Real-Time Interruption Management System for Efficient Distributed Collaboration in Multi-tasking Environments

ASHUTOSH SHIVAKUMAR, Wright State University, USA
AISHWARYA BOSITTY, Wright State University, USA
NIA S. PETERS, Air Force Research Laboratory, USA
YONG PEI, Wright State University, USA

Interruption dissemination in proactive systems remains a challenge for efficient human-machine collaboration, especially in real-time distributed collaborative environments. In this paper a real-time interruption management system (IMS) is proposed that leverages speech information, the most commonly used and available means of communication within collaborative distributed environments. The key aspect of this paper includes a proposed real-time IMS system that leverages lexical affirmation cues to infer the end of a task or task boundary as a candidate interruption time. The performance results show the proposed real-time lexical *Affirmation Cues based Interruption Management System* (ACE-IMS) outperforms the current baseline real-time IMS system within the existing literature. ACE-IMS has the potential of reducing disruptive interruptions without incurring excessive missed opportunities to disseminate interruptions by utilizing only the most frequently used mode of human communication: voice. Thereby, providing a promising new baseline to further the system development of real-time interruption management systems within the ever-growing distributed collaborative domain.

CCS Concepts: • **Human-centered computing → Collaborative and social computing systems and tools**; **Interactive systems and tools**; **Collaborative interaction**.

Additional Key Words and Phrases: Interruption management systems; human-machine collaboration; human-machine teaming

ACM Reference Format:
Ashutosh Shivakumar, Aishwarya Bositty, Nia S. Peters, and Yong Pei. 2020. Real-Time Interruption Management System for Efficient Distributed Collaboration in Multi-tasking Environments. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW1, Article 39 (May 2020), 23 pages. https://doi.org/10.1145/3392844

1 INTRODUCTION

Interruption science explores the disruptiveness of interruptions on human performance. This research area is motivated by the reality that as users increasingly multi-task with proactive systems, their tasks are being interrupted more often. An interruption within these interactions can be defined as an unanticipated request for task switching from a person, an object, or an event while multi-tasking [8]. The disruptiveness of interruptions within this context has been widely studied such as the implications of interruptions on productivity [4, 10, 12, 13] and affective state [5, 35]. For instance, previous studies have illustrated that interrupting users engaged in tasks has a considerable negative impact on task completion time [9, 11–13, 21, 25]. Interrupting tasks at

Authors' addresses: Ashutosh Shivakumar, shivakumar.5@wright.edu, Wright State University, Dayton, Ohio, 45435, USA; Aishwarya Bositty, Wright State University, Dayton, Ohio, 45435, USA, Bositty.2@wright.edu; Nia S. Peters, nia.peters.1@us.af.mil, Air Force Research Laboratory, Wright-Patterson Air Force Base, Ohio, USA; Yong Pei, yong.pei@wright.edu, Wright State University, Dayton, Ohio, 45435, USA.

2573-0142/2020/5-ART39 $15.00
https://doi.org/10.1145/3392844

random moments can cause users to take up to 30% longer to resume the tasks, commit up to twice the errors, and experience up to twice the negative effect than when interrupted at boundaries [5, 10, 19]. Other studies illustrate the implications of ill-timed interruptions particularly in medical settings from an inter-clinician communications perspective [15] and from work design and systems or processes' perspective in hospitals [17] or the cost of interruptions [7, 12, 26, 35] which some have suggested are attributed to differences in workload at the point of interruption [10].

The main aim of this research work and overall contribution to the literature is a proposed real-time interruption management system that explores the use of affirmation cues as lexical information and corresponding performance evaluation of this system within multi-user, multi-tasking distributed collaborative interactions. Interactions here are referred to task-oriented dialogues where participants communicate with one another verbally to accomplish a task at hand [18]. Within these interactions, there are two or more people not only collaborating, but also multi-tasking in distributed environments where speech is the most salient communication modality. An example is emergency management involving communication operators and first responders. Here the operator at the command center communicates with the emergency personnel on the ground and aligns his/her knowledge of the location of the hazard. The command center operator has two different tasks to perform simultaneously: 1.) Primary task: location alignment with first responder concerning the emergency; 2.) Secondary task: to monitor the system for other emergencies or system maintenance alerts. Other multi-user, multi-tasking distributed interactions include air traffic controllers and pilots, unmanned aerial systems (UAV operators) and military ground troops, and technical support agents and customers. Frequently interrupting the users in these scenarios with orthogonal tasks [31] or interruption task can lead to cognitive overload with potentially devastating consequences where the participant may be distracted and overwhelmed to complete their primary task effectively and efficiently [22].

2 RELATED WORK

To alleviate the consequences of the disruptiveness, manipulating the timing of interruptions [4, 10, 12, 13] using system-mediated interruptions [24] within multi-task environments [27] has been proposed and studied for different timing strategies. Interruption times explored include immediate delivery [12, 14, 23], random timing [4, 10, 12, 21, 23, 32], and delivery at task boundaries [5, 10, 12, 20] as examples. The benefit of appropriately timed interruptions, particularly the task-boundary based approach, is evident in works such as [29].

2.1 Interruption Management and Task Structure

One particular area of research that aims to alleviate the negative effects of these interruptions via system-mediated interruptions is the Interruption Management Systems (IMS) literature. The focus of this area is to leverage the available modalities of an interaction (i.e., visual, meta-data, speech etc.) within domains of varying participants, tasks, and objectives in order to disseminate information at the least disruptive times. Within this literature methods were proposed to determine the appropriate interruption timings via task structure inference, and a subset of this literature recommends point of interruptibility at boundaries within task execution. A task boundary is a time instance between two moments of task execution. Within single-user, multi-tasking interactions, task boundary modeling has been used to indicate appropriate points of interruptibility via system-state [5, 10, 12, 20] and physiological data [6].

2.2 Collaborative Communication Interruption Management System

Until recently the exploration of task boundary modeling to infer interruption decisions has been limited to single-user, multi-tasking interactions. The Collaborative Communication Interruption

Management or C-CIMS proposed by [31] extends the Interruption Management and Task Structure literature and aims to use task boundary modeling for interruption inference within distributed multi-user, multi-tasking interactions, as illustrated in Fig. 1. It laid out the foundation to extend the use of task boundary modeling for interruption inference within distributed (users can reside in different geographical locations at the same time), multi-user, multi-tasking domain.

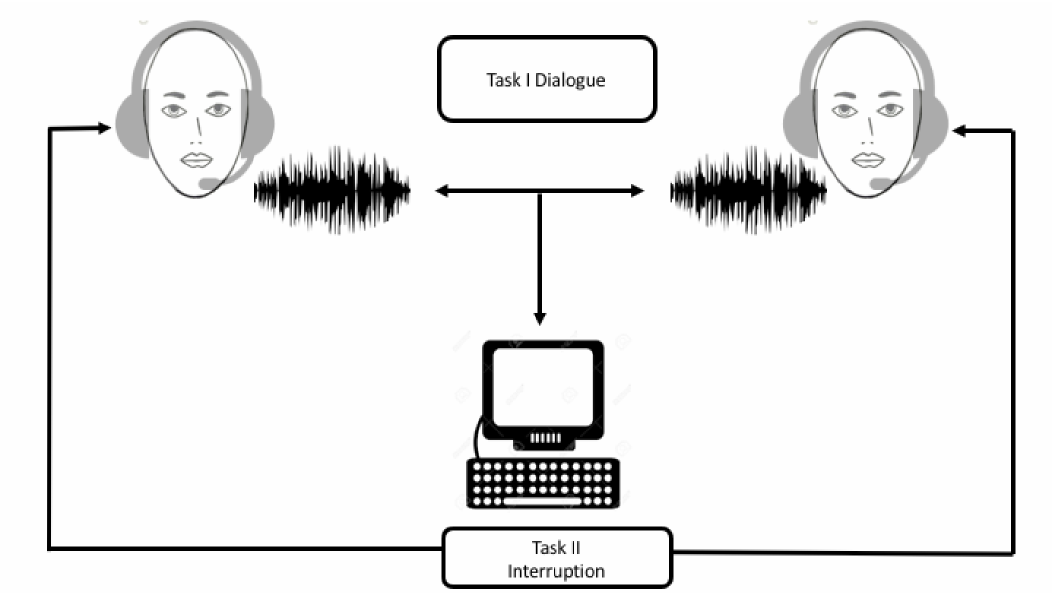


Fig. 1. Distributed Multi-user, Multi-tasking Interaction [31]

C-CIMS [31] leverages speech information within the distributed multi-user, multi-tasking interactions and aims to infer a task boundary as candidate points of interruption. C-CIMS explored this problem using offline (non-real-time) and online (real-time) machine learning techniques which train and test their proposed model on the entire available data collection. The offline implementation of C-CIMS explored lexical features ("what is said?"), and appears to offer increased performance in detecting task boundaries to infer interruption timings when compared to a prosodic-only implementation that leverages only prosodic information ("How it is said?"), such as: energy and pitch information. The real-time implementation of C-CIMS established a baseline performance for real-time IMS system in this domain. However, it leverages only prosodic information, and exhibits limited performance. The limitation of real-time C-CIMS to prosody-only model was reported due to prohibitive latency issues in processing lexical information in real-time [31].

Therefore, there is a need to explore a lexical-based interruption management system that can support real-time interactions. Additionally, within the offline models, the author of [31] inferred that affirmation cues are salient lexical predictors of a task boundary, although without providing a comprehensive investigation.

