Interactional Slingshots: Providing Support Structure to User Interactions in Hybrid Intelligence Systems

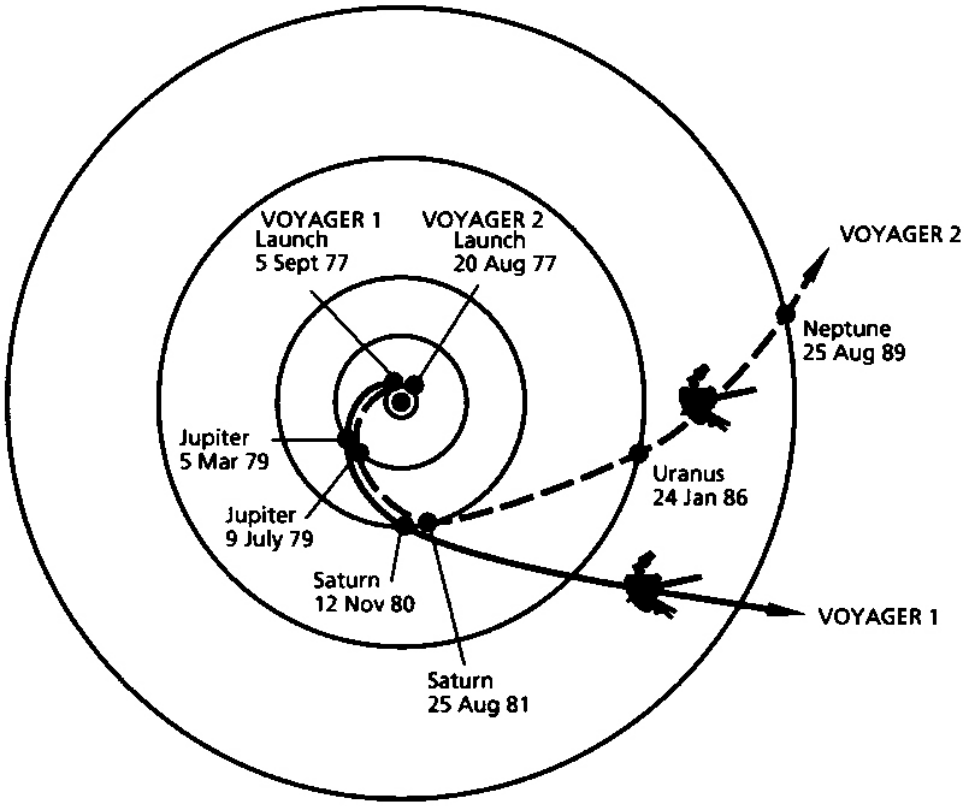
by

Sai Rohit Gouravajhala

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
2020

Doctoral Committee:

Professor Mark S. Ackerman, Chair
Professor Joyce Chai
Professor Mark J. Guzdial
Dr. Jonathan K. Kummerfeld



Voyager 1 and Voyager 2 spacecraft relying on gravitational slingshots from the gas planets to escape the solar system.
NASA / JPL / Public Domain

Sai Rohit Gouravajhala

sairohit@umich.edu

ORCID iD: 0000-0002-8686-0567

In loving memory of my grandfather, Ganti Suryanarayana, who left us way too soon, but not before giving us an enduring love of pursuing knowledge while snacking on aratikaaya bajjis and pesarattu.

Acknowledgments

My graduate school experience was like a roller coaster ride, filled with innumerable ups and downs, jaw-dropping twists and turns, G-force-inducing loop-de-loops, backward motions, and thrilling speeds—all mixed in with some truly spectacular views. That I was able to stay tethered to the coaster's rails throughout this journey is a testament to the help and support I received from the following people. (To be honest, it would be impossible for me to list all the people that have supported me over these years, but I want to highlight the following.)

Thank you, first and foremost, to my resplendent committee—Mark Ackerman (chair), Joyce Chai, Mark Guzdial, and Jonathan Kummerfeld—without whose support my dissertation would literally not have been possible. Mark Ackerman's willingness to supervise the last leg of my dissertation work is something for which I will forever be thankful. Joyce Chai's feedback was helpful to contextualize my last piece of dissertation work. Jonathan Kummerfeld's ability to create a supportive environment let me put my best foot forward at every step. Mark Guzdial's feedback was crucial to better understanding the why behind annotator behavior. (Plus, co-teaching with Mark—a teaching inspiration—was easily one of the highlights of my career at Michigan!) Thank you all so much.

Thank you, as well, to my undergraduate professors and mentors at Villanova University—Kevin Buckley, Edward Char, Edward Guinan, Lunal Khuon, and Xiafang "Maggie" Wang—who sparked my interest in doing research and who encouraged me to apply to graduate school. In particular, I only wish everyone could experience the mentorship, encouragement, and guidance that Dr. Khuon and Dr. Guinan provided me, and to them both, I am so incredibly thankful. And thanks to Rich D'Antonio and Jimmy Tripiciano for being dear and constant friends since we started undergrad together.

I am so thankful for the Graduate Employees' Organization for fighting for grad students' rights and making sure that we get tuition, a stipend, and health care, among other benefits. Thank you to the English Language Institute's Conversation Circles program, which provided me some of my happiest memories and interactions as I got to learn about other students' cultures. Thank you to the Rackham Graduate School for being such a supportive organization for graduate students: from funding conference travel to research projects to hosting amazing workshops, you have made graduate school a pleasure for so many of us. Thank you!

Thank you to the U.S. Food and Drug Administration, IBM Research, and Microsoft Research for hosting me for summer research internships. In particular, I want to thank Justin Cranshaw, my MSR mentor, for being the most supportive, understanding, and wisdom-dispensing mentor that I have had the luck and pleasure of having.

My graduate program years would have been listless if not for the advice, conversations, and general interactions that so many professors gave me. Thank you: Bill Arthur, Nikola Banovic, Stephen Checkoway, Mosharaf Chowdhury, Drew DeOrio, Tawanna Dillahunt, Ron Dreslinski, Nicole Ellison, Peter Honeyman, Chad Jenkins, Amir Kamil, Manos Kapritsos, John Laird, Barzan Mozafari, Michael Nebeling, Brian Noble, Steve Oney, Seth Pettie, Atul Prakash, Emily Mower Provost, Alanson Sample, Sarita Schoenebeck, Tom Wenisch, Jenna Wiens, and Don Winsor. In particular, I owe a huge debt of gratitude to Matthew Hicks (then a postdoc, now a professor at Virginia Tech) for nurturing my skill set and helping me find my footing. Matt's infinite patience, paired with inimitable research acumen, created the best environments for budding researchers.

The administrative staff at Michigan makes the department go around, and I'm privileged to have received their support. The staff is the best at removing any bureaucratic obstacle and for making sure that my experience as a grad student was as smooth as possible. Thank you: Ashley Andreae, Yolonda Coleman, Kelly Cormier, Dawn Freysinger, Karen Liska, Dia Moulton, Sarah Pena, Brian Rice, Alexis Santa-Cruz, and Punam Vyas. In particular, thank you especially to Christa Carr, Jamie Goldsmith, Charlie Mattison, Dana Mickle, Stephanie O'Keefe, Steve Reger, and Anne Lee Rhoades for giving me oodles of candy, heads up about free food opportunities (sometimes even before the rest of the department found out), advice, and most importantly, friendship that made my time at Michigan more fulfilling.

Some of my fondest memories during grad school involve having spontaneous chats and adventures with people on campus, in Ann Arbor, at internships, conferences, and beyond. Thank you: Nilmini Abeyratne, Mahdi Aghadjani, Gabriel Aguilera, Zakaria Aldeneh, Zaina Hamid Anwar, Javad Bagherzadeh, Lindsay Blackwell, Priyank Chandra, Dongyao Chen, Pin-Yu Chen, Meghan Clark, Jake Czyz, Karthik Desingh, MeiXing Dong, Christine McClelland Erickson, Eric Failes, Arun Ganesan, Sam Gilson, Gabriel Grill, Shangzhen Han, Mo Hussein, Ashok Jangir, Vaishnav Kameswaran, Ram Srivatsa Kannan, Preeti Kaur, Deepak Kumar, Jason Lee, Cindy Lin, Cyn SY Liu, Brandon Locke, Lajanugen Logeswaran, Biruk Mammo, Naveen Narisetty, Jeeheh Oh, Shwetha Rajaram, Janarthanan Rajendran, Preeti Ramaraj, Rohit Ramesh, Sanae Rosen, Yuru Shao, Kevin Errol Shipley, Drew Springall, Akshitha Sriraman, Bryan Stearns, Lan Triệu (hi Quân and Vivi!), Tim Trippel, Yu-Chih Tung, Sukrit Venkatagiri, Alyssa Woo, Shichang Xu, and Hongting Zhu. Thanks especially to Matt Bernhard, Allison McDonald, and Ben VanderSloot for being

partners-in-crime for discussing everything from the state of the department to the state of the world to just being there whenever I needed pick-me-ups. A special shoutout to Salessawi Ferede for being the most even-keeled calmer-of-nerves friend that I could count on during tough times.

What would my graduate school experience have been like if not for the company of those with whom I shared offices and labs? Thank you: Jeremy Erickson, Jaylin Herskovitz, Youxuan Lucy Jiang, Harman Kaur, Sang Won Lee, Anthony Liu, Ashkan Nikravesh, Amir Rahmati, Michael Rushanan, Jean Song, Joel Van Der Woude, and Zhefan Ye. Thanks especially to Yan Chen, Earlence Fernandes, Jordan Huffaker, Essam Idris Khan, Rebecca Krosnick, Aravind Vadrevu, and Jinyeong Yim for being trustworthy confidants, support beams, and the loveliest of friends. A special shoutout to my friend Divya Ramesh for being a wonderful companion: through our summer internship and my stressful semesters, you have given me incredible support. Though I wish we had gotten to collaborate on projects and spend more time in the lab, I am still so glad you came to Michigan and that I can call you a dear friend!

My life in Ann Arbor (and beyond) would have been devoid of joy if it weren't for the following folks. Thank you to Liam Casey and Lauren Dello Russo for always providing a warm welcome to their home. Thank you to Evan Chavis and Miranda Kharsa for their supportive messages, humor, and conversations that I so valued during difficult times. Thank you to Dev Goyal and Girish Kulkarni for being constant pillars of dependability and for their fantastic friendship. Looking forward to lifelong adventures!

I am so incredibly lucky that I can count on my closest friendships from middle and high school every day. These friends have been by my side through good times and bad, and I cannot imagine life without them. Thank you for your ever-strengthening companionship full of warmth and love: David Ellis, Nicole Irizarry (and Sierra!), Edward Kogan, Aditi Yokota-Joshi (and Heathie!), Jon Yu, and Alice Zhang.

Thank you to my extended family members for their undying support since day one. My beloved aunts, uncles, cousins, and grandparents (Saraswathy, Harinath, Annapurna, and the late Suryanarayana) are the well-wishers I depend on every day, so thank you for providing me with a home-away-from-home wherever I am. A special shoutout to Mishka, my cousins' Yorkie, for objectively being the greatest and sweetest dog in the universe. She is a little bundle of energy and love. Miskhu, if you're reading this, know that I had your picture as my desktop background throughout my time at Michigan! And another special shoutout to my baby cousins: Ayush, Kaavya, Smita, and Smaran. Ayush and Kaavya, although I missed hanging out with you as often as I wished, I am still amazed to see you both grow up into inquisitive and thoughtful kids. And to Smita and Smaran, or my "Michigan babies" as I like to call them, you two have given me so much joy and

happiness with the way you approach life (and how you ask all of your "Why?" questions, although I admit that your semesterly "Rohit anna, why are you still not done at Michigan?" question was a tough one to answer). I cannot wait to keep watching you four continue to transform into wonderful human beings!

Penny Triệu, from the bottom of my heart, thank you so much for being the best friend, confidant, cheerleader, life coach, constructive critic, and partner that I could ever wish for or have imagined. If my life at Michigan was like being in a black-and-white movie, the moment you walked in, it transformed into a technicolor masterpiece replete with wonder, adventure, and happiness beyond measure. You have improved my life in uncountable ways, and always make me strive to be a better person. I could not have finished graduate school if it weren't for your ability to calm my nerves, your constant encouragement, and consistent positivity. Although I may not exercise as often as you wish (or ever) and take more days than there are pages to read the books you give me, I cannot imagine going through life's adventures with anyone else. Of all the lives in all the grad schools in all the world, you walked into mine, and for that I am supremely grateful.

If I am anything at all in this life, it is only because of the unconditional love and support that my sister and parents always give me, and I owe all of my success to them. My sister Reshma—or shall I say, Dr. Reshma Gouravajhala—is my daily inspiration and role model in life. I could not be prouder of her brilliance, kindness, ferociousness, and ability to fight for what is right at every step of the way. My mom and dad sacrificed so much in their lives to provide a better life for me and my sister, and I am incalculably grateful for them every single second of every single day. They are the best parents in the entire universe (and all multiverses too!) and have taught me and my sister valuable lessons for how to surmount challenges, conquer fears, and become productive members of society. Although we did not have much while growing up and faced a surfeit of hardships, we always had each other, and I am so incredibly privileged for that. With my family by my side, I can move any mountain. Resh, Amma, and Nanna, I love you three so very much! Thank you for everything.

Finally, I would be remiss if I didn't acknowledge that my dissertation, though the culmination of a years-long effort, is set against a backdrop of unprecedented times, especially in 2020. I want to thank and dedicate further this dissertation's drop of water contribution to the ocean of effort undertaken by society every day: to the public health workers who tirelessly take care of us and work at the frontlines; to the invisible labor of so many that keeps society ticking along (starting with the amazing custodial staff at CSE, whose conversations livened up otherwise dreary late nights); to the immigrants who seek a better life for all; to the activists and fighters of injustice; to happiness pursuers; to knowledge seekers; and to dreamers everywhere.

