rrSDS: Towards a Robot-ready Spoken Dialogue System

Casey Kennington **Daniele Moro** **Lucas Marchand** **Jake Carns** **David McNeill**
Department of Computer Science
Boise State University
1910 W University Dr
Boise, ID 83725
firstnamelastname@boisestate.edu

Abstract

Spoken interaction with a physical robot requires a dialogue system that is modular, multimodal, distributive, incremental and temporally aligned. In this demo paper, we make significant contributions towards fulfilling these requirements by expanding upon the ReTiCo incremental framework. We outline the incremental and multimodal modules and how their computation can be distributed. We demonstrate the power and flexibility of our robot-ready spoken dialogue system to be integrated with almost any robot.

1 Introduction

Spoken Dialogue Systems (SDSs) are well-suited to handle complex artifacts of dialogue such as hesitations and clarification requests in many domains. However, to further extend SDSs to work effectively on physical robots, we offer the following additional requirements towards a *robot-ready* SDS: **modular**: robot components are modular and individual modules must be able to integrate with SDS modules, **multimodal**: robots are *situated* dialogue partners whose many sensors must be integrated with the SDS speech input, **distributive**: robot and SDS modules should easily communicate with each other in a distributed environment, **incremental**: modules must be able to process input quickly and immediately, **aligned**: sensors must be temporally aligned, i.e., synchronized in time.

Existing systems offer solutions to some of the requirements. The OpenDial toolkit gives researchers the ability to model dialogue states using probabilistic rules (Lison and Kennington, 2016), but any incrementality has not been systematically evaluated. InproTK (Baumann and Schlangen, 2012), is incremental and InproTK$_s$ (Kennington et al., 2014) added distributiveness and multimodality, and Kennington et al. (2017) offered an approach to temporal alignment, albeit with offline evaluation.

The PSI framework is inherently modular, multimodal, temporally aligned, has been evaluated on robot platforms, and has several options for distributing computation (Bohus et al., 2017). However, the PSI framework does not yet build on any incremental framework. Also similar to our work is the platform MultiBot presented in Marge et al. (2019), but that model does not work incrementally nor does it consider temporal alignment.

In this paper, we design and evaluate a modular, incremental, multimodal, and distributive *robot-ready* SDS, called $_{rr}$SDS which is primarily written in the Python programming language.[1] To address the requirements of *modularity* and *incrementality*, we adopt the Incremental Unit Framework (Schlangen and Skantze, 2011) where *incremental units* (IUs) are passed between modules (IUs can be *added* to reflect new information, or *revoked* if a previously added IU needs to be updated) by building on the *ReTiCo* (Michael and Möller, 2019) platform. To handle *distributiveness*, $_{rr}$SDS has modules (i.e., *ZeroMQ*, *ROS*) that afford interopability with processes outside of the system. To address the requirement of *multimodality*, we build on top of the modularity requirement and incorporate additional sensors (e.g., cameras and internal robot states).

2 The $_{rr}$SDS Spoken Dialogue System

ReTiCo has existing modules for built-in microphones and Google Speech API for speech recognition. We extend it to be multimodal by adding additional sensor modules such as cameras and internal robot states, depicted in Figure 1. We add distributive modules that handle interopability with

[1] Available at https://github.com/bsu-slim/rr-sds

Proceedings of the SIGdial 2020 Conference, pages 132–135
1st virtual meeting, 01-03 July 2020.

outside modules. The rest of this section explains the modules for $_{rr}$SDS.

Dialogue Management OpenDial is a toolkit for developing SDSs with probabilistic rules (Lison and Kennington, 2016) that can be used as a rule-based dialogue manager (DM), but can be extend any domain to include stochastic processes when data is available. We incorporate a recent Python implementation called pyOpenDial (Jang et al., 2019) into our SDS as a DM. Our pyOpenDial module takes any IU payload, expecting a list of attributes (i.e., variables) and values that it adds to pyOpenDial's dialogue state as attribute/value pairs.

Natural Language Understanding *Words-as-Classifiers* (WAC) is a model of grounded semantics that learns a 'fitness' score between physical entities and words (Kennington and Schlangen, 2015), where each word in a known vocabulary is represented as a classifier. WAC is inherently incremental and can learn word groundings with minimal training data. This module takes words from ASR and features from detected objects. It outputs the best fit word for the detected object as well as confidence scores for all the words in its vocabulary and their fitness to all detected objects.

Object Detection & Feature Extraction This module uses Huang et al. (2017), which builds on several other advances in fast object detection. The output of this module is a list of bounding boxes for each object, along with corresponding labels and confidence scores. The feature extractor takes those object bounding boxes, isolates the bounded object from the rest of the image, and passes that single object image through a pre-trained imagenet model, for example, EfficientNet (Tan and Le, 2019) or InceptionV3,[2] but designers can specify any Keras network and target layer. This module outputs a list of vectors that represent each object that was found in the input image.

Seeed Respeaker The respeaker is an array microphone with 6 individual microphones on a disc-like board.[3] Respeaker has built-in functionality for direction-of-arrival, noise suppression, keyword wake up, and network connectivity.