3 PROPOSED REAL-TIME IMS SOLUTION

The main aim of this research work and overall contribution to the literature is a proposed real-time interruption management system that leverages lexical information and the corresponding performance evaluation of this system within multi-user, multi-tasking distributed collaborative interactions. This is accomplished by comparing the proposed real-time lexical system performance to the baseline real-time prosody-only system of C-CIMS. Additionally, the aim is to explicitly explore lexical affirmation cues' contribution to enable the proposed lexical model in making interruption decisions. Affirmation cues are explored here because existing literature indicates humans tend to use affirmation cues such as *like, got it, or yeah* to signal transition to another topic or task and to signal turn-taking in task-oriented dialogue [16, 30]. Since the objective of this system is to predict a task boundary as a candidate interruption point which is defined as a time instance between two moments of task execution, we speculate detection of an affirmation cue can predict such moments. Hence, considering this background, we explore the questions in section 3.1 to guide our research.

3.1 Research Questions

The following research questions (RQ) bring this research work into perspective:

- **RQ1**: If affirmation cues signal task transitions [16], what is the extent of their occurrence prior to a task boundary in a task-oriented dialogue?
- **RQ2**: Can a real-time system be built to explore these lexical affirmation cues and identify task boundaries as potential points of interruption in distributed multi-user, multi-tasking environments?
- **RQ3**: Does the proposed real-time IMS outperform the baseline real-time C-CIMS [31] in accurately detecting task boundaries as candidate interruption times?

3.2 Contributions

The contributions of this research work involve an explicit and systematic exploration of implicit insights on affirmation cues observed in C-CIMS [31] and utilizing these observations to successfully develop a fully functional real-time Lexical *Affirmation Cues based Interruption Management System* (ACE-IMS). The three primary contributions of this paper include:

(1) Explicit exploration of affirmation cues and their relationship to task boundaries and evaluate their effectiveness as lexical features in inferring task boundaries as candidate interruption points.
(2) A real-time lexical based task boundary inference model as a proposed interruption management system to address the lexical processing bottleneck of C-CIMS [31].
(3) An improved new baseline real-time IMS system within distributed multi-user, multi-tasking domain that can also be readily adapted to support applications with potentially changing application requirements with respect to precision, recall and delay, during the course of its operation.

4 METHOD

The primary focus of this research paper is to explore the usage of affirmation cues to identify task boundaries in real-time for intelligent interruption dissemination in multi-user multi-tasking interactions. We focus on task-oriented dialogues that simulate multi-user, multi-tasking dialogues, where participants communicate with one another verbally to accomplish a task at hand. To accomplish this strategy, we follow these steps:

(1) Assess and understand the datasets: here we describe our task boundary annotated task – oriented dialogue datasets.
(2) Analyze and gain insights of affirmation cues preceding task boundaries.
(3) Provide a system design and prototype of an Affirmation Cues based Interruption Management System (ACE-IMS) to demonstrate real-time identification of task boundary.

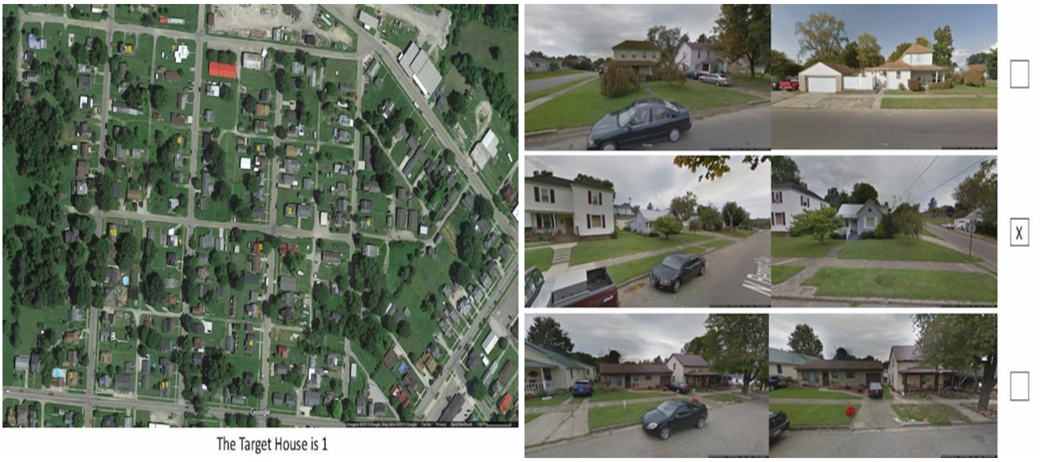
(4) Develop a progressive rule–set design of affirmation cues which forms the heart of the ACE-IMS.

Below subsections describe each of these steps in detail.

4.1 Dataset

The proposed ACE-IMS is trained and tested using two human–human task datasets from the research work in [31]: UMT and Tangram. The two datasets represent the domain of interest: distributed multi-user, multi-tasking task-oriented dialogues.

In these tasks two distributed human participants communicate using push-to-talk to accomplish a common task and the machine disseminates information related to an orthogonal task or interruption task. A brief description of the datasets is as follows:



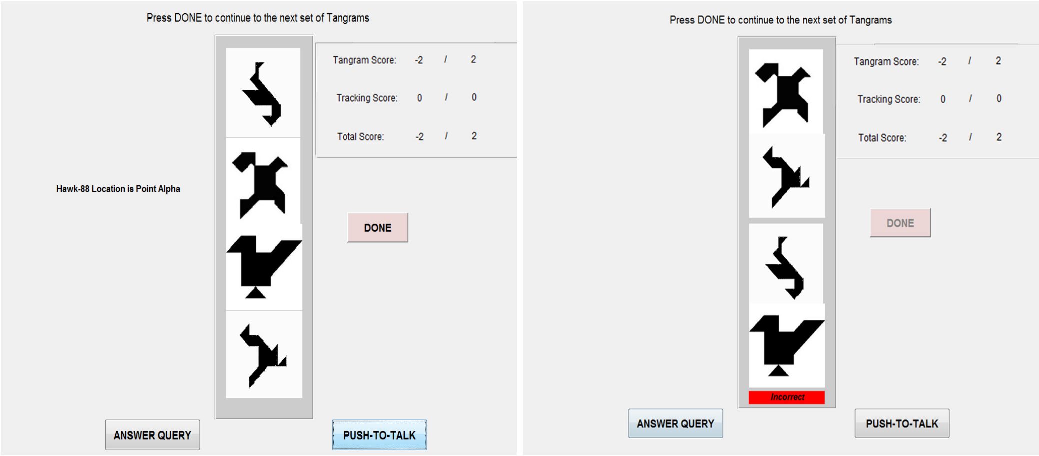
(a) Aerial Target View from UMT Task

(b) Street View identification from UMT task

Fig. 2. UMT Task [31]

- Uncertainty Map Task (UMT): The UMT is a distributed multi-user collaborative communication task where the two participants align their knowledge to identify a target house while looking at the house from a different perspective: birds eye/aerial or forward facing / street view. In the task, two participants are presented with one of these 4 target house views: a.) aerial target vs. street view identification, b.) street view target–aerial identification, c.) street view target–street view identification, and d.) aerial/street view target and identification. There is a total of 67 dual-channel audio files (one audio channel per speaker) for the UMT task that are used in our project. Each audio file consists of task conversations corresponding to 10 target identification tasks as illustrated in Fig. 2.
- Tangram: The Tangram task is a distributed multi-user collaborative communication task where participants use a push-to-talk to communicate on a task where they arrange the abstract shapes called Tangrams in corresponding order that is aligned with each other as illustrated in Fig. 3. 40 dual-channel audio files from Tangram are used in our project, each consisting of task operator–teammate conversations (one channel per speaker).

Fig. 2 and Fig. 3 are interface diagrams of the tasks respectively and more information about the data collection is available in [31].



(a) Tangram Task Teammate 1 interface (b) Tangram Task Teammate 2 interface

Fig. 3. Tangram Task [31]

4.1.1 *Metadata.*

The accompanying log file for the audio files in both datasets consists of task begin and end time. The task begin time is defined as the time at which both users were presented with a new set of targets (UMT task) or abstract shapes (Tangram task), and the task end time is defined as the time instance when both participants mutually acknowledge the completion of a task with a mouse click on "Done" button [30].

The timing of delivering information related to an orthogonal task or interruption task while participants are performing the primary task (UMT or Tangram) is the focus of this paper. Adding another task, i.e., orthogonal task as specified in [31] would make them multi-user, multi-tasking interactions.

Since the overall objective of this paper is to present a real-time IMS system that leverages affirmation cues as lexical features to infer a task boundary, we first create the training dataset and the testing datasets and explore how much the affirmation cues may account for the identification of task boundaries.

4.1.2 *Training and Testing datasets.*

- Training Dataset: A random portion of the UMT dataset 30 audio files (3066 utterances) out of 67, was designated as the training dataset and used to identify the affirmation cues and generate the rules of the classifier. Refer Table 1.
- Testing Datasets: The remaining 37 UMT audio files (2904 utterances) were added to the original 30 to create a 67 audio files (5970 utterances) test dataset. Additionally, 40 audio

Table 1. Training and Testing datasets

Dataset	**Audio Files**	**Utterances**
Training Dataset	30 randomly selected audio files from the UMT dataset (approximately 6 hours)	3066
UMT Test Dataset	67 (approximately 13 hours)	5970
Tangram Test Dataset	40 (approximately 8 hours)	4554

Table 2. Affirmation Cues Count for the N=329 task boundary utterances within the Training Dataset

Affirmation Cues	**Frequency in Task Boundaries**
got it	180
got you	13
yep	14
gotcha	2
awesome	3
sounds good	4
done	9
great	3

files (4554 utterances) of the Tangram dataset was also used as an additional test dataset to evaluate the generalizability of the identified affirmation cues as lexical features. Refer Table 1.