Table of Contents

List of Figures

Figure

List of Tables

Table

Abstract

The proliferation of artificial intelligence (AI) systems has enabled us to engage more deeply and powerfully with our digital and physical environments, from chatbots to autonomous vehicles to robotic assistive technology. Unfortunately, these state-of-the-art systems often fail in contexts that require human understanding, are never-before-seen, or complex. In such cases, though the AI-only approaches cannot solve the full task, their ability to solve a piece of the task can be combined with human effort to become more robust to handling complexity and uncertainty. A hybrid intelligence system—one that combines human and machine skill sets—can make intelligent systems more operable in real-world settings.

In this dissertation, we propose the idea of using *interactional slingshots* as a means of providing support structure to user interactions in hybrid intelligence systems. Much like how gravitational slingshots provide boosts to spacecraft en route to their final destinations, so do interactional slingshots provide boosts to user interactions en route to solving tasks. Several challenges arise: What does this support structure look like? How much freedom does the user have in their interactions? How is user expertise paired with that of the machine's?

To do this as a tractable socio-technical problem, we explore this idea in the context of data annotation problems, especially in those domains where AI methods fail to solve the overall task. Getting annotated (labeled) data is crucial for successful AI methods, and becomes especially more difficult in domains where AI fails, since problems in such domains require human understanding to fully solve, but also present challenges related to annotator expertise, annotation freedom, and context curation from the data. To explore data annotation problems in this space, we develop techniques and workflows whose interactional slingshot support structure harnesses the user's interaction with data.

First, we explore providing support in the form of **nudging** non-expert users' interactions as they annotate text data for the task of creating conversational memory. Second, we add support structure in the form of **assisting** non-expert users during the annotation process itself for the task of grounding natural language references to objects in 3D point clouds. Finally, we supply support in the form of **guiding** expert and non-expert users both before and during their annotations for the task of conversational disentanglement across multiple domains.

We demonstrate that building hybrid intelligence systems with each of these interactional slingshot support mechanisms—nudging, assisting, and guiding a user's interaction with data—improves annotation outcomes, such as annotation speed, accuracy, effort level, even when annotators' expertise and skill levels vary.

Thesis Statement: By providing support structure that nudges, assists, and guides user interactions, it is possible to create hybrid intelligence systems that enable more efficient (faster and/or more accurate) data annotation.

Chapter 1
Introduction

Intelligent systems provide a powerful way to interact with the digital and physical environments around us. Unfortunately, these systems often fail in contexts that are never-before-seen or full of complexity and nuance. In such cases, human intelligence and effort is often required to make these intelligent systems more robust to handling complexity and adversity. One such means of pairing humans and machine effort together is by creating hybrid intelligence systems.

In this dissertation, we propose the idea of using *interactional slingshots* as a means of providing support structure to user interactions in hybrid intelligence systems. Much like how gravitational slingshots provide boosts to spaceships en route to their final destinations, so do interactional slingshots provide boosts to user interactions en route to solving tasks. We explore the structure and nature of these interactional slingshots in the problem space of data annotation.

1.1 Motivation and Problem Statement

Advances in artificial intelligence (AI) methods and an explosion of available data, along with a concurrent revolution in computers' abilities, has driven AI use in numerous domains, including autonomous vehicles, robotics, computer vision, accessibility, and natural language processing (NLP). However, when faced with tasks that require expertise, nuance, human-level understanding, or are never-before-seen, state-of-the-art AI approaches often fail.

To make these systems more robust to real-world settings, researchers and practitioners have developed methods to incorporate human effort and intelligence into the AI systems' workflows [109], including methods such as: active learning (AL), where machine learning algorithms query humans for data labels; human-in-the-loop (HITL), a

combination of supervised machine learning and active learning in which human input on an algorithm's output is fed back into its input, directly impacting training, tuning, and testing; and, human computation (HCOMP), where human involvement and input is treated as a computational element in the overall task-solving process.

Furthermore, hybrid intelligence systems—systems that combine human (often crowdworkers) and machine intelligence together in order to solve tasks neither can solve on their own—are complementary of AI advances, and have shown promise in helping AI systems overcome some of their limitations. Crowdsourcing, or the practice of obtaining paid human input for a wide range of commodity tasks from online platforms such as Amazon Mechanical Turk [5], enables these hybrid intelligence systems to scale and become more widely deployed (for example, see the deployment of Chorus [46]).

Yet, the nature of hybrid intelligence systems, where technology and humanity work together, means that these systems face similar issues that computer-supported cooperative work (CSCW) systems face, namely the *socio-technical gap*. As Ackerman writes, "There is a fundamental mismatch between what is required socially and what we can do technically" [1]. If we want to solve problems being faced by humans, then one way to go about it is to build *better* hybrid intelligence systems, for different interpretations of better. Indeed, Lasecki has argued that collective intelligence in the form of crowdsourcing and HCOMP has the potential to bridge the socio-technical gap, and lead to users "attain[ing] super-human performance on a wide range of tasks that they may seek to accomplish" [64]. Though this may be overly-optimistic and over-claiming progress, such a hybrid system might still be able to partially overcome this gap.

Complementary to the above-discussed methods that make AI methods more effective, we are interested in closing this socio-technical gap by focusing on the user's experience when they interact with a system. That is, we want to improve the user's experience in using these systems for solving problems, thereby making their effort more efficient.

To examine what it means to lessen the user's effort as a tractable socio-technical problem, we situate our work within the space of data annotation. Not only is getting annotated and labeled data critical for the success of these AI approaches, but also humans are the crucial annotators of such data. These are annotation problems that require human understanding, ability, and interactions with data to fully solve, but nevertheless benefit from AI systems' ability to solve parts of the problem with computational tools. What makes data annotation a particularly intriguing problem space is that there is natural interplay between the human, machine, or both, when it comes to certain characteristics, since data annotation: requires expertise (domain knowledge or expertise can reside solely

with the machine, human annotator, or both); requires context (context might be curated, again, solely by the machine, human, or both); and, involves coordination (some tasks are easier solved by coordinating between machine components, some by coordinating between only the humans, and some by a combination).

We hypothesize that providing support to user interactions with the data itself can strengthen the partnership between human and machine in these hybrid intelligence systems. For instance, how well can people learn to do a task or use the interface for solving a task? How well can the system provide the right context? Can the system provide support to the user, and vice-versa, and if so, when?

We specifically focus on the last question and introduce the idea of using interactional slingshots as the means of providing support structure to user interactions in hybrid intelligence systems. The user interactions that we focus on are those utilized during the annotation process, by which we mean are the set of interactions in which a worker takes input data and annotates it with useful information, after which we get annotated output.

We introduce the following definition for an ***interactional slingshot (IS)***: *A computational tool that provides a supporting "boost" to the user interaction by solving a related piece of the overall task, but not the full task itself.* Much like how gravitational slingshots provide boosts to spaceships that are en route to their final destinations—thereby saving fuel and time—so do interactional slingshots provide boosts to user interactions en route to their solving tasks—thereby providing accuracy gains and time savings. As a result, interactional slingshots cannot solve the entire task, and only serve to provide support to user interactions with the system.

An example of a user interaction with a slingshot is as follows: Suppose a person wants to find a basketball that they know is located in the basement. Rather than searching the basement by every square inch, the person can first turn on the overhead lightbulb; in this case, the lightbulb acts as the slingshot that provides a supporting boost to the person's task of finding the basketball. Of course, turning on a lightbulb by itself doesn't solve the overall task, but it does help the person become more efficient (e.g., the person can find the basketball faster). If the basketball were to have a GPS tracker on it, that would not count as an interactional slingshot, since the overall task—finding the basketball—is completely solved by the computational tool (in this case, the GPS tracker). In this thesis, we explore the forms that interactional slingshot support can take.

Project	Research Question	Annotation Domain	Annotator Expertise	Support Type	Interactional Slingshots
Mnemo [38]	RQ1	Two-party chat logs	Non-expert	Nudges	- Reminders - Free-form note creation - Association with text - Aggregation of output
EURECA [41]	RQ2	3D point clouds	Non-expert	Assistance	- Mixed-initiative - Automated filter and fill of 3D points - Collaboration with other workers (visual context of selections)
MDCD [60] and Ch. 5	RQ3	Multi-party chat logs	Expert and Non-expert	Guidance	- Username highlighting - Machine learning model predictions

Table 1.1: Summary of thesis contributions.

1.2 Thesis Statement and Research Questions

We evaluate the following thesis statement in this dissertation:

> *By providing support structure that nudges, assists, and guides user interactions, it is possible to create hybrid intelligence systems that enable more efficient (faster and/or more accurate) data annotation.*

To evaluate that thesis statement, this dissertation explores three ways in which we provide support to user interactions, listed here as research questions:

- **[RQ1]**: For a task that relies on extracting latent mental models that could differ across annotators, what are the challenges associated with providing support by *nudging* a non-expert user's interactions with data?
- **[RQ2]**: For a task that involves dealing with spatial ambiguity, how can we create interactional slingshots that provide support by *assisting* a non-expert user's interactions?
- **[RQ3]**: For a task that is difficult and requires expertise, how effective are interactional slingshots that provide support by *guiding* those interactions, and how do non-experts and experts perceive them?

Each research question, and associated project, is summarized in Table 1.1, and expanded upon in the next section.

1.3 Research Contributions

In this dissertation, we demonstrate that building hybrid intelligence systems with each of these interactional slingshot support mechanisms—nudging, assisting, and guiding a user's interaction with data—improves annotation outcomes, such as speed, accuracy, and effort level, even when annotators' expertise and skill levels vary.

We situate these hybrid intelligence systems in three classes of data annotation problems: collective crowd memory, natural language grounding for objects in 3D scenes, and multi-domain conversation disentanglement. In each domain, respectively, an AI system could not immediately solve the overall task, whether due to identifying relevant information to remember about a user over time, dealing with novel objects in never-before-seen environments, or understanding complex input data that requires domain knowledge. As a result, this dissertation makes the following contributions for each research question listed above:

1. **RQ1** – Support as *nudging* in Mnemo [38]: By nudging non-experts' interactions with data to make them keep in mind the time-frame of their annotations, we show that an interface and methodology to help extract notes from conversations can be effective. We are the first to present a system for capturing this information concisely, and show that the interactional slingshots can aggregate effort from different crowd workers. However, we also show that nudging itself may not be sufficient when the task is too ambiguous, contains nuance that's endemic to human-human dialog, and provides too much freedom for the annotator (in the task, annotators summarize sentences in their own words, which introduces a lot of variability into the data annotations). See Chapter 3 for details.

2. **RQ2** – Support as *assisting* in EURECA [41]: We introduce a mixed-initiative system that enables workers to ground natural language references to objects in 3D scenes. User interactions with this system are assisted by the computational tools baked into the system itself, and in so doing, the human-machine collaboration resembles that of a critic system, harkening back to the approximation discussed in the socio-technical gap by Ackerman [1]. We show that human effort is substantially reduced when groups of people can collaborate with—and not simply use—interactional slingshots by providing initiative to the computational tools themselves (the assistive support here enables mixed-initiative interactions). Finally, we perform real-world case studies to study the efficacy of the EURECA hybrid intelligence system. See Chapter 4 for details.

3. **RQ3** – Support as *guiding* in Conversation Disentangling [60]: What happens when expert and non-expert users rely on the guidance provided by interactional slingshots? We show that, when both people with and without domain knowledge in computer science use our tools laden with interactional slingshots (which provide visual context, as well as predictions made from AI models), they are able to annotate text in multiple domains more accurately than if they were using interfaces without slingshot support. We test our interfaces with chat logs spanning four technical domains. We also show that non-expert crowd workers are able to succeed at getting higher quality annotations when using the interactional slingshot tools versus the baseline tools, sometimes beating the machine's performance, although as we discuss, limitations exist. See Chapter 5 for details.

1.4 Dissertation Outline

The rest of this dissertation is organized as follows:

- Chapter 2 describes background and related work that provides the bedrock upon which this dissertation is built.
- Chapter 3 describes Mnemo, a crowd-powered dialog system plugin that enables a way for workers to generate facts about users.
- Chapter 4 describes EURECA, a mixed-initiative system that enables workers to collaborate *with* computational tools, rather than simply use them.
- Chapter 5 describes MDCD, a novel interface that enables both experts and non-experts to disentangle chat logs across multiple domains.
- Chapter 6 summarizes the thesis, and discusses implications and future work directions.

Chapter 2
Background

This chapter reviews literature in the two domains in which we evaluate our work: Natural Language Processing and Robotics. Our work builds on previous research into crowd-powered conversational systems, context maintenance, and conversation disentanglement. Our work also builds on work related to crowdsourcing, human computation, 3D sensing for robotics, and visual scene understanding.

2.1 Crowdsourcing and NLP

2.1.1 Crowd-powered Conversational systems

Traditional conversational systems have been shown to be time-consuming and based on domain expertise [114]. Crowdsourcing has shown its huge potential as a fast and cost-efficient way to improve traditional computer systems by integrating human intelligence and knowledge [52] into automated methods in a variety of areas, such as protein prediction [20], image search [115], speech recognition [63], and multi-step writing and editing [9]. Due to the difficulty for software agents to handle the complexity of human language, many systems have used the crowd to interpret natural language data in document editing [8], twitter response generation [10], and vacation planning [58]. However, those systems can only handle a single round of computation because of the always-changing pool of crowd workers. Since there is no single worker that can be relied on to respond at any given time, it is inherently difficult for systems to maintain consistent communication with the different crowd workers.