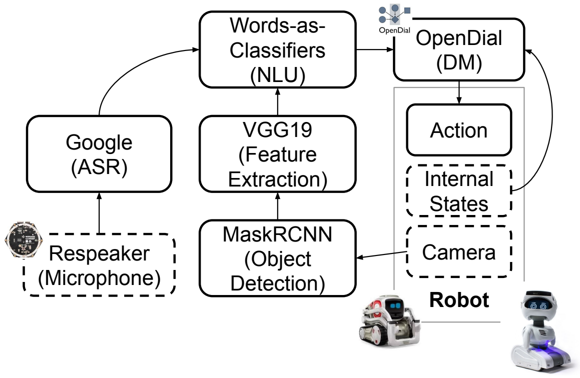

Figure 1: Overview of our $_{rr}$SDS integrated with the robot modules. Dashed lines represent sensory input modules.

Distributive Interop ZeroMQ is a universal message passing library that builds and maintains sockets that carry atomic messages across various transports.[4] ZeroMQ supports most programming languages and operating systems. The amount of code required to use ZeroMQ to pass messages between separate processes is very minimal. For our SDS, we have two types of ZeroMQ modules: `Readers` and `Writers`.

A key interopability module in $_{rr}$SDS is the Robotics Operating System (ROS), which is widely adopted in the robotics community.[5] ROS has a built-in communication layer across any robotic system's architecture that provides data pipelines on different scopes (Quigley et al., 2009). Our $_{rr}$SDS interfaces with ROS using Publish and Subscribe modules (similar to ZeroMQ's Writer and Reader modules). We evaluated our implementation using Turtlesim, a common test bed simulation for ROS.[6]

Additional Modules $_{rr}$SDS has additional modules that we do not use in our evaluation, but we do mention them for completeness: Azure Cognitive Services Speech Recognition (ASR), Azure Emotion Recognition API (takes in an image and returns a distribution over 8 emotional states), Azure Object Detector (similar to the MaskRCNN module above, this takes an image as input and returns a list of bounding boxes and corresponding labels), RASA (NLU) (Bocklisch et al., 2017) which has been evaluated to be competitive with commercial NLU platforms (Braun et al., 2017; Liu et al., 2019)

[2] This needs to match the vector representations that any grounded NLU module (e.g., WAC) was trained with.

[3] http://wiki.seeedstudio.com/ReSpeaker_Core_v2.0/

[4] https://zeromq.org/

[5] We note that our chosen interopability platforms are also available on PSI, which motivated our choices.

[6] http://wiki.ros.org/turtlesim

Figure 2: Misty in its task setting: Misty could move its head left and right, and had to look down at the objects on the table.

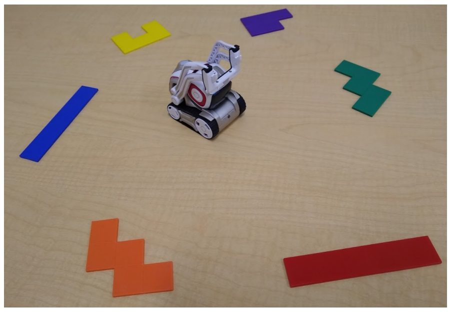

Figure 3: Cozmo in its task setting: in order for Cozmo to observe objects with it's camera, its head had to be pointed slightly down, and its lift had to be raised.

and has recently been incrementalized Rafla and Kennington (2019).

3 Evaluation

We used the Mistyrobotics Misty II and Anki Cozmo robot platforms to evaluate $_{rr}$SDS, depicted in Figures 2 and 3. We briefly explain the two platforms and the modules we built to integrate them into our $_{rr}$SDS, then we describe the evaluation.

Robot Modules Integration of Cozmo with $_{rr}$SDS is done using its Python SDK and Misty using its REST API, each broken into three ReTiCo modules: (1) camera, (2) internal state, and (3) action control. The output of each camera module is an IU with a still image as its payload. Both robots have internal state variables (e.g., `left-wheel-speed`, `head-height`, `light-height`). As the state of the robot changes, this module produces an IU containing a full attribute-value matrix of the internal state representation (e.g., wheel speed, lift height) at the state update. The action modules use the decisions made by the DM to produce the following actions: `explore`, `align`, `approach`, `confirm`, and `speak`.

A human user utters a short description and the robot attempts to explore its surroundings until it finds an object that matches the description. After a user description, the robot enters an `explore` state to seek out an object, then an `align` state to move the object into center view. Then the robot `confirms` if the description matches the object it is looking at. The robot `speaks`, uttering either *That looks X* or *Uhh that's not X that's Y* where X is the description and Y is the robot's best guess at a description (i.e., a better color word).

An overview of our $_{rr}$SDS is depicted in Figure 1. We use the Respeaker microphone, Google ASR, and WAC modules for spoken input, recognition, and understanding, respectively. Each robot's camera passed image frames to the MaskRCNN Object Detection module, then we used the VGG19 fc1 layer (4096 features; pre-trained on imagenet data) to represent objects for the WAC module. For dialogue and action management, we used the pyOpenDial module. For the WAC module, we used logistic regression classifiers pretrained on words that only focused on colors. We obtained the training data for WAC by capturing objects using Cozmo's camera; 5-10 training instances per color (trained using l2 normalization).