Since multiple audio recordings may come from the same participant, the audio files within each dataset were randomly selected to incorporate more speaking styles for variety in affirmation cues for both training and testing. Each channel of the dual–channel audio files from the training and test data was passed through automatic speech recognition for speech to text conversion. The resulting text transcripts of the two separate channels were then interleaved together with the aid of timestamps to create the dialogue text transcripts. For clarification, the timestamps were sorted in chronological order and the corresponding utterances were added to create dialogue text transcripts. Each audio file had a corresponding dialogue transcript file. The utterance that preceded the task boundary timestamp, as provided in the corresponding task log files, was labelled as the task boundary utterance. For a single dataset, for example, the training dataset, all dialogue transcript files corresponding to the audio files in the dataset are interleaved to form a 3066-utterance dataset. Similar operations were performed on UMT test and Tangram test datasets. The task start and end time in the dataset provide structure to a task-oriented dialogue. This gives us an excellent opportunity to delve into lexical affirmation cues preceding task end time to examine our Research Question 1 (RQ1) in Section 3.1.

4.2 Affirmation Cues Preceding Task Boundaries

As indicated by [16, 30] humans tend to use affirmation cues such as *like, got it* or, *yeah* to signal transition to another topic or task and to signal turn-taking. Since the objective of the proposed interruption management system is to predict a task boundary or a task transition as a candidate interruption point, we expect that detection of an affirmation cue can predict such moments.

To explore the use of affirmation cues and their relationship to task boundary utterances, we examine the existence of lexical features reflecting affirmation cues in the training dataset. The definition of a *task boundary* as presented in [30] is *a timestamp associated with both players clicking a button to indicate they are done with one task and ready to transition another task.* We then define *a task boundary utterance as the utterance immediately preceding this task boundary timestamp.*

The nine most frequent affirmation cue phrases present in task boundary utterances within the training dataset were manually identified and recorded. The list is shown in Table 2.

Then, we proceed to identify the occurrence of the identified affirmation cues in the labelled task boundary utterances for the training dataset and the two test datasets and report the results in

Table 3. Coverage calculation of each dataset

Dataset	Total number of Task Boundary Utterances With The Identified Affirmation Cues (k)	Total Number of Task Boundary Utterances (N)
UMT Training	230	329
UMT Test	508	808
Tangram Test	1020	1158

Table 3. Here we use a term called *Coverage* to measure the extent of affirmation cue occurrence within the task boundary utterances, as defined by Equation 1.

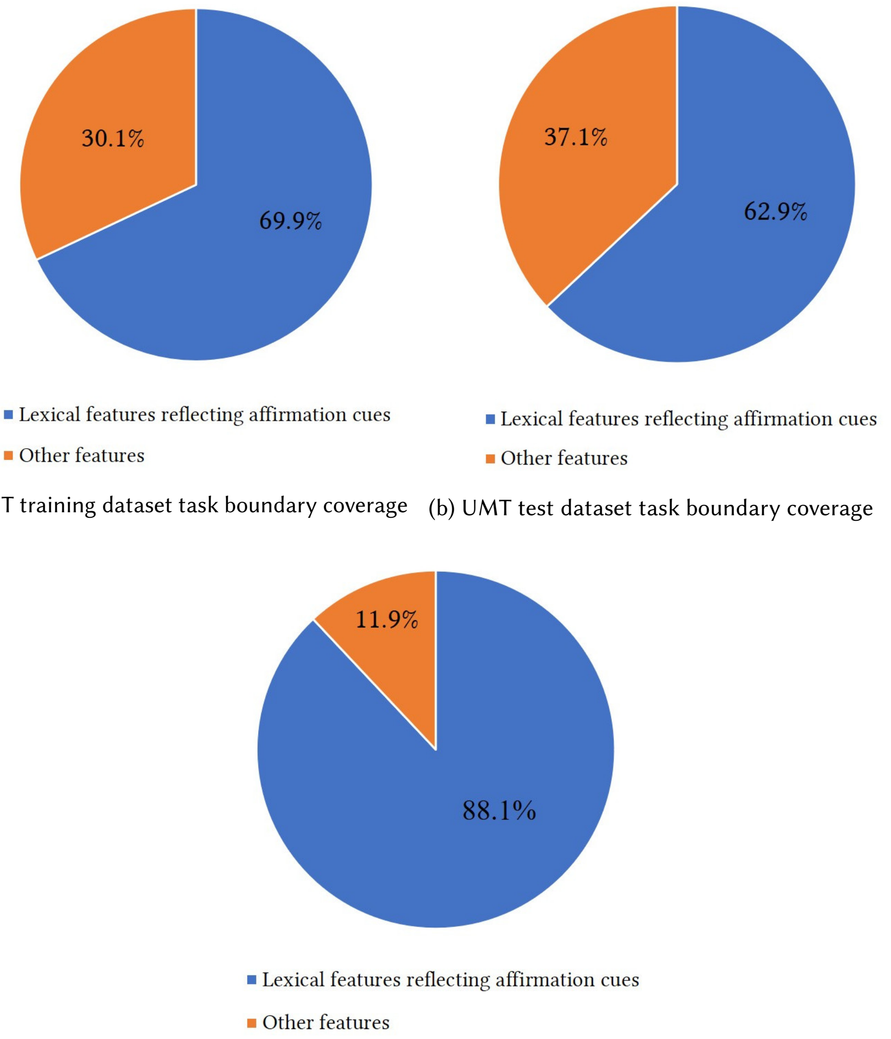
$$Coverage = k/N \tag{1}$$

Where k is the total number of task boundary utterances with the identified affirmation cues in the corresponding dataset, and N is the total number of task boundary utterances for the corresponding dataset. For the training dataset here, $N = 329$.

Fig. 4 shows the *Coverage* of the identified affirmation cues (as listed in Table 2) among task boundaries in the training dataset and the UMT and Tangram test datasets, respectively. Fig. 4a indicates that affirmation cues present in 69.9% of the total task boundary utterances in the training dataset (with total number of task boundary utterances N = 329), the remaining 30.1% can be mapped to other unexplored features. Fig. 4b shows that when the same identified affirmation cues are applied to the UMT test dataset task boundaries (N = 808), the coverage decreases by 7% to 62.9%. Fig. 4c indicates that, for Tangram test dataset, the same identified affirmation cues account for 88.1% of the total task boundaries (N = 1158), which is higher compared to both the UMT-based training dataset (a random selection from UMT as defined in Section 4.1.2) and the UMT test dataset. These results provide us with the following insights:

- The 9 affirmation cues present in Table 2 are strong feature candidates for identifying task boundary utterances
- The higher Coverage in the Tangram test dataset (88.1%) when using the affirmative cues obtained from the UMT-based training dataset implies that:
  (1) the identified lexical features from UMT-based training dataset generalize well and perform robustly across both datasets (UMT and Tangram).
  (2) the affirmation cues present at a higher rate in the task-boundary utterances of Tangram tasks, may imply that the dialogues of Tangram tasks could be more structured. Future investigation is warranted to identify the causes of such variations among task-oriented dialogues.
- The remaining task boundaries, those without affirmation cue phrases presented (e.g., 30.1% of task boundaries in the training dataset, 37.1% in the UMT test dataset and 11.9% in the Tangram test dataset), could be the focus of future research topics beyond the scope of this work.

4.2.1 Interference from Backchannel Utterances. Further examination also indicate that the same affirmation cues present in task boundary utterances may also be used as backchannels in a task-oriented dialogue. In the context of this work, backchannels are defined as verbal cues that represent continuity in a task-oriented dialogue [16]. For example, the affirmation cue *yep* could indicate continuity in a conversation by the interlocutor while also functioning as an affirmation cue indicating a task boundary.



(a) UMT training dataset task boundary coverage

(b) UMT test dataset task boundary coverage

(c) Tangram testing dataset task boundary coverage

Fig. 4. Coverage of extracted affirmation cues in Task boundaries

This could potentially result in false identification of a task boundary (i.e., false positive) when using affirmation cues; and eventually, lead to disruptive interruptions. Thus, while adding more affirmation cues into the feature set may improve the *Coverage* of the ACE-IMS, i.e., reduce the missed interrupting opportunities, it has the risk of increasing undesirable disruptive interruptions. Clearly, reducing false and missed interruptions are two conflicting objectives. Moreover, different distributed collaborative applications may prioritize them differently, for example some professions may be tolerant to interruptions from frequent alarms even if they are false rather than miss an alarm altogether and which lead to potentially disastrous situations. Therefore, there is a need for a balanced and flexible design approach that support application-specific operation requirements through convenient system adaptations. In short, the *Coverage* data of the affirmation cues from Fig. 4 inform us that they account for the majority of the task-boundary phrases addressing our first research question (RQ1: If affirmation cues signal task transitions [16], what is the extent of their occurrence prior to a task boundary in a task-oriented dialogue?). Thus, we can proceed with an implementation that utilizes these features to disseminate real-time interruptions. This is addressed in detail in Sections 4.3 and 4.4.

4.3 Real-time Affirmation Cues based Interruption Management System (ACE-IMS)

The proposed ACE-IMS solution addresses the issue of processing lexical information in real-time for the purpose of making intelligent interruption decisions. The developed prototype serves to validate its operation within real-time interactions. The prototype specifically emphasizes key phrases associated with task boundaries that reflect affirmation cues. For this reason, a rule-based classification approach is proposed which in future work can be expounded upon to consider other machine learning and deep learning modeling approaches to create a hybrid architecture. The system design is shown in Fig. 5.