In order to integrate the crowd in two-way interactive system, researchers recently started to coordinate crowd efforts in real time. Vizwiz is one of the first systems to obtain responses within seconds from the crowd by using a queuing model to recruit on-demand workers for later tasks [11]. Since real-time conversational systems require multiple on-

demand workers to be available at the same time, ChatCollect introduced a two-way dialog collection method to generate realistic conversations about definable tasks from pairs of workers [67]. To aggregate workers' efforts for better responses, Legion proposed a collective model of using multiple workers as a single agent to control existing user interfaces using natural language [70]. That model was later applied in Chorus, an intelligent conversational agent where workers interpret natural language interactions and collectively vote for the best response [75].

2.1.2 Consistency between Conversations

Good conversational systems need to maintain consistency, which we define as the capability to generate non-conflicting responses based on context discussed in previous sessions. To fill out the gap between the requirement of conversational consistency and the always-changing pool of crowd workers, the idea of collective memory has been proposed, in which workers are able to remember information over time that can benefit crowd-powered systems via learning [62]. Chorus allows workers to identify conversational segments containing critical information as facts to direct future workers to the most important aspects of the chat history [46, 75], but it still requires the history to be dense enough to be understood. Since Chorus only maintains and updates a list of 10 facts, the size of crowd-generated collective memory is limited when conversational history grows with an increasing chance of missing important facts.

To address that issue, attempts have been made to allow crowd workers to curate given data, which refers to the process of selecting, organizing, and maintaining a collection of material. Crowdsourced text summarization has been broadly used to extract the most important message from documents [110] or to summarize others' comments to post reflection [55]. Moreover, crowd workers have been shown to successfully summarize critical information from conversations into facts using their own words, as suggested in [65]. Jiang et al. propose "target summarization" as a technique to elicit higher quality paraphrases from crowd workers [48].

In order to reduce the amount of effort needed to identify meaningful memories within a large collection of digital content, Kurator introduces a hierarchical crowd-machine learning architecture that greatly improves the efficiency of the curation process [83]. Although these references contribute effective workflow for conversational context summarization via crowdsourcing, the feasibility of using crowd-generated memory facts to maintain conversational consistency and improve future problem-solving efficiency in

conversation still remains unstudied. To address that gap, our Mnemo introduces an alternate solution by allowing crowd workers to directly summarize memory facts with their estimation of expiration date, which allows us to investigate their ability to determine what information is critical or relevant in the long term.

2.1.3 Conversation Disentanglement

There has been more than a decade of research into conversation disentanglement [102], as automatic disentanglement of conversations can be used to provide more interpretable results when searching over chat logs and to help users understand what is happening when they join a channel. The most influential work has been on disentangling the #Linux channel [29–32]. There are other IRC disentanglement datasets: one studied disentanglement and topic identification, but did not release their data [3]; one had annotations for the #Ubuntu channel, but for the French version [92]; and, one that used heuristically extracted conversations [79, 80]. Finally, there is other work that use non-IRC logs as a way to do disentanglement in domains such as studying classes [28, 113], support communities [81], and customer service [27], but only one [28] is available as a publicly-released dataset.

With respect to the task of conversation disentanglement, annotators were always trained experts, as non-experts were thought to not contain the expertise required to disentangle these datasets. The approach used to disentangle data was either to create a ”reply-to” graph structure (i.e., for each message, identifying to what previous message or messages the current message is replying to) or separating chats into constituent conversations [30]. Machine performance on this task does not match that of human performance, even when models are trained on copious amounts of data [60].

2.2 Crowdsourcing and Robotics

2.2.1 Crowdsourcing and Human Computation

Prior work has focused on annotating objects using computer vision [90, 104], as well as on offline image labeling and model-building for robotics using data generated by crowdsourcing [105]. Crowdsourcing has also been used to augment robots with human intelligence to, for example, navigate a maze [86], and real-time crowdsourcing has been used to provide continuous control for an off-the-shelf robot that enabled it to follow NL

commands [71], over unbounded lengths of time [76] (thanks to the availability of crowd workers), with reaction times under 1 second [89]. Recent work has explored "hybrid intelligence" workflows that leverage both human and machine intelligence to solve tasks that neither could accomplish alone [94, 103].

Our work draws heavily on real-time crowdsourcing, which makes it possible to get rapid responses from crowds of workers—often in less than a second. This response speed makes it possible to create crowd-powered interactive systems. By leveraging real-time crowds to quickly annotate 3D scenes, we make it possible to interact with robots using natural language in real-world scenarios, even when the robot has no prior training on them. For example, Lasecki et al. have created systems capable of generating captions with less than 3 seconds of latency per word [69] and creating functional UI prototypes [68]. Berstein et al. have created systems for finding the best image from a short video, and generating varied images from a single source [7].

2.2.2 Robotics and Semantic Mapping

Point cloud data has enabled geometric mapping of 3D space [33,37,82] and a proliferation of robots capable of autonomous navigation in both indoor and outdoor environments. However, the perception capabilities are often limited to the mapping of space without a semantic parse of individual objects, as well as their afforded actions and language groundings. Even for simple object affordances ("picking" and "placing"), language annotation of objects is essential to establishing a common ground of object references that is both intuitive for humans and perceptible by robots. This problem of semantic mapping [57, 96] is often addressed through combinations of object detection, segmentation, and pose estimation. Each of these modules depend on some form of *a priori* information about the object, such as training data, object geometries, or probabilistic priors. This prior knowledge is provided to an autonomous agent as an object model and this modelling requires data collection through annotation.

2.2.3 Creating Object Geometries

For the robotic manipulation of objects, model-free approaches [35, 108] reason geometrically over 3D point clouds to grasp objects. Such methods do not attempt to semantically distinguish individual objects, and are unable to provide a common grounding for human-robot interaction or reason in a goal-directed manner. Methods using object geometries [26, 85, 88, 107] address these shortcomings, often through a combination of

generative and discriminative inference. However, such methods then rely upon object models to be provided *a priori*. As will be seen in Chapter 4, EURECA, as a crowdsourcing-based data annotation system, offers one viable option to building such object geometries suitable for real-world scenarios.

2.2.4 Scene Annotation Interfaces

In general, there is a lack of annotation interfaces for visual scenes; this is typically because existing work has focused on creating datasets for these applications offline, often curated by experts. Helpful tools like [95] are used to create large datasets, but are often time-consuming to create. Furthermore, our envisioned use cases are for the in-home or in-office setting, where preserving privacy becomes an important concern, especially in crowd-powered systems [51]. Indeed, removing the RGB data from an image will make it resemble a flat grid; without RGB, performing the kind of outlining discussed in Russell et al. [95] would not be as feasible. If an application wishes to work with another sensor, such as a Velodyne, lack of RGB would be the norm. However, with the 3D point selection that EURECA makes possible, privacy is maintained and segmentation is done online and in real-time.

2.2.5 Visual Scene Understanding

Since robust, general-purpose computer vision is still a distant goal, visual scene understanding via human computation has been explored by several prior projects. Objects and activities have also been recognized in video using the crowd—Glance [66] coded behavioral events, Salisbury et al. augment live video with natural language markers using real-time crowds [98], and Legion:AR [73] recognized human activities in real time. By building upon this body of work, we will integrate visual scene understanding into EURECA.

The ESP Game [111] was one of the first image labeling systems to engage crowds. It focused on providing alt-text for web images. VizWiz [11] and Chorus:View [74] leverage real-time crowds to answer visual questions for blind and low-vision users in their daily lives. RegionSpeak [117] elicits bounded regions for this Q&A task, which provides users with richer responses. LabelMe [95] used fine-grained bounded regions to train computer vision. Deng et al. [25] incentivized workers to indicate which visual information is most important when classifying images, allowing visual features to be extracted for training more accurately.

2.2.6 Robotics and Autonomous Control

Prior work has also explored how to use crowdsourcing to augment robotics. Crick et al. show that users provided reliable demonstrations and training for robots when the robots were in sensory-constrained environments [21]. Legion [72] used real-time crowds to provide continuous control for an off-the-shelf robot that enabled it to follow natural language commands. While Legion provided generalized user interface control, Salisbury et al. [99] introduced additional control mediators that improved performance by focusing specifically on robotics applications. de la Cruz et al. [23] got feedback from a crowd of workers in $\sim$0.3 seconds for mistakes made by an automated agent. Chung et al. [18] explored learning from initial demonstrations using crowd feedback for motion planning problems.

2.3 Conclusion

The research literature that we presented in this chapter helps inform our decision-making and approaches as we evaluate our thesis statement. Each of the ideas—crowdsourcing in an NLP context, in the robotics domain, and expert annotations for conversation disentanglement—will be the data annotation grounds within which we explore our research questions.

Chapter 3

Support in the Form of Nudging: Collective Conversational Memory for Crowd-Powered Dialog Systems

In this chapter, we explore what it means for interactional slingshots to nudge user interactions towards a particular goal. We address RQ1 and create a methodology for *fact generation* for conversational context maintenance as human-generated facts from goal-oriented dialogs. We evaluate this with a plugin, Mnemo, a crowd-powered conversational plug-in that allows crowd workers to read dialogs and predict, curate, and save critical information as notes for future conversations, not yet a capability of existing crowdsourced summarization techniques. We show that nudging itself is not powerful enough of a support mechanism to prevent users from annotating text in such a way that consensus cannot be reached well. This is because the annotator's freedom is unaffected by nudging support and leads to great variability in annotations, although as we discuss, this still bodes well for the overall task.

Conference: *Finding Mnemo: Hybrid Intelligence Memory in a Crowd-Powered Dialog System*, Collective Intelligence, 2018.

Coauthors: Youxuan Jiang, Preetraj Kaur, Jarir Chaar, Walter Lasecki

Collaborations: IBM Research

3.1 Motivation

Developing intelligent conversational systems that can interact with humans using natural language has been a longstanding goal of both artificial intelligence and human-computer interaction [4]. Automated systems, such as Apple's Siri, Microsoft's Cortana, and Amazon's Alexa, bring us closer to achieving this goal, and have been widely used in assisting users with task completion. However, the complexity innate to dealing with natural language, as well as identifying user intent, limits the capabilities of many of these systems to a set of predefined tasks or queries.

As a result, automated conversational systems struggle to capture the richness found in long term human interactions, which leads to the loss of conversational context important for human-human conversations. Examples of this context include users referring to prior interactions or any context-dependent phrases, which automated approaches have difficulty recognizing and capturing from the rest of the conversation. This is especially evident in the case of any new people being added to the conversation, as they will lack any context from any conversations that have taken place thus far. To maintain important context for sequential dialogs, many conversational interfaces save their entire chat history for future participants to read and comprehend [6, 34]. However, that method is barely useful in maintaining context of real-time chat because people can easily miss out on new content when they scroll up, especially with the out-of-order turns and high signal-to-noise ratio [112].

Crowd-powered conversational systems have been shown to successfully overcome some of the struggles faced by purely automated dialog systems with respect to capturing the richness found in human dialog, since these systems leverage crowds of human workers to respond to end user queries. These systems have shown that workers can recover context from prior conversations when presented with user-related information [46, 75]. Although those crowd-powered methods are able to direct future workers to understand context for an on-going conversation, they still require the chat history to be comprehensive enough to be understood, as well as require that the context exist in the first place. Moreover, such systems also fail to map information from past conversations to present ones, leading to a lack of contextual memory about the user and unnecessary repetition of information across conversations. Any conversational context is also lost when interactions take place over longer time scales and span multiple sessions, especially when no one crowd worker is around for all sessions. There is currently no way to extract concise, conversational context from those dialogs. 0

This is, of course, in stark contrast to human-human interactions. When interacting with each other over a long time, people store memories about their conversing partners and are seemingly able to recall long-term facts relevant to the particular topic at hand with relative ease. Collective memory, then, can be extremely useful in sharing knowledge about the user, and falls under the umbrella of the "first generation in knowledge management [as a] repository" [2]. Although similar in spirit to the early CSCW efforts outlined by Ackerman, our problem exploration entails not how people would use information repositories, but rather, how peopl can generate them in the context of conversational memory. Remembering such facts enables more efficient, perhaps even richer conversations. What if we could tease out these latent models of fact-saving that we innately build and instantiate this fact storage in our dialog systems?

3.2 Contributions

We propose a methodology for *fact generation* for conversational context maintenance by saving and aggregating human-generated facts from goal-oriented dialogs. We implement and evaluate the model in Mnemo, a crowd-powered conversational plug-in that allows crowd workers to read dialogs and predict, curate, and save critical information into facts that will be relevant for *future* conversations, which is not a capability of existing crowdsourced summarization techniques. Our findings show that combining worker-generated facts (which would be hard or impossible to do with summaries) provides higher F1 score than just combining lines from a conversation.
Specifically, we present the following contributions:

- We provide a first system exploration into the problem space of crowdsourced prediction and curation of future-relevant facts by introducing Mnemo, a crowd-powered dialog system plugin that allows crowd workers to effectively identify such relevant facts;
- We characterize the types of facts captured from conversations by the workers and show that worker "errors" are often not true errors, but are missing important context or have relevance for only a short amount of time;
- Aggregation methods for crowd-generated facts which act as tunable "knobs" that allow collective responses to outperform individuals' responses.