We recruited 15 participants from Boise State University (4 female, 11 male) to interact with each robot and fill out the Godspeed Questionnaire (Bartneck et al., 2009) after each interaction.

Our $_{rr}$SDS can run completely on a single machine;[7] output from all system processing modules were logged using PSI on a separate machine. We used the ZeroMQ modules to send information from $_{rr}$SDS to PSI.

We found in our evaluation that participants were able to accomplish the same number of tasks with both robots, but generally found Cozmo interesting, likeable and pleasant whereas Misty was judged as more mechanistic, rigid, stagnant and machine like.

4 Conclusions & Future Work

Our $_{rr}$SDS is flexible, being evaluated on multiple robot platforms to create engaging human-robot interactions, and fulfills the modular, incremental, multimoal, and distributive requirements for a robot-ready SDS. Our evaluation of $_{rr}$SDS allowed users to successfully interact with two different

[7]Our Machine had 32GB of RAM and an NVidia Tesla M40 with 12GB of Video RAM.

robots to accomplish a simple task with comparable performance. ${}_{rr}$SDS is agnostic to the robot platform used, enabling future research to experiment with robot platforms using our flexible system. For future work, we plan to add natural language generation modules and integrate ${}_{rr}$SDS more directly with PSI to make use of its architecture, thereby allowing developers and researchers to make use of PSI temporal alignment functionality, but spend most of their development time with Python.

Acknowledgements We thank the anonymous reviewers for their feedback and useful insights. We thank Microsoft Research for answering questions related to PSI. This work was approved under the Boise State University IRB #126-SB20-012.

References

Christoph Bartneck, Dana Kulić, Elizabeth Croft, and Susana Zoghbi. 2009. Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *International journal of social robotics*, 1(1):71–81.

Timo Baumann and David Schlangen. 2012. The InproTK 2012 release. In *NAACL-HLT Workshop on Future directions and needs in the Spoken Dialog Community: Tools and Data (SDCTD 2012)*, pages 29–32.

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open Source Language Understanding and Dialogue Management. *Proceedings of the 31st Conference on Neural Information Processing Systems*.

Dan Bohus, Sean Andrist, and Mihai Jalobeanu. 2017. Rapid Development of Multimodal Interactive Systems: A Demonstration of Platform for Situated Intelligence. In *Proceedings of ICMI*, Glasgow, UK. ACM.

Daniel Braun, Adrian Hernandez Mendez, Florian Matthes, and Manfred Langen. 2017. Evaluating Natural Language Understanding Services for Conversational Question Answering Systems. In *Proceedings of the SIGDIAL 2017 Conference*, pages 174–185.

Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. 2017. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-Janua, pages 3296–3305.

Youngsoo Jang, Jongmin Lee, Jaeyoung Park, Kyeng-Hun Lee, Pierre Lison, and Kee-Eung Kim. 2019. PyOpenDial: A Python-based Domain-Independent Toolkit for Developing Spoken Dialogue Systems with Probabilistic Rules. In *Proceedings of EMNLP*.

Casey Kennington, Ting Han, and David Schlangen. 2017. Temporal Alignment Using the Incremental Unit Framework. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, ICMI 2017, pages 297–301, New York, NY, USA. ACM.

Casey Kennington, Spyros Kousidis, and David Schlangen. 2014. InproTKs: A Toolkit for Incremental Situated Processing. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 84–88, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Casey Kennington and David Schlangen. 2015. Simple learning and compositional application of perceptually groundedword meanings for incremental reference resolution. In *Proceedings of ACL-IJCNLP 2015*, volume 1.

Pierre Lison and Casey Kennington. 2016. OpenDial: A toolkit for developing spoken dialogue systems with probabilistic rules. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - System Demonstrations*.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019. Benchmarking Natural Language Understanding Services for building Conversational Agents.

Matthew Marge, Stephen Nogar, Cory J Hayes, Stephanie M Lukin, Jesse Bloecker, Eric Holder, and Clare Voss. 2019. A Research Platform for Multi-Robot Dialogue with Humans. Technical report.

Thilo Michael and Sebastian Möller. 2019. ReTiCo: An open-source framework for modeling real-time conversations in spoken dialogue systems. In *Tagungsband der 30. Konferenz Elektronische Sprachsignalverarbeitung 2019*, ESSV, pages 134–140, Dresden. TUDpress, Dresden.

Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. 2009. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*.

Andrew Rafla and Casey Kennington. 2019. Incrementalizing RASA's Open-Source Natural Language Understanding Pipeline. *arXiv*.

David Schlangen and Gabriel Skantze. 2011. A General, Abstract Model of Incremental Dialogue Processing. In *Dialogue & Discourse*, volume 2, pages 83–111.

Mingxing Tan and Quoc V. Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv*.