The system consists of a multi-channel audio input, i.e., one audio input per user, that records voice data and relays its digital manifestation to an acoustic preprocessor. The acoustic preprocessor reduces the noise and fine-tunes the gain of the audio using Audacity API (Application Programming Interface) [2]. The preprocessed audio is then sent to cloud-based Automatic Speech Recognition (ASR) engine, e.g., the Google Cloud Speech service in our implementation, which uses a server-client implementation for speech to text transcription [1]. The real-time ASR is one of the key components that enables real–time operation of the proposed ACE-IMS in addition to the lexical analysis system. The adoption of the widely available cloud-based ASR services, such as Google Cloud speech, helps mitigate the potentially prohibitive computation burden of running an ASR on the local machine and, ultimately, the delay associated with high-accuracy speech recognition (WER=4.9%, where WER stands for “Word Error Rate”)[33]. Our experimental studies observe that

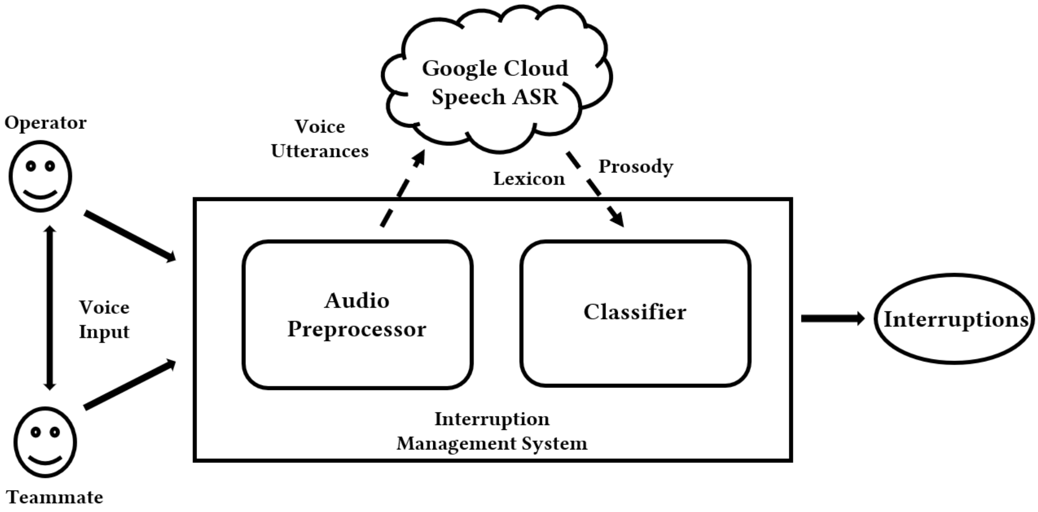


Fig. 5. System design of the Interruption Management System

the real-time ASR latency is generally under 350 ms. The resulting text utterances are then fed into a progressive rule-based classifier that controls and disseminates the interruptions in real-time.

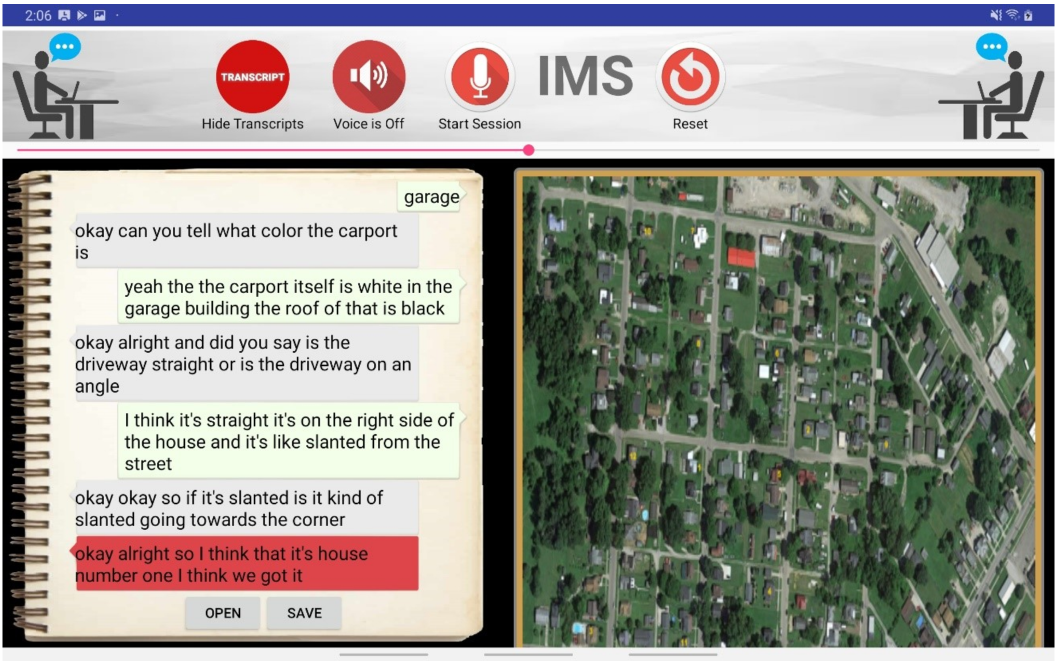
To visualize and evaluate the feasibility of real-time task boundary detection capability, an Android-based prototype of the ACE-IMS is implemented and illustrated in Figs. 6 and 7. The implementation consists of two Android tablets. Fig. 6 and 7 illustrate the interfaces of the ACE-IMS for supporting the distributed operations of a UMT task.

The dialogue visualizer on the left side of the interface displays the real-time speech-to-text output from ASR for the two-person dialogue within a distributed multiuser multi-tasking interaction. This visualizer can be toggled on and off. These capabilities allow researchers to view the dialogue and interruption decisions made by the IMS in real-time. The dialogue highlighted in red indicate the task boundaries identified by the ACE-IMS running in the background.

On the right side of the interface in Fig. 6 and Fig. 7 is a customizable primary task interface (e.g. Aerial view in Fig. 6 or street view in Fig. 7 for supporting the UMT tasks). This portion of the interface can be customized to any visual interface that is conducive to simulating a task within the domain of interest. For instance, we can change the interface and interaction so that players are engaged in the Tangram task or other collaborative tasks as opposed to UMT. The prototype also provides features like data collection for both voice and text. This data can be further used to expand the existing dataset, train the classifier and improve the accuracy of task boundary identification and other interruption inference models. In section 4.4 we present the progressive rule-set design of the classifier and its implications.

4.4 Progressive Rule-Set Design

Since a primary contribution of this work is to focus on affirmation cues as an indicator of a task boundary for intelligent interruption dissemination, the affirmation cues need to be identified in



garage

okay can you tell what color the carport is

yeah the the carport itself is white in the garage building the roof of that is black

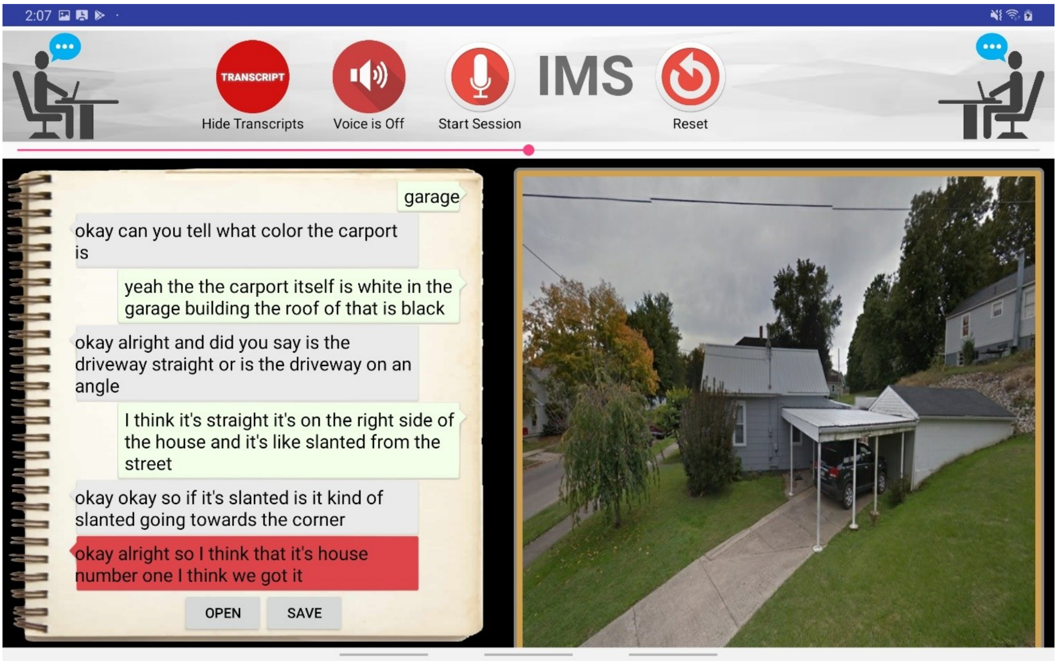
okay alright and did you say is the driveway straight or is the driveway on an angle

I think it's straight it's on the right side of the house and it's like slanted from the street

okay okay so if it's slanted is it kind of slanted going towards the corner

okay alright so I think that it's house number one I think we got it

Fig. 6. ACE-IMS Prototype supporting a UMT Task (Aerial View)



garage

okay can you tell what color the carport is

yeah the the carport itself is white in the garage building the roof of that is black

okay alright and did you say is the driveway straight or is the driveway on an angle

I think it's straight it's on the right side of the house and it's like slanted from the street

okay okay so if it's slanted is it kind of slanted going towards the corner

okay alright so I think that it's house number one I think we got it

Fig. 7. ACE-IMS Prototype supporting a UMT Task (Street View)

speaker utterances. Hence, a rule-based classifier is the first pass at implementing the proposed real-time ACE-IMS. A rule–based classifier provides the following advantages in the context of task boundary classification:

(1) Deterministic decision-making based on domain specific features that indicate task completion,
(2) Flexible operating points that trade off between missed interrupting opportunities and disruptive interruptions that prioritize application requirements,
(3) Flexible feature adaptation: to accommodate multiple modalities of data and revise them based on the availability of new features.