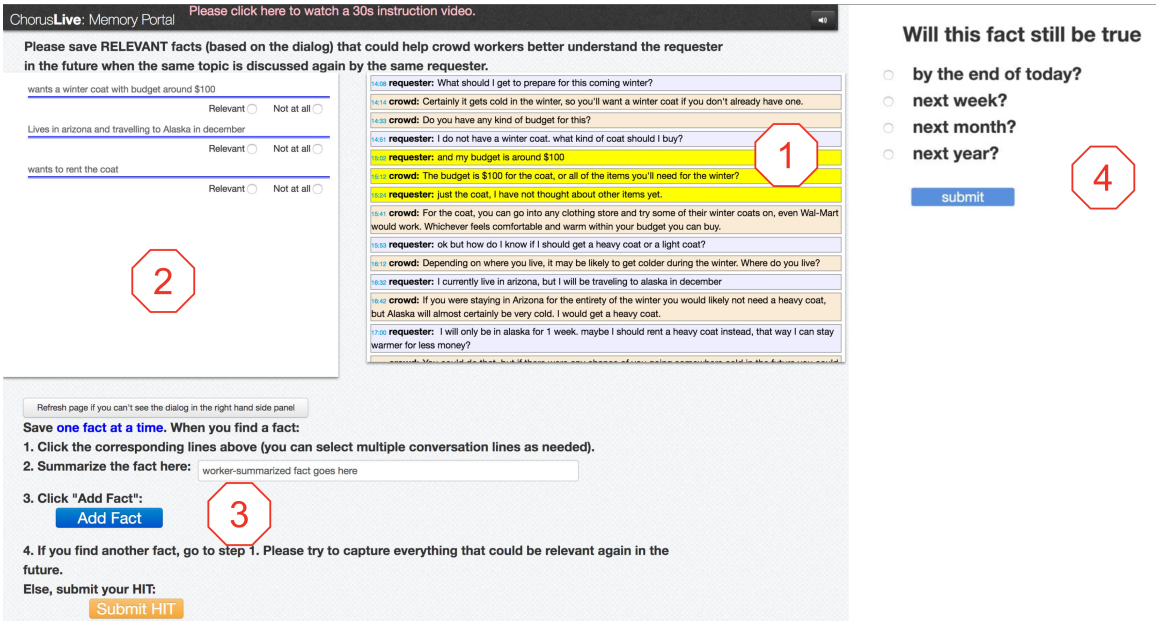


Figure 3.1: Mnemo's Fact Creation Interface. The worker interface consists of four parts: (1) DialogView: display raw dialog lines; (2) FactView: display already-saved facts; (3) FactSummary: allow workers to summarize facts; and (4) TimeSelect: allow workers to estimate fact longevity.

3.3 Mnemo Interface

We built the Mnemo plugin on top of the existing Chorus system. Mnemo's interface (see Figure 3.1) consists of four parts: (1) the DialogView panel, where we show the dialogs under consideration. Workers can select one or more sentences from the dialog and group them as part of one "fact"; (2) the FactsView panel, where worker-created facts are displayed. Workers can click on the fact, which opens a dropdown that shows the lines from the conversation associated with that fact; (3) the FactSummary panel, where workers summarize relevant information in the selected lines into a fact; (4) the TimeSelect view, which allows workers to estimate the expiration time of the fact by picking one of the four time labels: will be true for a day, a week, a month, and a year.

By *relevance* for the TimeSelect modal, we refer to facts that will still be true about the user in the long term (e.g., a requester's allergy information), rather than facts that will be true in general (e.g., information about a particular restaurant). The information contained in the latter category can be obtained by searching existing information sources, but the former contains information that must be extracted from the requester's conver-

Topic	Scenario	Length	WF	GTF
Food	Obtain a quick lunch	30	3.4	4
	Go on a fancy date	31	6.5	4
Shopping	Buy a coat for winter	17	3.9	5
	Buy Christmas gifts	20	5.5	7
Jobs	Get a summer internship	21	5.8	7
	Get a full-time job	21	4.9	7
Laptop	Buy laptop for school	29	5.8	6
	Replace broken laptop	52	6.9	6
Pets	Get a pet for oneself	20	3.7	5
	Get a pet for a partner	22	3.6	3

Table 3.1: Summary stats for all 10 dialogs. There are two scenarios per dialog topic. Length = number of lines in each dialog; WF = the average number of worker facts for that dialog; and GTF = the number of ground truth facts.

sation, which maintains the context for future workers. For our studies, we define long term as any fact that will still be true even after a month (or longer). This allows us to avoid user information that is important in the intermediate time frames, and focus on user information that is relevant when the user makes a request at least a month later.

3.4 Dialog Creation

To collect dialogs where a helper provided advice to a requester's queries, we recruited 5 student participants to generate 10 human-human dialogs across 5 different topics. The study participants conversed in pairs using a simple chat interface in a round-robin fashion. None of the participants had any background with or prior experience using the Mnemo.

Each pair of participants is tasked with generating two dialogs about two scenarios within the same topic (Table 3.1 lists the topics) in order to mimic real-world use cases. For example, for the food topic, the two scenarios are *Find a restaurant for quick lunch* and *Find a fancy restaurant for a date*. One participant is assigned to the "requester" role and keeps this role for both scenarios; the other participant is assigned to the "crowd"

role and switches to another topic between sessions. Switching the "crowd" role worker allows us to emulate the scenario where the same requester discusses the same topic at another time with different crowd workers, which is very common in real world settings due to full turnover of workers in reality.

For each task, the requester received an initial script to follow (created by the authors), which contains the preferences of their assigned identity (e.g. likes Mexican food) and considerations for their query (e.g. find a good restaurant for lunch). Participants were instructed to follow the script as closely as possible, but they still had the freedom to converse naturally.

Summary statistics for each generated dialog can be seen in Table 3.1. Each generated dialog has an average of 26 lines, along with an average of 12 words per sentence. We use these dialogs for our experiments, described next.

3.5 Experimental Design

Given a conversation about a topic and a requester's context and preferences, *how accurately can workers predict relevant facts about that user for use in future interactions?* To answer that question, we conducted the following experiment using Mnemo to characterize types of facts generated by workers and explore the viability of crowd-curated conversational context.

The procedure is as follows (refer to Figure 3.1): workers are presented with one of the dialogs from the 10 scenarios and are instructed to save relevant facts, one at a time, by clicking on the corresponding lines in the dialog. They can select multiple lines from the dialog, then summarize in their own words into a "fact." Workers were instructed to only save facts that will be relevant about the requester, so any facts that are true about the world will be considered irrelevant. There was no limit on the number of facts each worker should submit, but they needed to submit at least one fact in order to receive compensation for completing the task. After summarizing each fact, workers were asked to use the TimeSelect modal to affix each fact with an estimated expiration date, ranging from one day, one week, one month, and one year.

For each dialog, 10 unique workers were recruited on Amazon Mechanical Turk to save facts. Workers were paid $0.67 per task and the expected task duration was four minutes, for an effective pay rate of about $10 per hour.

Ground Truth

As mentioned in the Interface Section, we consider worker-generated facts to be long-term only if they still apply to the requester even after a month. These facts are traditionally the most difficult to maintain between different conversational sessions for the same user.

To define the relevant facts for ground truth in each dialog, two of the authors used Mnemo's interface to independently generate relevant facts for the 10 dialogs. The final gold standard fact set was constructed by taking the union of both authors' fact sets. Since the authors developed the scripts that the participants followed in the Dialog creation stage, rather than asking the participants themselves, we use the preferences assigned to the identities as seeds for the ground truth long-term facts; as a result, this ground truth serves as a proxy for a real-world usage scenario. The numbers of long-term ground truth facts range from 3 to 7 per dialog.

For worker-generated facts whose summaries contain multiple pieces of information (e.g. "a computer science student looking for a summer internship"), the authors manually broke them down into constituent pieces (e.g. "a computer science student" and "looking for a summer internship") in order to ensure that different relevant facts existing in one single statement would be counted individually for precision and recall. To avoid the need for manual breakdown of worker-generated facts, future work should provide more specific instructions about the exact definition of "one fact" per submission rather than letting workers interpret themselves.

Evaluation Measures

To evaluate workers' ability to capture relevant facts about the requesters, we use *precision* to measure the percentage of worker-generated facts that were in the ground truth and *recall* to measure the percentage of all relevant ground truth facts that were retrieved by the workers. We also use the *F1 score* (the harmonic mean between precision and recall) to give us a single overall accuracy measure, which also permits us to examine the tradeoffs between precision and recall.

For our system, high precision implies that workers are able to succinctly predict facts that will be relevant in the future without too many irrelevant facts included; high recall implies that workers capture most or all of the relevant facts in the ground truth for that conversation. We set the default precision value, which represents no guess, to be 1, since no guess from a worker is better than a wrong guess about the user.

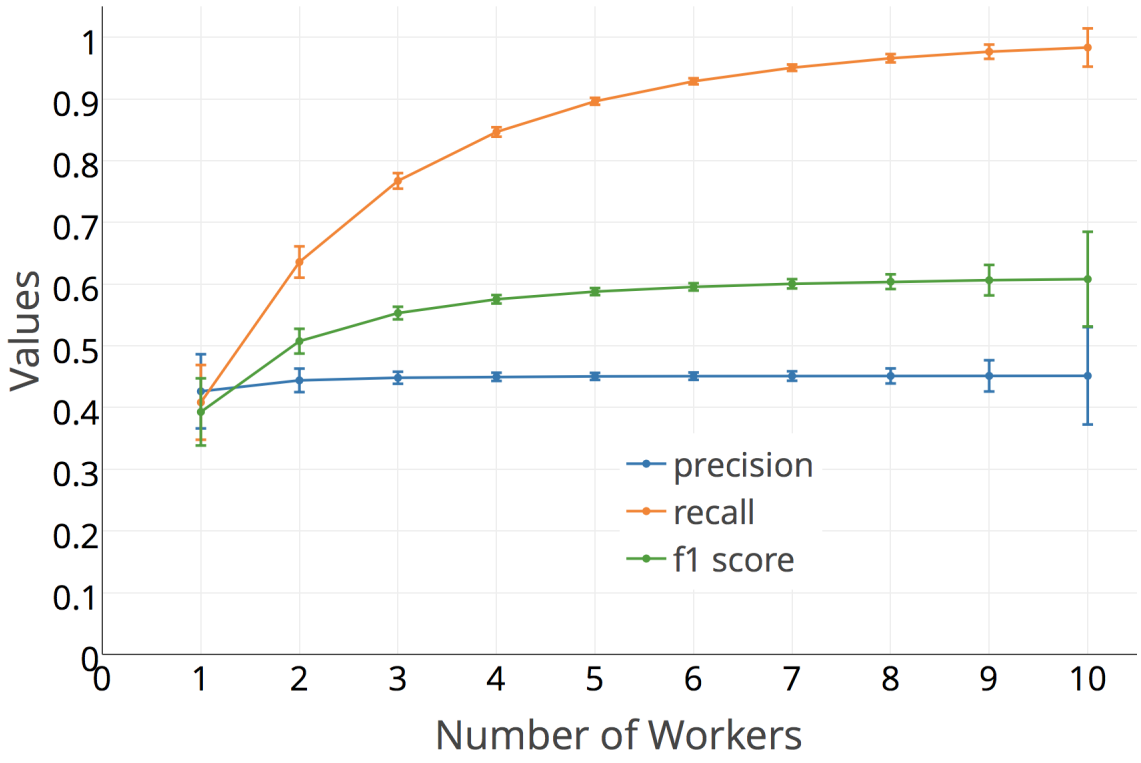


Figure 3.2: Precision and Recall for worker sizes across all dialogs. We find that we need just 5 workers to reach at least 90% recall.

3.6 Results

There were a total of 500 worker-generated facts distributed across the 10 dialogs, with an average of 5 facts per worker ($\sigma = 2.87$). We manually annotated each worker fact as either being a true positive (match a ground truth fact) or a false positive (do not match any ground truth facts). When averaged across all conversations, individual workers' precision and recall are 44% and 42%, respectively. We also see that individual precision and recall values are similarly consistent such that no one topic outperformed all others, as seen in Figure 3.3.

3.6.1 Characterizing Worker Errors

Of the 500 worker-generated facts, 214 were classified as true positives (42.8%), and the remaining 286 (57.2 %) were considered as false positives. In order to present a systematical analysis of workers' errors, we manually categorized each of the false positive facts into six categories depending on the error types (See Table 3.2).

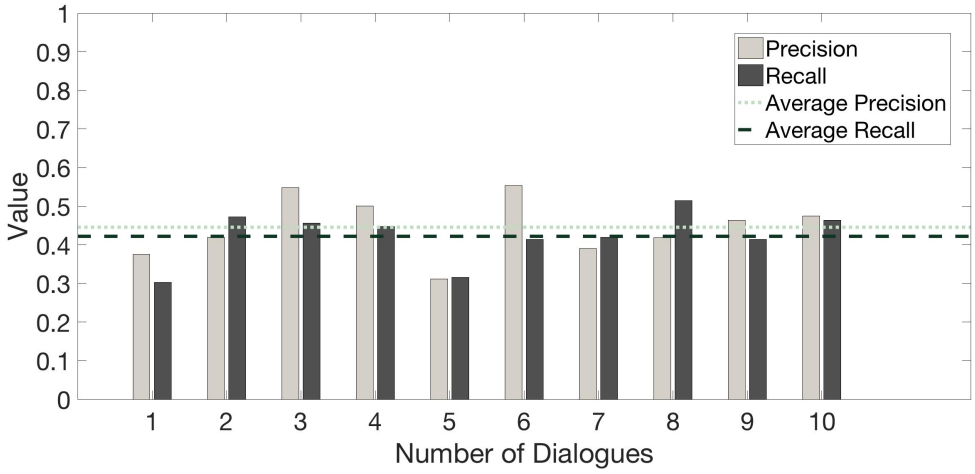


Figure 3.3: Precision and Recall for Individual dialogs. We can see that worker performance is relatively consistent across all the different dialogs.

A: Completely wrong

13 facts, or 4.6% of all the 286 false positives, were classified as being completely wrong when looking at the worker-generated summaries. These facts can be gleaned from pieces in the dialog, but no useful summarization took place. An example of a Category A fact: "very fact to the myself."

