in user interface evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'92)*. 397–404. DOI: https://doi.org/10.1145/142750.142873

[35] Jesper Kjeldskov, Mikael B. Skov, and Jan Stage. 2004. Instant data analysis: Conducting usability evaluations in a day. In *Proceedings of the 3rd Nordic Conference on Human-Computer Interaction (NordiCHI'04)*. 233–240. DOI: https://doi.org/10.1145/1028014.1028050

[36] Rafal Kocielnik, Saleema Amershi, and Paul N. Bennett. 2019. Will you accept an imperfect AI? exploring designs for adjusting end-user expectations of AI systems. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI'19)*. 1–14. DOI: https://doi.org/10.1145/3290605.3300641

[37] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Proceedings of the 29th AAAI Conference on Artificial IntelIIgence*. 333, (2015), 2267–2273. DOI: https://doi.org/10.1145/2808719.2808746

[38] Megh Marathe and Kentaro Toyama. 2018. Semi-automated coding for qualitative research. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI'18)*. 1–12. DOI: https://doi.org/10.1145/3173574.3173922

[39] Sharon McDonald, Helen M. Edwards, and Tingting Zhao. 2012. Exploring think-alouds in usability testing: An international survey. *IEEE Trans. Prof. Commun.* 55, 1 (2012), 2–19. DOI: https://doi.org/10.1109/TPC.2011.2182569

[40] Sharon McDonald and Helen Petrie. 2013. The effect of global instructions on think-aloud testing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'13)*. 2941–2944. DOI: https://doi.org/10.1145/2470654.2481407

[41] David Meignan, Sigrid Knust, Jean-Marc Frayret, Gilles Pesant, and Nicolas Gaud. 2015. A review and taxonomy of interactive optimization methods in operations research. *ACM Trans. Interact. Intell. Syst.* 5, 3 (2015), 1–43. DOI: https://doi.org/10.1145/2808234

[42] Jakob Nielsen. 1993. *Usability Engineering*. Elsevier.

[43] Mie Nørgaard and Kasper Hornbæk. 2006. What do usability evaluators do in practice? An explorative study of think-aloud testing. In *Proceedings of the 6th ACM Conference on Designing Interactive Systems (DIS'06)*. 209. DOI: https://doi.org/10.1145/1142405.1142439

[44] Erica Olmsted-Hawala and Jennifer Romano Bergstrom. 2012. Think-aloud protocols: Does age make a difference? In *Proceedings of the STC Technical Communication Summit*.

[45] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg et al. 2011. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* 12 (2011), 2825–2830.

[46] Jonathon Read, Rebecca Dridan, Stephan Oepen, and Lars Jørgen Solberg. 2012. Sentence boundary detection: A long solved problem? In *Proceedings of the International Conference on Computational Linguistics (COLING'12)*. 985–994.

[47] Qingxin Shi. 2008. A field study of the relationship and communication between Chinese evaluators and users in thinking aloud usability tests. In *Proceedings of the 5th Nordic Conference on Human-computer Interaction Building Bridges (NordiCHI'08)*. 344. DOI: https://doi.org/10.1145/1463160.1463198

[48] Andreas Sonderegger, Sven Schmutz, and Juergen Sauer. 2016. The influence of age in usability testing. *Appl. Ergon.* 52, (2016), 291–300. DOI: https://doi.org/10.1016/j.apergo.2015.06.012

[49] Howard Tamler. 1998. How (much) to intervene in a usability testing session. *Common Gr.* 8, 3 (1998), 11–15.

[50] Karel Vredenburg, Ji-Ye Mao, Paul W. Smith, and Tom Carey. 2002. A survey of user-centered design practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Changing Our World, Changing Ourselves (CHI'02)*. 471. DOI: https://doi.org/10.1145/503457.503460

[51] Zuowei Wang, Xingyu Pan, Kevin F. Miller, and Kai S. Cortina. 2014. Automatic classification of activities in classroom discourse. *Comput. Educ.* 78, (2014), 115–123. DOI: https://doi.org/10.1016/J.COMPEDU.2014.05.010

[52] Brad Wuetherick. 2010. Basics of qualitative research: Techniques and procedures for developing grounded theory. *Can. J. Univ. Contin. Educ.* 36, 2 (2010). DOI: https://doi.org/10.21225/D5G01T

[53] Jasy Liew Suet Yan, Nancy McCracken, and Kevin Crowston. 2014. Semi-automatic content analysis of qualitative data. In *Proceedings of the iConference*. DOI: https://doi.org/10.9776/14399

[54] Jasy Liew Suet Yan, Nancy McCracken, Shichun Zhou, and Kevin Crowston. 2014. Optimizing features in active machine learning for complex qualitative content analysis. In *Proceedings of the ACL Workshop on Language Technologies and Computational Social Science* 56, Ml (2014), 44–48. DOI: https://doi.org/10.3115/v1/w14-2513

[55] Qian Yang, Alex Scuito, John Zimmerman, Jodi Forlizzi, and Aaron Steinfeld. 2018. Investigating how experienced UX designers effectively work with machine learning. In *Proceedings of the Designing Interactive Systems Conference (DIS'18)*. 585–596. DOI: https://doi.org/10.1145/3196709.3196730

[56] Qian Yang, John Zimmerman, Aaron Steinfeld, and Anthony Tomasic. 2016. Planning adaptive mobile experiences when wireframing. In *Proceedings of the ACM Conference on Designing Interactive Systems*. 565–576.

[57] Tingting Zhao, Sharon McDonald, and Helen M. Edwards. 2014. The impact of two different think-aloud instructions in a usability test: A case of just following orders? *Behav. Inf. Technol.* 33, 2 (2014), 162–182. DOI: https://doi.org/10.1080/0144929X.2012.708786

[58] Haiyi Zhu, Robert E. Kraut, Yi-Chia Wang, and Aniket Kittur. 2011. Identifying shared leadership in Wikipedia. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'11)*. 3431. DOI: https://doi.org/10.1145/1978942.1979453
[59] Christian's Python Library: A Python library for voice analysis. Retrieved from https://homepage.univie.ac.at/christian.herbst/python/namespacepraat_util.html.
[60] Praat: Doing Phonetics by Computer. Retrieved from http://www.fon.hum.uva.nl/praat/.
[61] Sound: To Pitch (ac)... Retrieved from http://www.fon.hum.uva.nl/praat/manual/Sound__To_Pitch__ac____.html.
[62] tf.random.uniform | TensorFlow Core r2.0. Retrieved from https://www.tensorflow.org/api_docs/python/tf/random/uniform.

Received May 2019; revised February 2020; accepted February 2020