These advantages make the rule-based classifier a viable candidate for classifying task boundaries in IMS. Mapping affirmation cues to utterances may seem to be a straight-forward task where one must parse utterances for affirmation cues to determine task-boundaries. However, due to the interference from backchannels as discussed earlier in Section 4.2.1, it is not so trivial when optimality, scalability, adaptability and real-time needs of interruption dissemination are brought into focus. Let us consider the problem of designing the optimal collection of affirmation cue features. In the context of this research, based on the initial assessment of coverage within the training dataset, we have short-listed 9 affirmation cues, as shown in Table 2. There is a chance that these affirmation cues may also appear as backchannels in non-task boundary utterances. Reducing false and missed interruptions are two conflicting objectives, and this brings us to our first question: *what should be the objective function that is used to optimize the rule–set with conflicting goals of minimizing false interruptions and minimizing missed interruptions?*

Furthermore, if the application domain of the IMS dictates that interruptions must be disseminated frequently, but is tolerant to the number of false interruptions or vice versa, *How can the user* *modify the behavior of the classifier to facilitate application-specific interruption dissemination?* And most importantly *How can these rules be selected and arranged to enable real–time operation?* In the following subsections we address these concerns by first presenting our proposed progressive rule-set design method in detail and comment on its effectiveness in addressing the challenges.

4.4.1 Objective Function and Optimization Algorithm. To take a more balanced consideration for performance assessment, we will use the F1 score used in [31], which is a combined measure of both false interruptions and missed interruptions. To determine the right sequence of affirmation cues-based rules that support progressive operation, we adopt the *steepest ascent method for multivariate optimization* [3] which optimizes *F1 score*, a multivariate objective function. For each iteration of the algorithm, we calculate the F1 score and the delta F1 score. The formula for F1 score is given in Equation 2:

$$F1\,score = (2 * Precision * Recall)/(Precision + Recall) \tag{2}$$

Where,

$$Precision = True\,Positives/(True\,Positives + False\,Positives)$$

and

$$Recall = True\,Positives/(True\,Positives + False\,Negatives)$$

Here $True\,Positives$ are utterances that are correctly identified as task boundaries, $False\,Positives$ or false interruptions are non-task-boundary utterances which are wrongly identified as task boundaries, and $False\,Negatives$ or missed interruptions are task-boundary utterances which are wrongly identified as non-task-boundary utterances. Within each iteration, the $\Delta F1\,score$ is given in Equation 3:

$$\begin{aligned}\Delta F1\,score = F1\,score(current\,set\,of\,lexical\,affirmation\,cues + new\,lexical\,affirmation\,cue) - \\ F1\,score(current\,set\,of\,lexical\,affirmation\,cues)\end{aligned} \tag{3}$$

Then, the *current set of lexical affirmation cues* is updated by adding the lexical feature that produces the maximal $\Delta F1\,score$ improvement at each iteration.

4.4.2 Iteration–Wise Description of Affirmation Cue based Rule-Set. The iteration-wise development of lexical affirmation cue-based classifier is shown in Table 4 with the optimal rule-set highlighted in bold. Additionally, a graphical representation of affirmation cue development of Table 4 is rendered in Fig. 8 where the X-axis consists of *Iteration Number or Operating Point* and the Y axis represents the performance measures in F1 score, Precision or Recall. Iteration Number corresponds to the iteration of steepest search algorithm. Operating point is the F1 score that characterizes the performance of the progressive rule-set based classifier. In addition to F1 score, Precision and Recall are also presented to understand the contribution of each affirmation cue to false interruptions and missed interruptions in greater detail. The characteristics for the lexical affirmation cue classifier is displayed in dashed lines.

A detailed description of the operations performed in the first 2 out of 9 iterations of the steepest ascent search algorithm is described with the aid of an illustration in Fig. 9. Although *Iteration 0* is mentioned in the description as the initial iteration with an empty rule-set, it is done for theoretical purposes to serve the conceptual explanation of the steepest ascent search algorithm while for all purposes of implementation the operations begin from Iteration 1.

Iteration 1: The F1 scores of the individual affirmation cues are calculated by checking for the presence of each affirmation cue in each utterance of the 3066 UMT training dataset. At this iteration, the affirmation cue producing the highest value of $\Delta F1\,score$ = 65.4% is *got it*. Therefore, *got it* is

Table 4. Iteration-wise progression of affirmation cues

Iteration Number or Operating Point	Rule-set for lexical-only classifier
0	NULL
1	got it
2	got it, yep
3	got it, yep, sounds good
4	got it, yep, sounds good, done
5	got it, yep, sounds good, done, got you
6	got it, yep, sounds good, done, got you, awesome
7	**got it, yep, sounds good, done, got you, awesome, gotcha**
8	got it, yep, sounds good, done, got you, awesome, gotcha, sweet
9	got it, yep, sounds good, done, got you, awesome, gotcha, sweet, great

chosen as our base affirmation cue in the progressive rule-set and represented as a point on the F1 score line at Iteration 1 or operating point 1 in Fig. 8.

Iteration 2: the affirmation cue *got it* from the first iteration is individually combined with the remaining affirmation cues, two at a time, to calculate the combined F1 score. At this iteration, the affirmation cue producing the highest value of $\Delta F1\ score$ = 1.6 is *{got it, yep}*, represented as the F1 score = 67.0% on Iteration 2 or operating point 2 in Fig. 8.

This process of iteration-wise addition of affirmation cues to the existing set of affirmation cues contributes to our definition of a "Progressive rule-set" design. This progressive development of affirmation cues continues for 9 iterations (for 9 affirmation cues), as shown by a monotonically increasing F1 score line graph in Fig. 8. However, on close observation we find that the trend decreases slightly after the F1 score of 70.2% corresponding to iteration number 7. Hence 70.2% is the maximum F1 score. The corresponding sequences are the optimum sequences and represented in bold in Table 4.

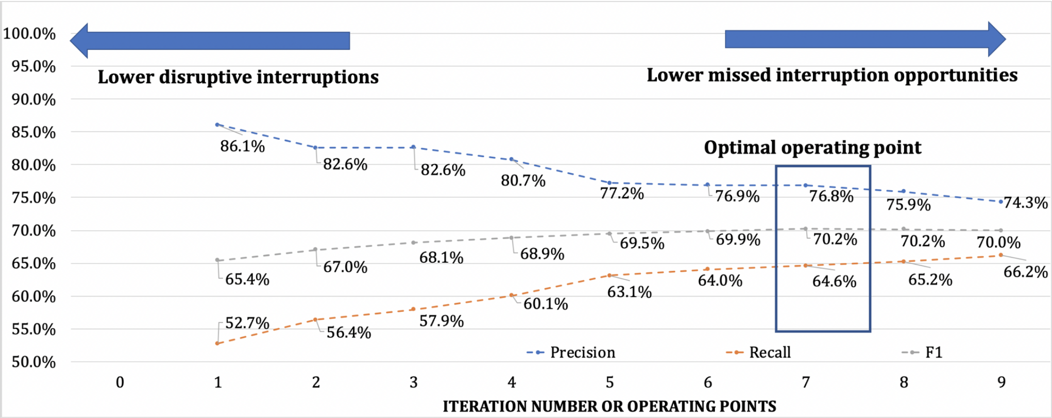
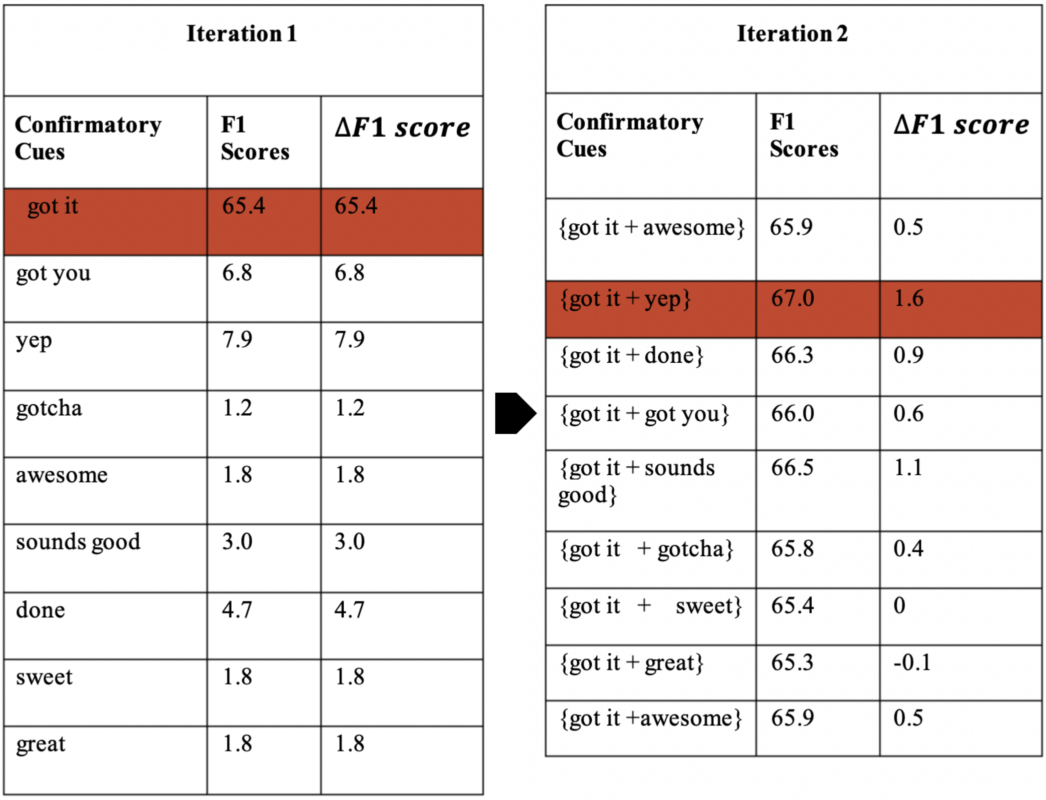


Fig. 8. Iteration-wise graphical representation of metrics of affirmation cues in steepest ascent search algorithm