B: Missing context

58 facts, or 20.3%, were missing a key piece of context for the fact. For instance, the fact "Green Bay, Wisconsin" is missing the critical piece of "located in" or "lives in"; the facts are not relevant without these critical pieces. Indeed, if the key pieces of information had been included in the worker summaries, these Category B facts would all be considered true positives.

C: Statement about the world

18 facts, or 6.3%, were statements that were made about the world at large, rather than specifically about the requester. Example of a Category C fact: "tanks are like a glass or plastic cages that have no holes in them, you can keep fish in them."

Category	Description	Makeup
A	Completely wrong	13 (4.6%)
B	Missing context	58 (20.3%)
C	Statement about the world	18 (6.3%)
D	About requester, but short-term only	121 (42.3%)
E	About requester, but not in ground truth	45 (15.7%)
F	Presupposed information	31 (10.8%)

Table 3.2: Summary stats for all 10 dialogs. There are two scenarios per dialog topic. Length = number of lines in each dialog; WF = the average number of worker facts for that dialog; and GTF = the number of ground truth facts.

D: Statement about the requester, but short-term relevance only

121 facts, or 42.3% of all false positives, were statements about (or involved) the requester, but are classified as only being relevant in the short term. These facts include any specific preferences the requester expressed, as well as general information from the requester's conversation. An example includes: "Spilt his coffee over laptop, the laptop is fried."

E: Statement about the requester, but not picked up in ground truth

45 facts, or 15.7%, of the facts were statements about the requester, but neither author saved this information as part of the ground truth. Though we consider these facts as being wrong for our evaluation, we nevertheless recognize that they are important. In fact, these are facts that workers recognized as being relevant for the future, even if the authors did not, so these could be viewed as belonging to the true positive pile as well. An example: "Gives 20% tips for good service."

F: Presupposed information

31 facts, or 10.8%, were summaries that contain presupposed information necessary for the facts in the ground truth. We defined "presupposed information" as background or expositional information about the requester related to the current discussed topic, but not helpful for future conversation. For example, these include: "lives in an apartment."

3.6.2 Case Study: Focusing Workers on Specific Time Frames

We see that worker responses contain numerous false positives, but not all of them are completely wrong. Some of the categories were marked as irrelevant because they belonged to a shorter time frame. However, what happens to worker performance if we have a time frame already in mind for relevance?

We investigate whether focusing workers on specific time frames through guided instructions can improve performance and choose a time frame of six months (average of the two long term conditions of one month and one year). For this experiment, workers use Mnemo's interface to create facts, but they are not shown the TimeSelect view anymore; rather, they are explicitly instructed at the beginning of the task to save facts that will still be relevant six months or longer. We randomly selected Topic 1 and reran data collection for the two scenarios, again with 10 workers each.

Table 3.3 shows the results from this experiment: we can see that worker performance improves when they are given a focused time frame in the task query, as they capture more true positives with fewer overall facts created. When compared with the facts for Topic 1 in the "TimeSelect" condition, we can see that a lot of worker error categories (A, B, and F) have disappeared in the "FocusedSelect" condition.

3.7 Collective Performance

While our results show any individual contributor is imperfect (as should be expected), extensive work in crowdsourcing has focused on how groups can collectively outperform individuals. In this section, we outline methods for improving overall precision and overall recall given multiple crowd workers by aggregating workers and then clustering facts sharing similar content. We show the increase in performance along both of these measures using the workers' responses collected above.

3.7.1 Improving Recall with Aggregation

Since individual performance in terms of recall is just over 40% on average, we explore whether combining workers into groups would improve performance, because having more contributors increases the chance to bring in more relevant information.

Category	TimeSelect	FocusedSelect
A	7	0
B	13	0
C	5	6
D	27	15
E	15	6
F	3	0
Total facts	99	66
True positives	29	**39**
Precision	0.39	**0.63**
Recall	0.30	**0.45**

Table 3.3: Worker-generated facts for Topic 1 in the initial "TimeSelect" condition and the "FocusedSelect" condition. By providing focused queries to the workers, we are able to capture more true positives with fewer overall facts, as well as reduce other categories of worker errors.

We measure the effect of aggregation on worker summaries by calculating average precision and recall across all possible combinations of each group size from 2 to 10. This provides a more robust measure of group performance that is less tied to the specific members' performance. The results show that recall increases steadily as more workers are added to each group. Figure 3.2 shows that, on average, we need 5 workers to exceed 90% recall. This implies that each additional worker brings in new information that allows for improved recall, meaning that the diversity of different responses is high.

However, as can be expected, precision does not increase when additional workers are added, since adding more people also increases the chance of adding noise to the system. To address this issue, we developed two clustering methods to investigate the impacts of workers' agreements on fact precision, which will be discussed below.

3.7.2 Improving Precision Through Voting

When combining workers into different group sizes, rather than aggregating facts indiscriminately across all input, we use an agreement or voting scheme based on the idea that the facts more workers agree on are more likely to be true positive.

Agreement-based Clustering

If there are facts that contain similar content, we can cluster them and find "representative" facts for that cluster. We devise two methods for that agreement-based clustering:

1. Clustering based on worker summaries (`Word` clustering):
   In this method, we consider two facts as "share similar content" if at least half of the facts share similar words. We define two words as similar if the Levenshtein distance between them is less than or equal to 2.

2. Cluster based on selected lines (`Line` clustering):
   In this method, we cluster two facts together if the workers select the exact same lines from the dialog when creating those facts. This clustering method uses the raw dialog lines and does not use the worker generated summaries.

Facts assigned to the same cluster are considered as a single fact in measurement and represented by the most frequent label (true/false positive) from members of the cluster. We then calculate precision and recall based on three levels of workers' agreement: 1) **any** agreement, in which at least 2 workers agree on a fact; 2) **majority** agreement, in which at least half of the workers in the group agree on a fact; and, 3) **unanimous** agreement, in which all workers agree on the fact.

Clustering Results

Figure 3.4 shows the precision, recall, and F1 score for both clustering methods. Unlike in the no clustering case from before (in Figure 3.2), when we add one more worker and take into account similarity, precision drastically improves: in both the Word and Line cluster graphs, we see precision increase past 80% compared to the 44% individual worker baseline. However, there is a drastic drop in recall in both cluster conditions, as it drops from 42% to below 10%.

Furthermore, we see that, as we increase worker group size, precision for the "any" agreement condition decreases as expected (suggesting that workers agree on both relevant and irrelevant facts), whereas recall increases (there is greater potential for any two workers to agree on a fact that is a true positive), although this percentage is still lower than the without-clustering recall seen in Figure 3.2. However, this trend changes when we look at the "majority" and "unanimous" agreement levels.

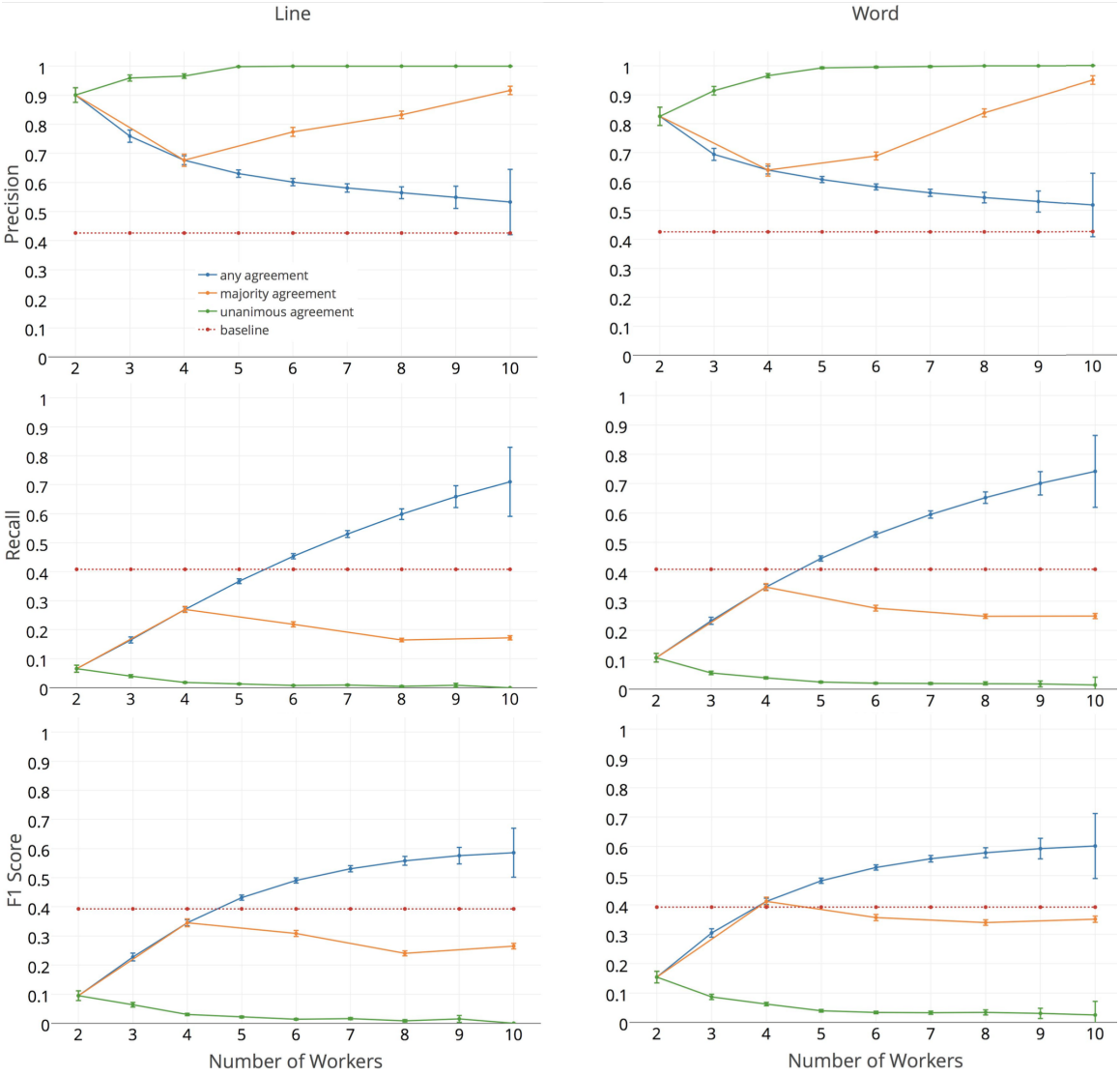


Figure 3.4: Precision and Recall by the number of workers based on Clustering the words in the worker summaries (left) and clustering the raw dialog lines (right). We find that "any" agreement performs the best for recall, but precision suffers; on the other hand, unanimous agreement leads to 100% precision, but to low recall. This implies that additional workers bring in new information with respect to recall, and tend to agree with other workers with respect to precision. We also find that worker summaries are self-consistent enough with each other to offer better clustering performance.

Both `Word` and `Line` clustering methods show similar precision and recall change trends, where precision increases but recall decreases. For instance, for "unanimous" agreement, we achieve an expected value of precision of 100% with teams of five workers. We see that "majority" agreement levels trend upward towards the "unanimous" agree-

ment case as worker group size rises past 4. This implies that the facts more workers agree on are facts that are more likely to be found in the ground truth (i.e., be relevant); on the other hand, clustering decreases the likelihood that all relevant facts are covered with agreement from workers.

Finally, we see that `Word` clustering performs better than `Line` clustering (see F1 score); this implies that worker summaries are consistent enough that they contain more information than the original dialog lines do. Despite having a small dataset compared to those typically used in Natural Language Processing applications, we are able to make meaningful comparisons between worker generated summaries and achieve better precision results.

3.8 Discussion and Future Work

Creating long term collective memory for conversational consistency has been a challenge for crowd-powered systems. However, crowd-sourcing research has yet to explore methodologies for generating user memory, specifically with a focus on long-term validity. To the best of our knowledge, Mnemo is the first attempt to develop time-independent collective memory through crowdsourced fact generation from dialogs. Here, we discuss further findings from our studies.

3.8.1 Worker Errors Are Often Not True Errors

From the worker errors characterization, we can see that precision is not low because workers inundated the system with superfluous information; rather, the categories show the variety of information saved by workers. Some facts are irrelevant because they are only for the short term (e.g. Category D), but other facts, though irrelevant for this evaluation, uncover new information not contained by the ground truth (e.g. Category E and F), which could be helpful in use cases that favor breadth of fact coverage. Future dialog systems, crowd-powered or fully-automated, can keep in mind these categories when designing collective memory curation.

3.8.2 Quality Control

In general, we avoided well-known quality control steps (like a secondary voting round on submitted answers) because of the cost they add. However, it is possible to add them with Mnemo to improve response accuracy. The goal of our experiments is to investigate workers' definition of "facts important to know" and their ability to predict relevant facts, which provides us with a baseline. Even without expensive quality control, we still see worker "errors" are often not true errors, so applications with financial bandwith may approach even better performance.

3.8.3 Validity of Facts

Our goal with Mnemo is to provide an interface for workers to predict relevant facts about a user. Worker use of facts, while an interesting behavior to explore, has been studied in prior work enough to show it to be possible *if* we could generate such content (which is our goal here). We are interested in how workers predict relevant facts when compared with a gold standard and how automated methods can help boost performance (as aggregators). To explore usage of these facts, we would need a relatively large deployment study, which is outside the scope of this initial paper on the system itself.