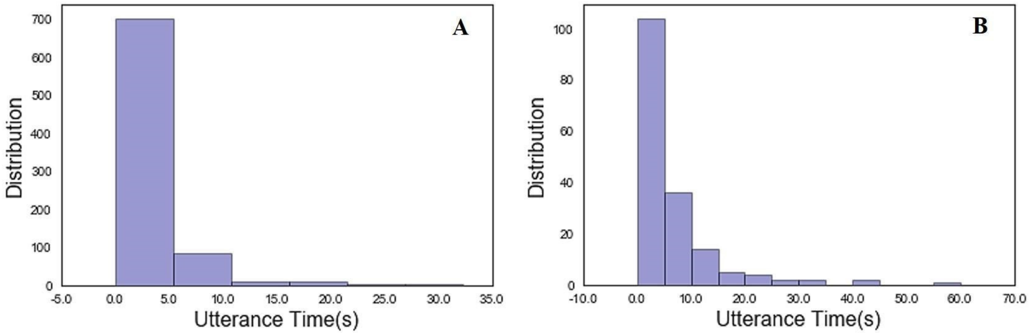
Iteration 1

Confirmatory Cues	F1 Scores	ΔF1 score
got it	65.4	65.4
got you	6.8	6.8
yep	7.9	7.9
gotcha	1.2	1.2
awesome	1.8	1.8
sounds good	3.0	3.0
done	4.7	4.7
sweet	1.8	1.8
great	1.8	1.8

Iteration 2

Confirmatory Cues	F1 Scores	ΔF1 score
{got it + awesome}	65.9	0.5
{got it + yep}	67.0	1.6
{got it + done}	66.3	0.9
{got it + got you}	66.0	0.6
{got it + sounds good}	66.5	1.1
{got it + gotcha}	65.8	0.4
{got it + sweet}	65.4	0
{got it + great}	65.3	-0.1
{got it +awesome}	65.9	0.5

Fig. 9. First 2 of the 9 iterations of steepest ascent search algorithm



(a) UMT task boundary frequency distribution

(b) UMT False positive frequency distribution

Fig. 10. Frequency Distribution

4.4.3 Utterance Duration as an Extra Feature. To further mitigate the interference from backchannels, we also examine the duration of an utterance as an extra rule to help reduce false positives. The distribution of duration of the task boundary utterance is shown in Fig. 10a Here 97.5% of task boundary utterances have a duration less than 10 seconds, which leads to a 2.5%

extra missed interruptions detection if we impose a threshold of 10s for task boundary utterances. However, when we examine the distribution of the duration for non–task boundary utterances that are false positives, as shown in Fig. 10b, it becomes clear that 17.7% of false positives are above the 10 second limit and will be filtered out of the identified task boundary if we impose a threshold of 10s for task boundary utterances. Thus, we expect a portion of these false positive can be removed at a low cost of missed interruption opportunities.

Therefore, we also look at the effect of adding one extra rule "DURATION is less than or equal to 10 seconds" into the rule-set.

4.4.4 Implications of the Progressive Rule–Set Design. It is evident that systematic selection of lexical affirmation cues is necessary to construct an accurate, scalable and adaptable rule–based classifier for interruption dissemination. The steepest ascent search algorithm with F1 score as objective function facilitates this systematic selection of optimal progressive rule set. In this subsection, let us examine how the progressive rule-set based classifier addresses the challenges of optimality, complexity, adaptability and real-time operation.

(1) *Solution to optimal rule-set*: By utilizing F1 score as the objective function, the steepest ascent search algorithm produces a steepest ascending curve that peaks at the maximum F1 score for the rule-set at each iteration before it starts descending as shown Fig. 8. As a result, the rule-set of affirmation cues sequenced until the maximum F1 score can be considered as a series of incremental optimal rule-sets.
(2) *Solution to complexity [34]*: The search process of the steepest search algorithm for $n = 9$ affirmation cues performs n affirmation cue F1 score calculations in the first iteration, *(n−1)* calculations in the second iteration, *(n−2)* in the third and so on. Hence, it can conclude that the search process has a quadratic complexity of $O(n^2)$, which has less complexity when compared to the exponential complexity of $O(2^n)$ when using a brute–force approach, to find all rule-set combination of all size. This difference in time complexity allows us the rule-set to operate relatively faster with more affirmation cues.
(3) *Solution to adaptability*: In Fig. 11 we also label the X-axis as operating point. This is done to emphasize operating point dependent behavior of the classifier. By choosing operating point A we favor lower disruptive interruptions over missed interruption opportunities and vice-versa by choosing operating point B. Hence, we can tune the classifier to prioritize the needs of interruption dissemination for the application.

Thus, we have demonstrated that the steepest search algorithm can be utilized to generate a progressive rule set that facilitates real-time operation. It is optimal, scalable and adaptable to application specific requirements (Missed interruptions vs False interruptions). This rule–set should enable the classifier to distinguish between task–boundaries and non–task boundaries according to the selected operating point. To evaluate its performance against task–oriented dialogues we proceed to Section 5 where we present the experimentation design and results.

5 RESULTS

In this section we present the experimental results of the proposed ACE-IMS. We first present a brief description summarizing our experiments, then analyze the generalizability of the progressive rule-set based classifier by comparing the results between the Training and testing datasets. We also provide a performance comparison with the real-time C-CIMS, current baseline ACE-IMS.

As described in Section 4.1, we have used two test datasets for the performance evaluation: UMT test dataset and Tangram test dataset. The UMT test dataset allows us to test if the classifier performance generalizes to other utterances from the same task, while the Tangram dataset allows us to assess classifier performance on task-oriented dialogue of a different task. To compute the classifier metrics of Precision, Recall and F1 score, the ACE-IMS assigned labels of Task boundary versus Non − task boundary are compared with the manual annotations of task boundary information, Task boundary versus Non − task boundary based on corresponding log files of the two datasets, UMT and Tangram, described in Section 4.1.

Table 5 summarizes the classification results for the ACE-IMS for the UMT training dataset, UMT test dataset and Tangram test dataset. We present the results for both lexical-only classifier and lexical–Duration classifier, with the latter exploring additional potential to suppress false interruptions due to back–channels as discussed in Section 4.4.3.

5.1 Inter Dataset Performance Evaluation

Firstly, let us evaluate the performance of the ACE-IMS by considering the lexical-only classifier across UMT test and Tangram test datasets. Results in Table 5 show that ACE-IMS perform robustly across both UMT test dataset and Tangram dataset. It achieves 68.8% F1 score for UMT test dataset, which is close to its performance of 70.2% for the UMT training dataset. It proves that ACE-IMS generalizes well for the same type of task. Furthermore, ACE-IMS achieves 91.1% F1 score for the Tangram test dataset, this validates the earlier Coverage of affirmation cues in the task-boundary

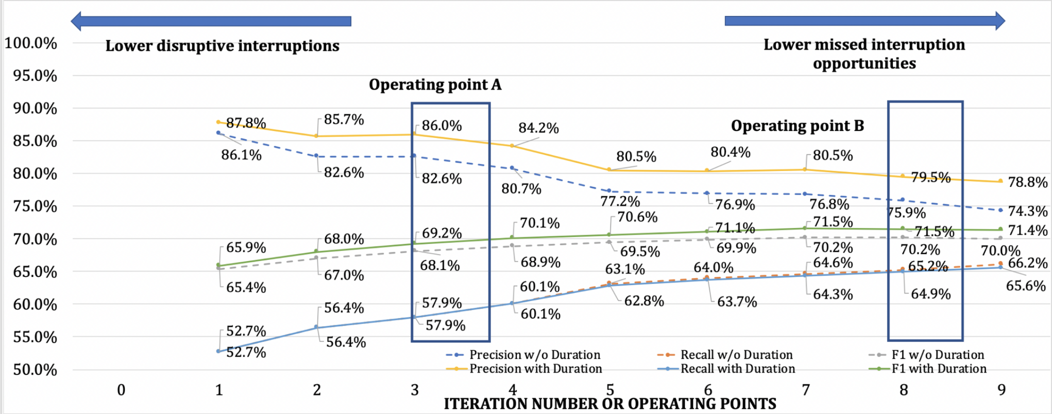


Fig. 11. Operating point-based adaptability of the Progressive Rule–set

Table 5. Classification results of training versus test datasets

Tasks	Features	Precision (%)	Recall (%)	F1 Score (%)	Data Split (Non-task boundary, task boundary)
UMT (Training)	Lexical	76.8	64.6	70.2	(2737, 329)
	Lexical + Duration	80.5	64.3	71.5	(2737, 329)
UMT (Test)	Lexical	76.8	62.4	68.8	(5162, 808)
	Lexical + Duration	80.2	60.2	68.9	(5162, 808)
Tangram (Test)	Lexical	94.6	87.9	91.1	(3396, 1158)
	Lexical + Duration	95.5	87.2	91.1	(3396, 1158)

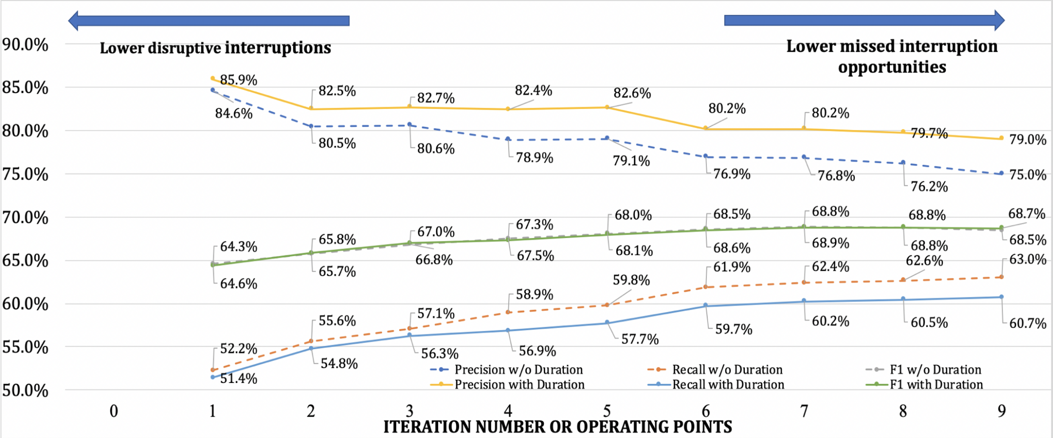


Fig. 12. Performance and Operation Adaptability of ACE-IMS (UMT Test Dataset)