3.8.4 Balancing Precision and Recall

Mnemo lays the foundation for conversational context maintenance since our system is the first attempt to address this issue. We also provide "knobs" to system builders that can be used to trade off precision and recall for particular applications. Our results show that recall-focused applications can be well supported currently (using the "any" aggregation method) and we believe that, overall, recall is probably the most important piece for a first exploration of the problem space such as this paper, which presents an enabling technology/approach. If an application decides precision is most important, it can instead use the "majority" or "unanimous" aggregation method. Finally, if an application already has an "expiration" for the long-term relevance in mind, we have seen that worker precision and recall improve a lot more with the FocusedSelect rather than the TimeSelect interface, so perhaps those applications would prime workers before fact creation. We contribute to improving accuracy, but hope that our work sparks future work that further improves accuracy in this and other use cases.

3.9 Conclusion

Prior work has shown that crowd-powered dialog systems are effective at holding conversations with end users and has shown that crowd workers, when presented with a small set of concise information, can recover context from prior conversations and effectively use it to guide future interactions. However, we are the first to present a system for capturing concise, relevant set of information. Our work demonstrates that workers can do an effective job of identifying relevant facts individually, and vary enough in what they identify to perform better when combined into groups. Future work may explore how to leverage this curated information in supporting interaction beyond a conversational context. For example, this context can be used to guide automatic recommendations, or in deciding which resources to provide workers who are helping to complete a user-supporting task.

Chapter 4

Support in the Form of Assistance: Natural Language Grounding for Objects in 3D Point Clouds

In the previous chapter, we find that, although nudging is helpful for helping tease out the latent mental models across workers when it comes to saving facts from chats, it still does not help prevent annotation variability, leading to difficulties in aggregating across worker annotations. In this chapter, we address RQ2 and introduce support in the form of *assisting* the user with their annotation process itself. With this change, we develop EURECA, a mixed-initiative crowd-powered system that leverages non-expert human workers to annotate objects in 3D scenes on the fly. We show that worker effort level is reduced with the coordination provided by the interactional slingshot support, as EURECA can achieve high precision (84%) and recall (92%) while keeping latency on par with fully-automated methods (26.5s/object in group scenarios), and evaluate the system's effectiveness in case studies that mimic real-world scenarios.

Conference: *EURECA: Enhanced Understanding of Real Environments via Crowd Assistance*, In AAAI Conference on Human Computation and Crowdsourcing (HCOMP), 2018.

Coauthors: Jinyeong Yim, Karthik Desingh, Yanda Huang, Odest Chadwicke Jenkins, Walter Lasecki

Collaborations: Laboratory for Progress (University of Michigan)

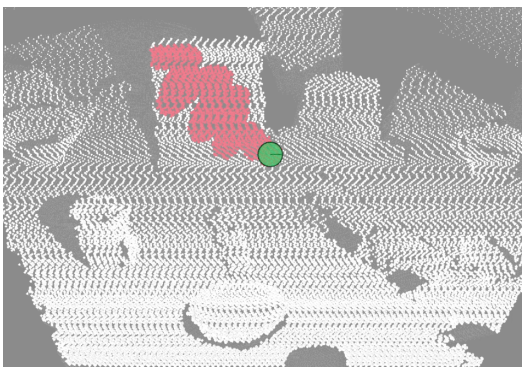
4.1 Motivation

Autonomous robots capable of fulfilling high-level end-user requests could revolutionize in-home automation and assistive technology, potentially improving access to the world for people with disabilities, providing a helping hand, and enabling more complete on-demand access to remote physical environments. Yet, robots' ability to identify objects in diverse environments, particularly for objects in settings that have not been previously encountered, remains a barrier to creating and deploying such systems in the wild. Existing 3D computer vision algorithms often fail in new contexts where training data is limited, or in complex real-world settings where scene contents cannot be fully specified in advance. Furthermore, supporting natural language (NL) interaction with end users introduces the significant additional challenge of associating linguistic information with visual scenes (e.g., to identify the target of a request).

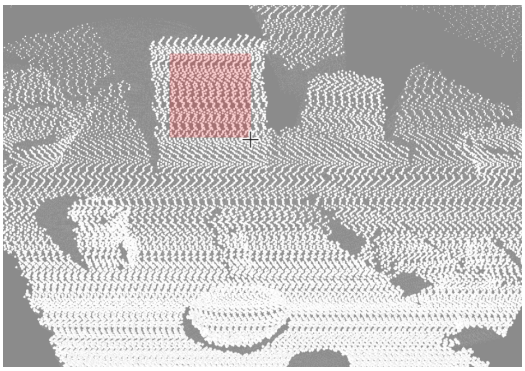
We leverage real-time crowdsourcing to create EURECA, a system that helps bridge the gap in understanding between visual scenes and the language used to describe the objects in them in order to make systems that can robustly operate in real-world settings possible.

Although existing approaches provide useful selection UIs, none solve the NL resolution problem on the fly. Instead, they are systems for offline segmentation and data creation. Further, a majority rely on high quality segmentations or classifications from automated systems, which we do not assume is available to EURECA due to our focus on novel objects and settings. Our solution provides near real-time segmentation based on an NL query even in domains where references and objects may be completely unknown to the system (i.e., no available training data).

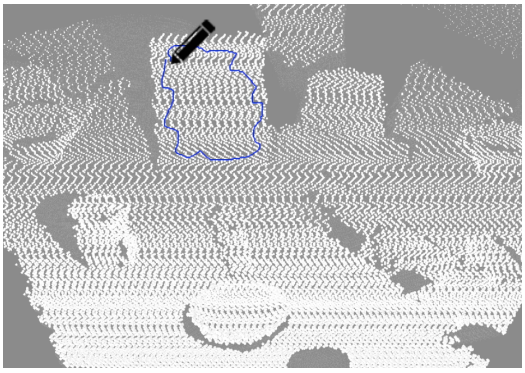
As an example, imagine a scenario in which an in-home assistive robot routinely fields requests to pick up or move different household objects. The robot is trained to carry out these tasks and can rely on a wealth of training data available for most objects. However, if the user asks for a newly introduced object (e.g., something they recently purchased) to be retrieved, the robot may fail to complete the requested task using automated methods alone if it does not understand the reference to a new object. EURECA helps robots overcome such failure modes by leveraging on-demand crowds of human workers to collaboratively segment and label unfamiliar objects based on an NL request and a 3D point cloud view of the current scene. Within tens of seconds, the robot understands which object is being referenced and can immediately carry out the request, as well as increase the setting-specific training dataset to improve future automation.



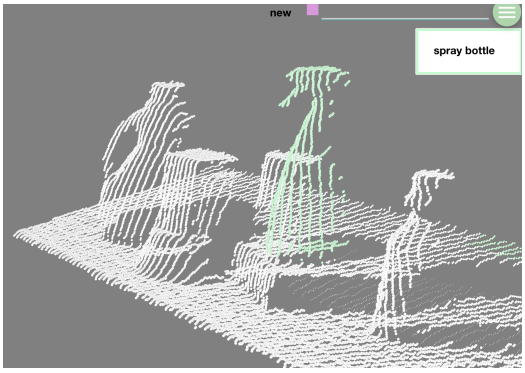
(a) `Paint` tool



(b) `Region` tool



(c) `Trace` tool



(d) An end-to-end case study

Figure 4.1: (a)-(c) Intelligent and collaborative selection tools in EURECA. Crowd workers can choose from three tools to select a group of points for segmenting and labeling 3D point clouds. EURECA takes initiative to automatically augment initial user selections; unintentionally selected points are "filtered" out, and missed points are "filled" in, making final worker selection easier, faster, and more accurate. (d) The scene used in our robot case study, as well as a real crowd worker's annotation of "spray bottle."

While this presents a powerful way to make robots more robust in real-world settings through on-the-fly training, using human workers as part of the sensing process—especially in non-public spaces, such as home or office settings—introduces privacy concerns. Workers may be able to identify individuals, observe information on documents or whiteboards, and more. To address this, we designed EURECA to be effective even with only depth information (without an RGB image overlayed). This both helps preserve privacy and makes EURECA compatible with a wider range of sensor technology currently used on robotic platforms (e.g., LIDAR sensors).

4.2 Contributions

By combining the machine's ability to precisely select content with people's ability to understand scene semantics, EURECA presents a *hybrid intelligence* approach to 3D annotation—allowing it to benefit from as much automation as possible, while using human intelligence to fill in the gaps. To improve crowd workers' ability to quickly and accurately select objects in a 3D scene, EURECA takes steps towards a mixed-initiative workflow, allowing the crowd to work collaboratively with the system to refine selections for segmentation. Based on initial worker selections, the system automatically infers points to augment those selections (which can even draw on existing 3D vision approaches), with workers able to progressively correct those automatic refinements.

EURECA comprises an interface for selection and scene manipulation (allowing workers to rotate, pan, and zoom) using a series of selection tools, and automated assistance for selection refinement. To further reduce segmentation task latency, EURECA recruits multiple workers on-demand to synchronously complete tasks faster than any lone worker. Coordination mechanisms are provided to prevent redundant or conflicting worker effort.

While EURECA's approach requires no prior human or machine training (and can actually generate training data), it is possible to integrate the output of computer vision approaches for even better results. In fact, we explicitly avoid relying on preprocessing because we target settings where automated systems have already failed. However, if output from vision approaches exist (e.g., preprocessed clusters, labels, etc.), EURECA can use that to make selection easier for workers. This reduces the effort needed from crowd workers and, over time, enables our approach to smoothly transition towards full automation as 3D computer vision methods improve and as more data is collected.

We validate our approach on scenes from an established, publicly-available dataset [61] and demonstrate that our annotation baseline tool, `Paint`, leads to per-object segmentation times of 85 seconds for individual workers. From this base approach, we then show that our machine-augmented selection tools, `Region` and `Trace`, which infer final selections based on worker input, further decrease segmentation times by 32%, while increasing object precision and recall by 5% and 9%, respectively. Next, we demonstrate that our techniques for supporting coordination among workers lead to speedups that increase with the number of contributors, further decreasing the average time it takes to annotate objects to just 26.5 seconds each.

We conclude with a demonstration of the end-to-end EURECA system with a Fetch robot[1] that is able to respond to a user's natural language command and accomplish a grasping task. Our work will allow automated object recognition systems to be trained on the fly, creating a seamless, reliable experience between end users and robots. Specifically, we contribute the following in this paper:

- **EURECA**, a mixed-initiative crowd-powered system that leverages non-expert human workers to annotate objects in 3D scenes on the fly.
- **Mixed-Initiative Annotation Tools** for EURECA that help coordinate multiple simultaneous workers on an annotation task to further reduce latency.
- **Validation** that EURECA can achieve high precision (84%) and recall (92%) while keeping latency on par with fully-automated methods (26.5s/object).

4.3 EURECA: Collaborative 3D Tagging

We build on this related work to recognize objects in settings where automated approaches fail or lack sufficient training data. EURECA recruits crowds of workers on demand, then takes initiative to augment user selections, after which users can further correct updated selections. In this section, we describe EURECA's architecture, including the mixed-initiative workflow, worker UI for interacting with the point cloud, automated support to refine users' object selections, and annotation tools for collaborating with remote workers.

4.3.1 Web-Based Annotation Tool

EURECA presents workers with an interactive visualization and annotation tool for 3D point clouds (Figure 4.2) built in JavaScript using the ThreeJS library[2]. Full 3D point clouds can contain more datapoints than can be rendered at interactive speeds (e.g. a Kinect generates over 300,000 points). To address this, EURECA keeps only every eighth point for a final point cloud size of ∼35,000 points. On page load, crowd workers are shown the point cloud and asked to select and label objects mentioned in a natural language query. Workers can adjust their view of the 3D space using camera controls that let them easily

[1] http://fetchrobotics.com/platforms-research-development/
[2] http://threejs.org

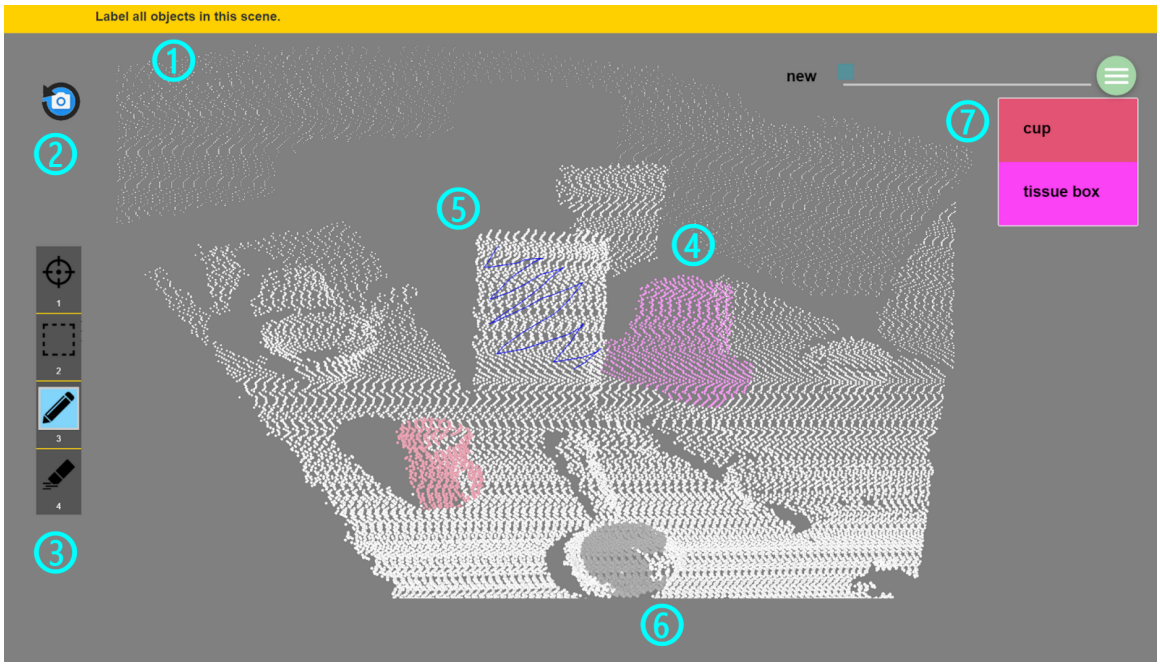


Figure 4.2: EURECA's worker labeling interface. A typical view includes: (1) Natural language query issued by the end user.; (2) Camera controls allow a worker to easily zoom, pan, and orbit around in the scene; (3) Collaborative selection tools make it easy to select objects (as well as undo any erroneous selections); (4) An object already segmented and labeled by the worker; (5) An object that is currently being selected by the worker; (6) Gray points indicate a remote worker's real-time activity for collaboration tasks; (7) Labeling interface for associating the NL query to object segments.

---

pan, zoom, and orbit a scene. Workers see color highlights of the points they select. To select points, workers are provided with the `Paint` tool (Figure 4.1a) which works by dragging an adjustable-size cursor over the 3D points in a continuous motion (akin to "painting" on the 3D canvas).

To help the crowd select 3D objects more efficiently, we create two additional tools, `Region` and `Trace`. The `Region` tool (Figure 4.1b) allows workers to drag-select a rectangle over a region of interest. Once the click-and-drag event is finished, points that are inside the 2D rectangular region are selected by ray casting a shape matching the worker-indicated region and including all intersected points. For objects that are harder to select with just the `Region` tool—e.g., objects with a more organic shape, or objects that are partially occluded—workers can use the `Trace` tool (Figure 4.1c). Unlike `Region` tool, `Trace` allows workers to draw a free-form region of interest. The points enclosed within the region are highlighted using a ray casting method similar to that used for the `Region` tool.

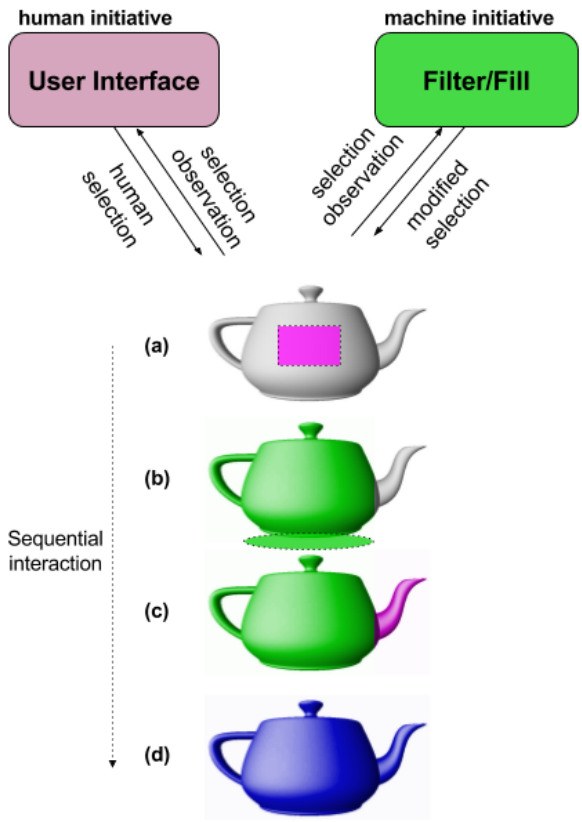


Figure 4.3: EURECA's iterative, mixed-initiative approach. In (a), the user makes an initial selection (magenta); in (b), the machine observes the selection, takes initiative, and modifies it to fill the rest of the base (green); in (c), the user sees that the system overfilled points (dotted oval), and retakes initiative to clean up that excess selection, and then selects the tea pot's spout; finally, in (d), the machine enters a "no-op" state since there is no more filter/fill to be had.

4.3.2 Mixed-Initiative Workflow

Selection using the tools described above will not always result in perfect object boundaries. Automated refinement is one way to overcome this limitation. A user's ultimate goal of fine-grained selection of a novel object can be thought of as a two-part approach: there is the user's *latent intent*, which involves wanting to perform a fine-grained segmentation of an object (the "goal state"), and then there is the user's *expressed intent*, which involves using the tools in the system. A user's expressed intent is often limited by the selection tools' capabilities (there will be imperfection in this process).

One approach would be to provide smarter and more capable tools to the users. However, direct manipulation using selection tools might not always help achieve the goal state because the user's latent intent is unknown to the system. Our key insight into overcoming this limitation is to instead use a mixed-initiative workflow [43, 45]. Within this mixed-initiative framework, users can now *collaborate* with the system's initiative to interactively refine the machine's selection (by taking back initiative). Based on the initial user selections, EURECA takes initiative to **filter** out points that were unintentionally selected by workers, and **fill** in points that it believes were missed in the initial worker selection. As users make repeated point selections for the same object, EURECA starts to better understand the user's high-level (latent) intent that is being expressed through low-level selection actions, thereby building a shared context to achieve the goal of fine-grained segmentations (Figure 4.3). This lets EURECA's automated selection methods iteratively refine the current selection state in tandem with the worker, thereby informing future selections.

As an example of where this mixed-initiative approach is beneficial is in cases where users might not always know the exact object boundaries in the 3D scene. If they mistakenly lump two objects into one selection (if, say, their viewpoint hid the boundaries), a confident system can take initiative, jump in, and adjust the filter / fill process. This mixed-initiative approach, then, can let both the user and the system collaborate effectively.

Point-Filtering ("Filter")

To infer points to be removed from a worker's initial selection, EURECA uses a combination of two methods: filtering first by performing outlier detection, and then finding the selection of interest using the Kernel Density Estimation from an off-the-shelf JavaScript library [22].

Standard outlier detection, in which points that are significantly distant from the bulk of the selected points are removed, is first performed. This method is not resilient to filtering out points that are within the distance threshold, but still clearly belong to another object (e.g., if there are two objects that occlude each other, a worker's wayward selection can catch points from both objects). Outlier detection is augmented with the KDE method. (We note that EURECA's architecture supports any method that takes in an initial user selection and outputs a refined segmentation, and so use KDE as one such method.)

EURECA builds a density curve of points from the camera's line-of-sight to the initial selection set, based on camera distance. Since the goal is to filter out erroneous selections within the user's line of sight, the algorithm splits on the first local minimum and discards points outside the first cluster. This method filters out points that are behind the object that was "intended" to be selected. A threshold learned from training data is used to avoid splitting off and selecting a cluster that contains only a few points.

Selection-Completion ("Fill")

For fill, there is a higher likelihood that points close together belong to the same object. To infer points to add to the initial selection set, EURECA uses a label propagation-based method that is similar to "flood fill" tools in modern graphic editing software (the simplest example of which is "bucket fill" in Microsoft Paint and similar applications).

For each unselected point, EURECA first calculates a constant influence value from a selection point to all points within its neighborhood. Using the kd-tree structure allows for rapid calculation of each point's distance relations to all its neighbors. This is augmented with a term that takes into account how far away this unselected point is from the selection center. Since we assume a worker's initial selection lies mostly within their target object, the second term helps prevent runaway propagation, as points that are too far away will be less likely to be filled in. An inclusion threshold is used to determine which points to add to the final filled-in selection set. A version of the Brushfire algorithm [17] is used to estimate the influence on subsequent points. In practice, every point influences its neighbors within a radius that is proportional to the average distances between neighborhoods of points. To slow down the effect of the "brushfire," EURECA adds a penalty on the length of the propagation chain. An inclusion threshold is again used to determine points that are added to the final selection set.

4.3.3 Collaboration and Scaling with Crowd Size

Moreover, EURECA facilitates coordination between multiple workers via real-time feedback on the selection and labeling of synchronous workers. Because we have little to no information about a given scene in our problem formulation, it is difficult to direct workers to non-overlapping parts of the scene to avoid redundant work. Lasecki et al. previously explored using "soft locking" in Apparition [68], where workers manually placed markers to signal to others that they were contributing in the 2D scene's physical location. We adapt this idea by automatically providing real-time feedback on what other workers

are marking via highlighting. This approach, while intuitive, is novel in crowdsourcing systems and generalizes to broader classes of real-time coordination problems. EURECA uses Meteor[3] to create a shared tagging state that allows remote events to be synchronized between workers' local views.

4.4 Evaluation

Making interactive robotics applications possible via crowd-augmented sensing requires a combination of speed and accuracy. In the previous section, we described EURECA's architecture. In this section, we introduce the experiments we use to validate the efficacy of this architecture.

4.4.1 Recruiting Crowd Workers

We recruited 78 unique workers with a minimum approval rating of 95% from Amazon Mechanical Turk [5]. For each task, we paid at an effective hourly rate of $10 per hour, along with a built-in bonus amount for successfully completing a multi-stage tutorial. Once workers pass the tutorial, they are routed to the main task in which they use EURECA to respond to the posted query (e.g. "Select and label the dinner plate").

4.4.2 Point Cloud Dataset

Our evaluation uses scenes from the RGBD Object Dataset [61], which consists of color and depth images of naturalistic household and office scenes. Because we wish to explore sensing modes that preserve user privacy, we use only the depth images to generate a 3D point cloud. We selected five scenes with enough diversity in object type, clutter, and orientation to validate object reference resolutions and crowd segmentations (Figure 4.4). To create the ground truth for evaluation, two researchers carefully annotated the various object segments for each scene.

[3] https://www.meteor.com/

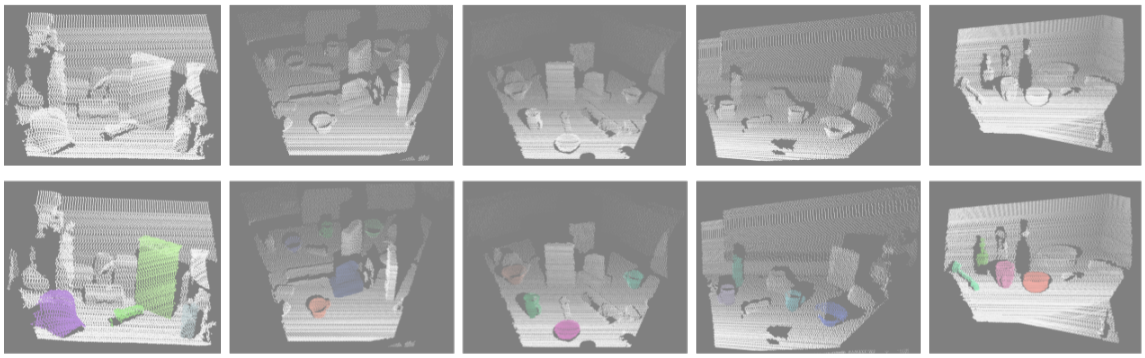


Figure 4.4: Point clouds used in worker studies. Workers are instructed to segment common household objects. The exact natural language query differs for each scene. For each scene, non-trivial camera movements are required to overcome object occlusion, shadow effects, and orientation in order to identify objects.

4.4.3 Measures

We evaluate worker performance using four measures: latency, precision, recall, and the F1 score. We measure latency in terms of the entire session's duration, from understanding to segmentation to NL annotation, and not simply the time spent selecting the object. This includes time to understand the scene ("perception") and time to select objects ("selection"). Factors that impact perception include dealing with occlusion, understanding an object's orientation within the scene, and recognizing how distinct an object is by its shape. Factors that impact selection include how easily separable the objects are, as well as how difficult it is to select the object's shape. Therefore, latency will always be longer than time spent segmenting objects.

We divide latency by the number of objects we detect that the worker has labeled. This normalization on a per-object basis lets us compare across scenes with different numbers of objects. We then automatically align worker selections to the best-fit ground truth objects to calculate precision and recall. We report precision and recall for both objects (important for object recognition) and points (important for grasping / motion planning). The F1 score (harmonic mean of precision and recall) gives a combined accuracy measure. We perform paired two-tailed t-test to measure significance.

4.4.4 Study Conditions

We focus on evaluating 1) EURECA's overall efficacy, 2) the effect of our selection tools (with automated refinement) on how quickly and accurately workers can segment objects, and 3) the impact of workers collaborating in teams.

Study 1: EURECA's Effectiveness. To measure the overall effectiveness of EURECA in enabling workers to segment and label objects in 3D point clouds, we identify at the object level how many object instances were correctly identified by the workers. Across the five scenes, there are 21 unique objects (four scenes with four objects each, and one scene with five objects).

We evaluate this recognition in terms of precision (how many objects did the worker correctly identify?) and recall (did the worker correctly identify all the requisite objects in the scene?). We treat each iteration of the scene as a new data point, as there is a chance that a worker might not recognize an object when they see the scene the first time around, but do end up recognizing it the second time they see the scene.

Study 2: The Effect of Automatic Refinement on Selection. To study the efficacy of EURECA's initiative when automatically refining user selections, we task workers with segmenting and labeling objects in the same scene twice: once with only the `Paint` tool enabled (PaintMode) and then once with all tools at their disposal (ToolsMode), presented in a randomized order.

Study 3: Collaboration in Teams. Next, we want to demonstrate that EURECA's collaborative features enable it to efficiently scale with the number of workers available. We select two scenes with a total of nine distinct objects that needed to be identified. Workers are recruited to a "retainer pool" whenever EURECA is running, and can be directed to a task within one second of a query that the system does not understand arriving. By varying the team size from one to three workers, we can investigate the efficacy of EURECA in enabling worker coordination and collaboration when performing multiple selections.

4.5 Results

In this section, we describe the experimental results of related to the core EURECA system, the mixed-initiative tools that support workers, and the benefits of collaboration.