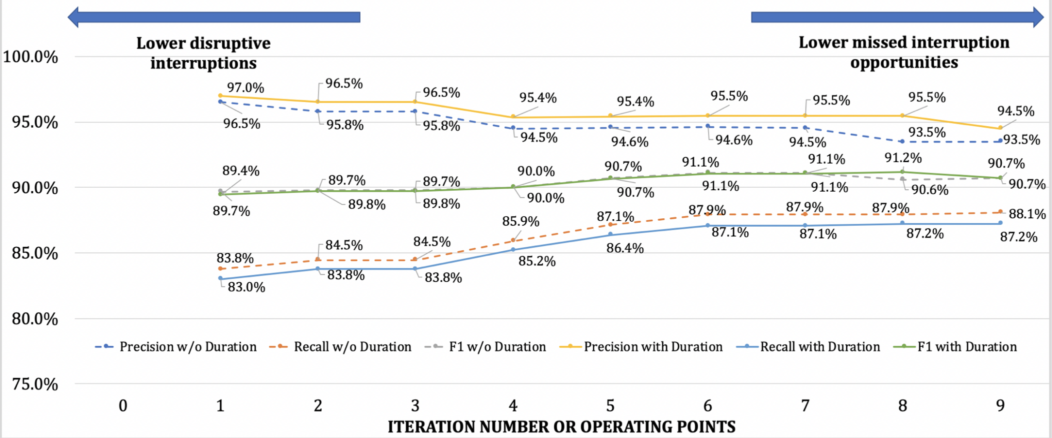


Fig. 13. Performance and Operation Adaptability of ACE-IMS (Tangram Test Dataset)

utterances of Tangram tasks. It shows that ACE-IMS generalizes well for a different type of task. Moreover, it demonstrates that ACE-IMS can take full advantage of the higher rate of affirmation cue usages in the Tangram tasks.

5.2 Inter Classifier Performance Evaluation

Next, we evaluate the inter classifier performance between lexical-only and lexical–Duration classifiers. Both classifiers achieve comparable optimal F1 scores on all three datasets, refer Table 5, with only marginal loss of Recall (which means it misses interruptions but to a lesser degree) which validate our approach, motivated by the observations in Fig. 10 in Section 4.4.3. Furthermore, on closer examination we find that this lexical-Duration classifier improves the precision score across all three datasets when compared to its lexical-only counterpart. Although not to a greater extent, the Duration feature helps reduce the number of false interruptions. But, more importantly, by adding Duration feature, it achieves much more gracefully descending trend of the Precision lines as shown in Figs. 11, 12 and 13. This is of importance when reducing false positives (which lead to disruptive interruptions) is of high priority.

Table 6. Real-time IMS results comparison with Real-time C-CIMS [31]

Tasks	**Features**	**Precision (%)**	**Recall (%)**	**F1 Score (%)**	**Data Split (Non-task boundary, task boundary)**
Tangram	Real-time ACE-IMS (proposed solution)	94.6	687.9	91.1	(3396, 1158)
	Real-time prosodic (C-CIMS)	79.1	70.8	74.7	(1205, 811)
UMT	Real-time ACE-IMS (proposed solution)	76.8	62.4	68.8	(5162, 808)
	Real-time prosodic (C-CIMS)	44.7	75.6	56.2	(3517, 961)

5.3 Real–time ACE-IMS vs Real-time C-CIMS [31]

Since one of the primary focus of this research work is to study the role of lexical affirmation cues in identifying task boundaries and compare its performance against existing literature, our focus, in this sub-section is limited to compare the performance of the real-time lexical-only classifier against the real-time C-CIMS implementation. Table 6 summarizes the performance results.

The performance results in Table 6 clearly demonstrate that the proposed Lexical based ACE-IMS classifier outperforms the real-time C-CIMS implementation. The ACE-IMS shows improvements in F1 score against the C-CIMS for both Tangram test dataset and UMT test dataset, 16.4% and 12.6%, respectively. For the Tangram dataset, the proposed ACE-IMS shows an improvement of 15.5% for Precision which means it disseminates less disruptive interruptions, while, at the same time, achieves 17.1% improvement for Recall, i.e., missing less opportunities to interrupt. For the UMT test dataset, the proposed ACE-IMS shows an improvement in precision by 32.1% when compared to C-CIMS. Although C-CIMS shows a better Recall by 13.2% for UMT test dataset, it is largely due to its imbalanced treatment between Precision and Recall, which results in a loss of 12.6% in F1 score when compared to ACE-IMS.

The numerical results suggest that the proposed IMS generalizes well across the UMT and Tangram datasets and outperforms the existing real-time implementation of C-CIMS in identifying task-boundaries.

6 DISCUSSION

In this research, we propose a real-time Affirmation Cues based Interruption Management System (ACE-IMS) that utilizes affirmation cues used in human conversations to address the problem of disseminating well-timed interruptions. To accomplish this, we adopt the latest real-time automatic speech recognition (ASR) capabilities offered through cloud service to obtain lexical information of task–oriented dialogues, and exploit the simplicity, scalability and adaptiveness of the progressive rule-set based classifier design. The experimental results obtained clearly show the generalizability and improvement achieved by our proposed system. To understand these results from a wider perspective we utilize this section to describe our results within the context of our three research

Table 7. Tabular Coverage Results of Fig. 4

Dataset	Total number of Task Boundary Utterances With The Identified Affirmation Cues (k)	Total Number of Task Boundary Utterances (N)	Coverage (k/N) (%)
UMT Training	230	329	69.9
UMT Test	508	808	62.9
Tangram Test	1020	1158	88.1

questions presented in Section 3.1. We describe its implications on this line of research and provide recommendations for improvements in potential future work.

6.1 Contributions

6.1.1 Affirmation cues indicative of Task boundaries . The first research question *"RQ1: If affirmation cues signal task transitions [16], what is the extent of their occurrence prior to a task boundary in a task-oriented dialogue?"* focusses on the relationship between affirmation cues and task boundaries. From Table 7 we infer that the identified lexical affirmation cues accounts for 69.9%, 62.9% and 88.1% of the set of manually annotated task boundaries for UMT training, UMT test and Tangram test dataset, respectively, which is also shown earlier in Fig. 4. The said Coverage data for the affirmation cues suggests that the identified affirmation cues account for majority of the task boundary utterances and can contribute to intelligent interruption dissemination, provided that the task structure of task–oriented dialogues are similar to UMT and Tangram datasets.

6.1.2 Proposed ACE-IMS Inference of Task–boundaries . From Table 5 it can be noted that the Recall or Coverage score is 64.6% for UMT training dataset, only 5.3% under the manually annotated Coverage measure of 69.9% as shown in Table 7. Similarly, the recall scores on the UMT test and Tangram test datasets are 62.4% and 87.9%, respectively, and are only 0.5% and 0.2% under the corresponding manually annotated Coverage scores of 62.9% and 88.1%, respectively. These results clearly demonstrate that the proposed ACE-IMS classifier can identify the affirmation cues with high probability while effectively suppressing false interruptions caused by interference from backchannels. Concerning real-time operation, the rule-based classifier, augmented by the lexical information flow from the automatic speech recognition system, identifies these task boundaries instantaneously when a user speaks through a microphone, as demonstrated through the Android prototype described in Section 4.3. This further validates the real-time interruption dissemination capability of the ACE-IMS. Within a multi-user multi-tasking context, the proposed ACE-IMS system plays the role of an information controller. It helps close the gap between the natural human interruptions and the artificial machine-generated interruptions. Thereby, minimizing the disruptiveness of an interruption. These insights and features answer our second research question, *"RQ2: Can a real-time system be built to explore the lexical affirmation cues and identify task boundaries as potential points of interruption in distributed multi-user, multi-tasking environments?"* in the positive.

6.1.3 Proposed IMS as the new baseline for real-time task boundary identification: Finally, the performance improvement in F1 score of 16.4% for Tangram dataset and 12.6% for UMT dataset of the real − time IMS system against real-time C-CIMS implementation clearly demonstrates the proposed real-time ACE-IMS improves on the real-time C-CIMS. Since, ACE-IMS utilizes the lexical (what is said?) feature against a pure prosodic (how it is said?) information of speech and demonstrated performance improvement, we can infer that the lexical feature's contribution to task boundary identification is more substantial than prosodic feature. This successfully establishes the proposed ACE-IMS as the new baseline for real-time task boundary identification and answers our final research question, *"RQ3: Does the proposed real-time IMS outperform the baseline real-time C-CIMS [31] in accurately detecting task boundaries as candidate interruption times?"*.

Therefore, these results clearly demonstrate a.) The identified lexical features based on human affirmation cues can be leveraged to effectively control and disseminate interruptions; b.) The identified lexical features improve the performance of the proposed classifier, when compared to other related work in the domain.

6.2 Future Work

Motivated by human beings' usage of affirmation cues to signal task completion or task transition, the proposed ACE-IMS infers the task–interruption candidate points, i.e., task boundary utterances, using lexical affirmation cues present in task-oriented dialogues, and creates a new baseline for real-time IMS. There are new opportunities emerging from this research that can be explored to further improve the IMS. We would like to present this from two perspectives:

(1) *Data Analysis*: The main aim of this research article was to explore the use of affirmation cues in real-time task–boundary identification. From Fig. 4 it was evident that most of the task boundaries are covered by the lexical affirmation cues as listed in Table 3. Hence, it would be of interest to analyze the remaining task–boundary utterance, i.e., those without the identified affirmation cues. One direction is to add prosodic features as described in [31] with the identified lexical affirmation cue features into a combined solution. The challenge could be "How to combine prosodic and lexical features to enable real-time operation?". Additionally, in this work, the task–oriented dialogues were analyzed on an utterance basis, this analysis could be expanded to include the entire task conversation, i.e., analyze the entire discourse of the dialogue to explore the turn–taking characteristics and discourse structure. This will lead to a more comprehensive understanding of the grounding mechanism [28], and in turn help identify task-boundaries without the explicit presence of affirmation cues in the utterance.

(2) *System Design*: To accommodate the additional features mentioned above, other statistical classification or deep learning methods can be exploited. For instance, the author of C-CIMS [31] did use algorithms like maximum entropy classifiers, and random forests for lexical only task boundary modeling, but stop short to establish a real-time operation for the system. Hence, investigating new approach and implementation and understanding the performance and constraints of these approaches in the future could be a new direction.

7 CONCLUSION

Ill-timed interruptions disrupt the ongoing task and degrade human productivity and affective state, which poses a serious threat, especially in distributed multi-user multi–tasking interactions. To address this challenge, this paper proposed ACE-IMS, an adaptive and real-time interruption management system that utilizes typical affirmation cues present in human-human communication to construct a highly effective and efficient solution for disseminating interruptions at task-boundary. This work, motivated by the observation that affirmation cues are present in majority of task-boundary utterances in task–oriented dialogues, demonstrated that affirmation cues can be leveraged in a real-time IMS to effectively control and disseminate interruptions in order to mitigate the challenges imposed by disruptive interruptions. The proposed solution has the potential to improve the effort in growing field of computer-supported distributed collaborations that strive to augment human cognitive capability with computational and sensory power of machines, while suppressing the risk of cognitive overload. Thus, the resulting cooperative teaming between humans and machines paves way for a new wave of human-machine teaming applications in stressful distributed multi-tasking environments, such as: emergency and disaster management, law enforcement, telemedicine, and other field operations.

ACKNOWLEDGMENTS

This work was accomplished with support from the 711th Human Performance Wing at the US Air Force Research Laboratory under contract number FA8650-14-D-6501.

REFERENCES

[1] 2019. Cloud Speech-to-Text - Speech Recognition | Cloud Speech-to-Text. https://cloud.google.com/speech-to-text/

[2] 2019. Developers | Audacity ®. https://www.audacityteam.org/community/developers/

[3] 2019. Steepest ascent method for multivariate optimization - Application Center. https://www.maplesoft.com/applications/view.aspx?SID=4194&view=html

[4] Christopher A. Monk, Deborah A. Boehm-Davis, and J. Gregory Trafton. 2002. The Attentional Costs of Interrupting Task Performance at Various Stages. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (2002).

[5] Piotr D. Adamczyk and Brian P. Bailey. 2004. If Not Now, When? The Effects of Interruption at Different Moments Within Task Execution. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Vienna, Austria, 271–278.

[6] Piotr D. Adamczyk, Shamsi T. Iqbal, and Brian P. Bailey. 2005. A method, system, and tools for intelligent interruption management. In *TAMODIA '05 Proceedings of the 4th international workshop on Task models and diagrams*. ACM, Gdansk, Poland, 123 – 126.

[7] Erik M. Altmann and J. Gregory Trafton. 2004. Task interruption: Resumption lag and the role of cues.. In *Annual Meeting of the Cognitive Science Society*, Vol. 26.

[8] Ernesto Arroyo and Ted Selker. 2011. Attention and Intention Goals Can Mediate Disruption in Human-Computer Interaction. In *INTERACT 2011: Human-Computer Interaction – INTERACT 2011*, Vol. 6947. Springer, Berlin" 454–470.

[9] Brian P. Bailey and Shamsi T. Iqbal. 2008. Understanding changes in mental workload during execution of goal-directed tasks and its application for interruption management. *ACM Transactions on Computer-Human Interaction* (2008), 1–28.

[10] Brian P Bailey and Joseph A Konstan. 2006. On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state. *Computers in Human Behavior* 22, 4 (July 2006), 685 – 708.

[11] Edward Cutrell, Eric Horvitz, and Mary Czerwinski. 2001. Notification, disruption, and memory: Effects of messaging interruptions on memory and performance., Vol. 1. 263.

[12] Mary Czerwinski and Eric Horvitz. 2000. Instant Messaging and Interruption: Influence of Task Type on Performance. In *OZHI 2000 conference proceedings*, Vol. 356. 361–367.

[13] Mary Czerwinski and Eric Horvitz. 2000. Instant messaging: Effects of relevance and timing.. In *People and computers XIV: Proceedings of HCI*, Vol. 2. 71–76.

[14] Laura Dabbish and Robert E. Kraut. 2004. Controlling interruptions: awareness displays and social motivation for coordination.. In *In Proceedings of the 2004 ACM conference on Computer supported cooperative work*. 182–191.

[15] Ashley Edwards, Leslie-Anne Fitzpatrick, Sara Augustine, Alex Trzebucki, Shing Lai Cheng, Candice Presseau, Cynthia Mersmann, Bruce Heckman, and Stan Kachnowski. 2009. Synchronous communication facilitates interruptive workflow for attending physicians and nurses in clinical settings. 78, 9 (Sept. 2009), 629–637.

[16] Agustin Gravano, Julia Hirschberg, and Stefan Benus. 2012. Affirmative Cue Words in Task-Oriented Dialogue. *Computational Linguistics* 38, 1 (March 2012), 1 – 39.

[17] LindaMcGillis Hall, Cheryl Pedersen, Pam Hubley, Elana Ptack, Aislinn Hemingway, Carolyn Watson, and Margaret Keatings. 2010. Interruptions and Pediatric Patient Safety. *Journal of Pediatric Nursing* 25, 3 (June 2010), 167 – 175.

[18] Peter Heeman. 1993. Speech actions and mental states in task-oriented dialogues.. In *Spring Symposium on Reasoning About Mental States: Formal Theories and Applications*.

[19] Shamsi T. Iqbal and Brian P. Bailey. 2005. Investigating the effectiveness of mental workload as a predictor of opportune moments for interruption. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*. Portland, OR, USA, 1489 – 1492.

[20] Shamsi T. Iqbal and Brian P. Bailey. 2006. Leveraging characteristics of task structure to predict the cost of interruption. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Montreal, Quebec, Canada, 741 – 750.

[21] J. G. Kreifeldt and M. E. Mccarthy. 1981. Interruption as a test of the user-computer interface. In *Proceedings of the Seventeenth Annual Conference on Manual Control*. United States, 655–667.

[22] Jari Laarni, Hannu Karvonen, Satu Pakarinen, and Jari Torniainen. 2016. Multitasking and Interruption Management in Control Room Operator Work During Simulated Accidents. In *Engineering Psychology and Cognitive Ergonomics*, Don Harris (Ed.). Springer International Publishing, 301–310.

[23] K. A. Latorella. 1996. Investigating Interruptions: An Example from the Flightdeck. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 40, 4 (1996).

[24] D. Scott McCrickard, Christa M. Chewar, Jacob P. Somervell, and Ali Ndiwalana. 2003. A model for notification systems evaluation—assessing user goals for multitasking activity. *ACM Transactions on Computer-Human Interaction (TOCHI) 10* 10, 4 (2003), 312–338.

[25] Daniel C. McFarlane. 1997. *Interruption of people in human-computer interaction: A general unifying definition of human interruption and taxonomy.* Ph.D. Dissertation. Cornell University, Office of Naval Research Arlington, VA 22217-5660.

[26] Daniel C. McFarlane. 1999. Coordinating the Interruption of People in Human-Computer Interaction. In *Human-computer interaction, INTERACT*, Vol. 99. IOS Press, 295–303.

[27] Daniel C. McFarlane and K. A. Latorella. 2002. The scope and importance of human interruption in human-computer interaction design. *Human-Computer Interaction* 17, 1 (2002).

[28] Brian Paltridge. 2012. *Discourse Analysis: An Introduction (Bloomsbury Discourse)* (2 ed.). Continuum.

[29] Nia Peters, Griffin Romigh, George Bradley, and Bhiksha Raj. 2017. When to Interrupt: A Comparative Analysis of Interruption Timings Within Collaborative Communication Tasks, Vol. 497. Springer, 177 – 187.

[30] Nia Peters, Griffin Romigh, George Bradley, and Bhiksha Raj. 2018. A Comparative Analysis of Human-Mediated and System-Mediated Interruptions for Multi-user, Multitasking Interactions. In *Advances in Human Factors and Systems Interaction*, Isabel L. Nunes (Ed.). Springer International Publishing, 339–347.

[31] Nia S. Peters. 2017. *Collaborative Communication Interruption Management System (C-CIMS): Modeling Interruption Timings via Prosodic and Topic Modelling for Human-Machine Teams.* Dissertations. Carnegie Mellon University, Pittsburgh, Pennsylvania.

[32] Joshua S. Rubinstein, David E. Meyer, and Jeffrey E. Evans. 2001. Executive control of cognitive processes in task switching. *Journal of experimental psychology: human perception and performance* 4 (2001), 763.

[33] Jamie Sutherland. 2017. What is word error rate and who is winning? https://medium.com/@sutherlandjamie/what-is-word-error-rate-and-who-is-winning-e623db5d7913

[34] Arun C. Thomas. 2020. What is computational complexity? https://towardsdatascience.com/what-is-computational-complexity-66722cd5f8dd

[35] Fred RH Zijlstra, Robert A. Roe, Anna B Leonora, and Irene Krediet. 2010. Temporal factors in mental work: Effects of interrupted activities. *Journal of Occupational and Organizational Psychology* 72, 2 (2010), 163–185.

Received October 2019; revised January 2020; accepted March 2020