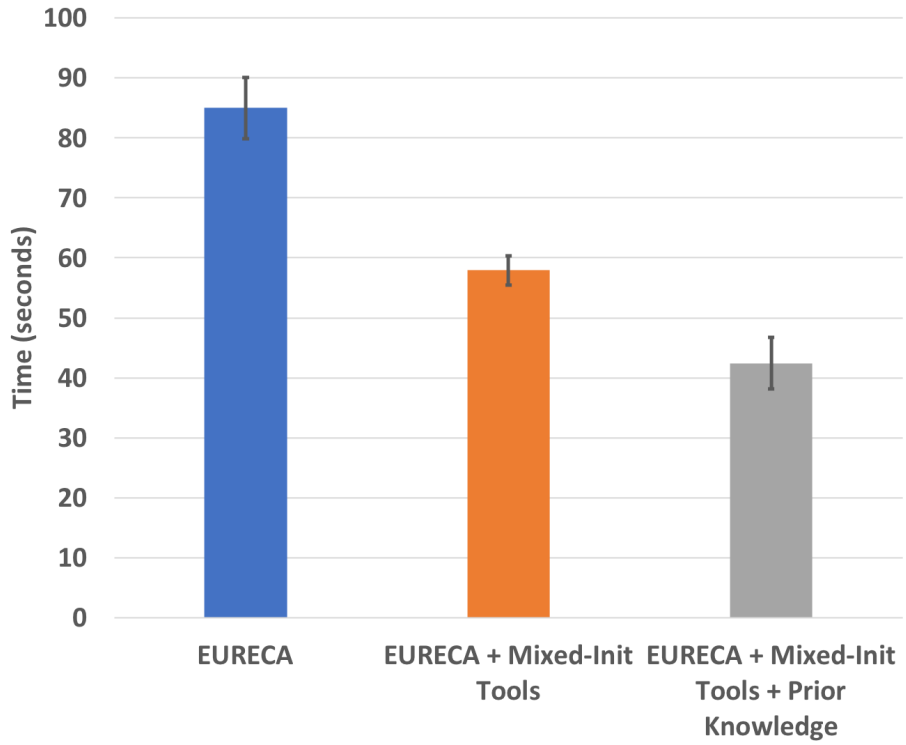


Figure 4.5: Overall latency per object among the crowd workers in EURECA's various iterations.

4.5.1 Study 1: EURECA (It Works!)

We recruited 34 workers to use EURECA using only the `Paint` tool. We dropped one outlier whose task duration was more than 3σ from the mean. The remaining 33 workers were distributed across the five scenes: three scenes had seven workers each, and two scenes had six workers each.

We find that the average total time to task completion (both perception and selection for the never-before-seen scene) for all 33 workers, when normalized on a per-object basis, was 85 seconds ($\sigma = 56$s; $p < 0.005$), with 99.6% object-level precision (only one false positive), and 93.9% recall. Of the total of 17 object instances were missed (not recalled) by workers, 10 were completely missed and 7 were combined with other objects into one label. As mentioned earlier, because scene understanding involves both perception and selection factors, we see scene latency times range from 65s to 92s on average.

4.5.2 Study 2: Mixed-Initiative Selection Tools

Although 85 seconds latency per object segmentation and labeling already allows for on-the-fly understanding of novel 3D scenes, we seek to improve this further with EURECA's mixed-initiative tools. Does the progressive refinement of selections help speed workers up?

For the 33 workers, as we see in Figure 4.5, when ToolsMode is enabled, we see a 35% relative improvement in annotation speed to 58 seconds ($\sigma = 28$s). Additionally, we observe an improvement in the average precision and average recall: precision improves from 0.82 to 0.86, whereas recall improves from 0.82 to 0.90).

Worker Improvement Over Time

In addition to worker speedups from the mixed-initiative tools, we want to know if workers learn with repeated exposure to the conditions in Studies 1 and 2. We further break down the performance on tasks accounting for the order of task conditions (i.e., whether workers used PaintMode or ToolsMode first). There were 19 workers who used PaintMode first and 14 who used ToolsMode first. Workers using PaintMode first completed the first task in 87.4s (followed by a second task using ToolsMode completed in 61.1s). Workers using ToolsMode first completed the first task in 55.6s (followed by a second task using PaintMode completed in 81.8s). Comparing across both orders, we find that workers improve over time when they use EURECA.

Moreover, we note that 10 workers (30.3%) did not use ToolsMode when they had the option available, which means that they used the `Paint` tool for all object tagging. With this in mind, we can focus specifically on those workers who used at least one of our ToolsMode selection tools. For these 23 workers, when in the ToolsMode condition, we see a statistically significant 36% improvement ($p < 0.02$) in time taken to tag objects when compared with the time taken in the PaintMode condition.

Leveraging *a priori* Clustering Information

With EURECA, we obtain performance improvements when we add our new selection tools with the system initiative to refine user selections. However, active research is being conducted on devising systems that can segment out objects or surfaces in visual scenes. Such information can provide our tools with improved knowledge and understanding of the scene. In fact, for all new elements that have never been seen before, this kind of automated segmentation is the best that can be done. But, even though we can

delineate it in the scene, we still require the proper NL annotation for it. Since both of EURECA's selection tools have the ability to integrate results from any off-the-shelf segmentation algorithm, can performance be improved if our envisioned robot has such *a priori* understanding of its environment?

To test this hypothesis, we recruited 10 workers from Amazon Mechanical Turk and ran the same experimental setup as seen in Study 1 with one of our scenes. We use perceptual grouping of RGBD segments to form object cluster information [91]. When we take the average worker performance across all of ToolsMode *with* clustering information available, we find a further 37% improvement in speed when compared with the average across all of ToolsMode *without* clustering information. Therefore, if prior clustering knowledge exists, EURECA's selection tools can leverage that information to further reduce per-object tagging time (Figure 4.5).

4.5.3 Study 3: Collaboration Leads to Lower Latency

To understand the ability of teams of workers to complete the annotation tasks, we recruited 24 workers to create four different teams for four scenes. Each team had to segment between four and five objects. We find a large decrease in segmentation time required as we add more workers (Figure 4.6). Individual workers (teams of size one) took on average

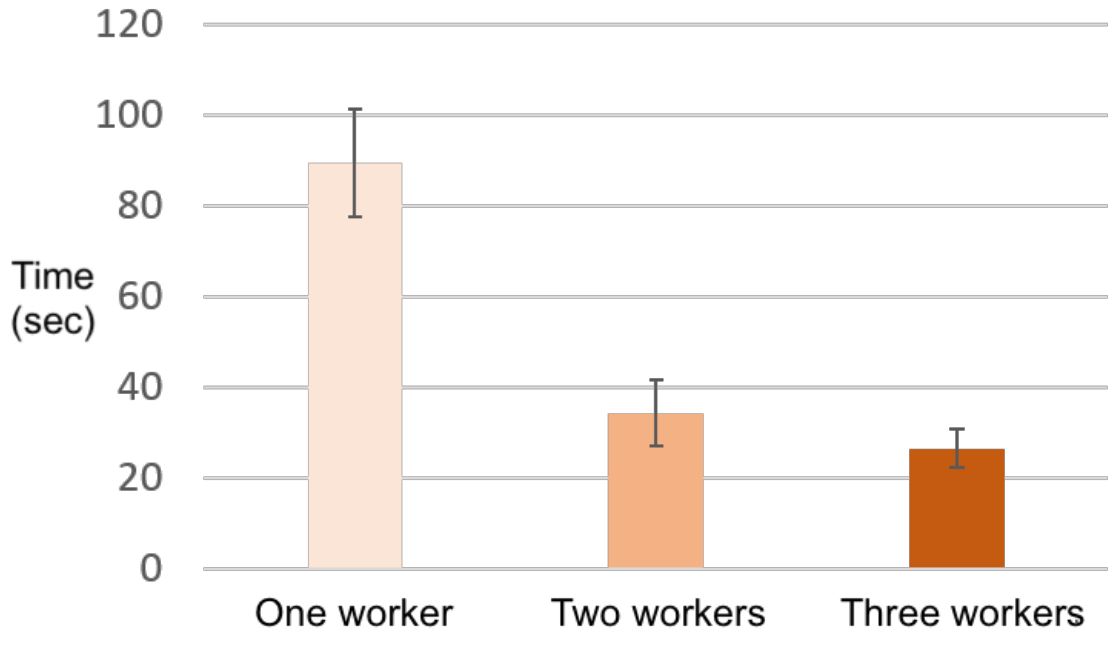


Figure 4.6: Latency per object in a collaborative setting.

89 seconds ($\sigma = 24$s) to segment the objects, with an overall precision of 0.96 and an overall recall of 0.99 (F1 score of 0.97). When we add one worker (teams of size two), we see a 62% relative decrease in time taken to 34 seconds ($\sigma = 15$s); however, we also see a 14% decrease in precision along with a 0.2% increase in recall (F1 score of 0.91). Finally, for teams of size three, we see a further 22.5% decrease in time taken for segmentation and labeling to 26.5 seconds ($\sigma = 8.4$s), with a relative decrease in precision of 0.12%, and a relative decrease of 7.3% in recall (F1=0.88).

These results suggest that having workers collaborate with each other offers immediate speed benefits; increasing team size to two leads to a drastic reduction in latency, but at the expense of a decrease in precision. However, precision losses seem to stabilize when an additional worker is added.

During one of the trials, we observed that one team of three did not complete the task because they entered a conflict state in which an error confused all workers into tagging erroneous objects. This suggests EURECA still needs to find more effective ways of enabling explicit worker coordination, especially to rectify mistakes. Future work may explore addressing such conflicts by automatically changing a worker's camera view such that no one worker is looking at the same part of the scene during collaborative tasks.

4.6 Case Studies

We have experimentally validated that EURECA enables crowd workers to quickly identify and accurately select, segment, and annotate objects in 3D point clouds, all with near real-time latency. With the novel `Paint` tool, we see speeds of 85 seconds, which we are able to reduce to a best-case scenario of 26.5 seconds with teams of 3 workers. In this section, we explore some case study scenarios to better evaluate EURECA's performance for real life applicability.

4.6.1 Case Study: End-to-End Test with a Robot

We are using the Fetch robot, a mobile manipulation platform mounted with an ASUS depth camera to sense the environment. For this case study, we assume that the robot has bounding boxes and training data for numerous objects. Provided the object locations in the point cloud, the robot uses handle grasp localization [108] and MoveIt! [106] (a motion planning library) to manipulate an object. However, when a new object—a *spray bottle*—is introduced, the robot has no way of detecting it, so it places an on-demand re-

quest to EURECA. In our case study, the robot successfully picked up the spray bottle—of which it had zero training data on—based on the crowd-generated annotation. Our case study validates that the precision obtained from the crowd's segmentation and annotation using EURECA is enough to enable object manipulation, which is typically seen as a harder task than object annotations for room navigation (as manipulation requires higher segmentation accuracy).

Figure 4.7: An example case study where the Fetch Robot successfully picked up a spray bottle based on an Amazon Mechanical Turk worker's annotation using EURECA.

Figure 4.8: An colorized (RGB) version of the point cloud that is seen in Figure 4.2. Even though RGB helps to visually differentiate between the objects, we did not see any significant improvements in annotation speed and accuracy.

4.6.2 Case Study: Using RGB Color Information

If the lack of RGB color information constraint were relaxed, would that improve worker performance? To investigate this, we repeat Study 1, but workers now see the RGB point clouds. Figure 4.8 shows a point cloud with RGB information: in this figure, it is easy to visually separate the objects. However, after accounting for two outliers, we find that with N=25 workers, segmentation is 5.9% faster with PaintMode (80s), with a 4.9% gain in precision and 2.4% gain in recall. Worker feedback seems to shed some light on this result, as workers found it difficult to delineate the selection colors from the point cloud colors. Future work could look into ways of toggling point cloud colors to make the selection object stand out more clearly.

4.6.3 Case Study: Deformable Objects

We study EURECA's performance on a custom scene with deformable objects where the task is to identify a scarf. Compared with the segmentation using an off-the-shelf region growing algorithm in PCL [97] in Figure 4.9, which erroneously breaks the scarf into multiple regions, crowd workers are able to properly segment the scarf as one object.

We can see that an off-the-shelf region growing algorithm erroneously identifies multiple regions for the scarf, whereas crowd workers are able to correctly segment the scarf using EURECA. However, because deformable objects can be complex, workers need contextual information to disambiguate between objects if confusion arises (e.g., if the table were to consist of only scarves, then picking out the correct one would not work). Perhaps crowd workers can separate out the individual scarves and the robot can rely on clues in the natural language query to properly annotate the scarf (e.g., "the middle one").

4.6.4 Case Study: Labeling in Open-Ended Queries

We have seen from our studies that workers have extremely high precision and recall when it comes to identifying objects when given directed queries (e.g. "Pick up the bottle"). We want to know how well people can identify and label objects in an unbounded, or open-ended, query setting. For this case study, we recruited 10 workers (two outliers did not complete the task before submitting the HIT) and asked them to label all objects in the scene. We picked seven unique objects that the workers should have selected. Since each worker sees the scene twice, we have a total of 7 * 2 instances of objects per worker; with 8 workers, we have a total of 112 object instances that need to be recognized.

Since this is a visual scene understanding task, we accept an object as having been "recognized" if its object label matches a version of the ground truth's label (e.g. "cup" and "mug" map to the same abstract label). We find the overall object instance recognition precision and recall to be 0.44 and 0.52, respectively. However, based on the default camera view, there were two separate objects, a bowl and a flashlight, that seemed to resemble a saucepan or a frying pan. Five of the 8 workers mistook these two separate objects as a single object. When we account for this mislabeling, as well as other mislabels, we see precision and recall climb to 0.68 and 0.80, respectively.

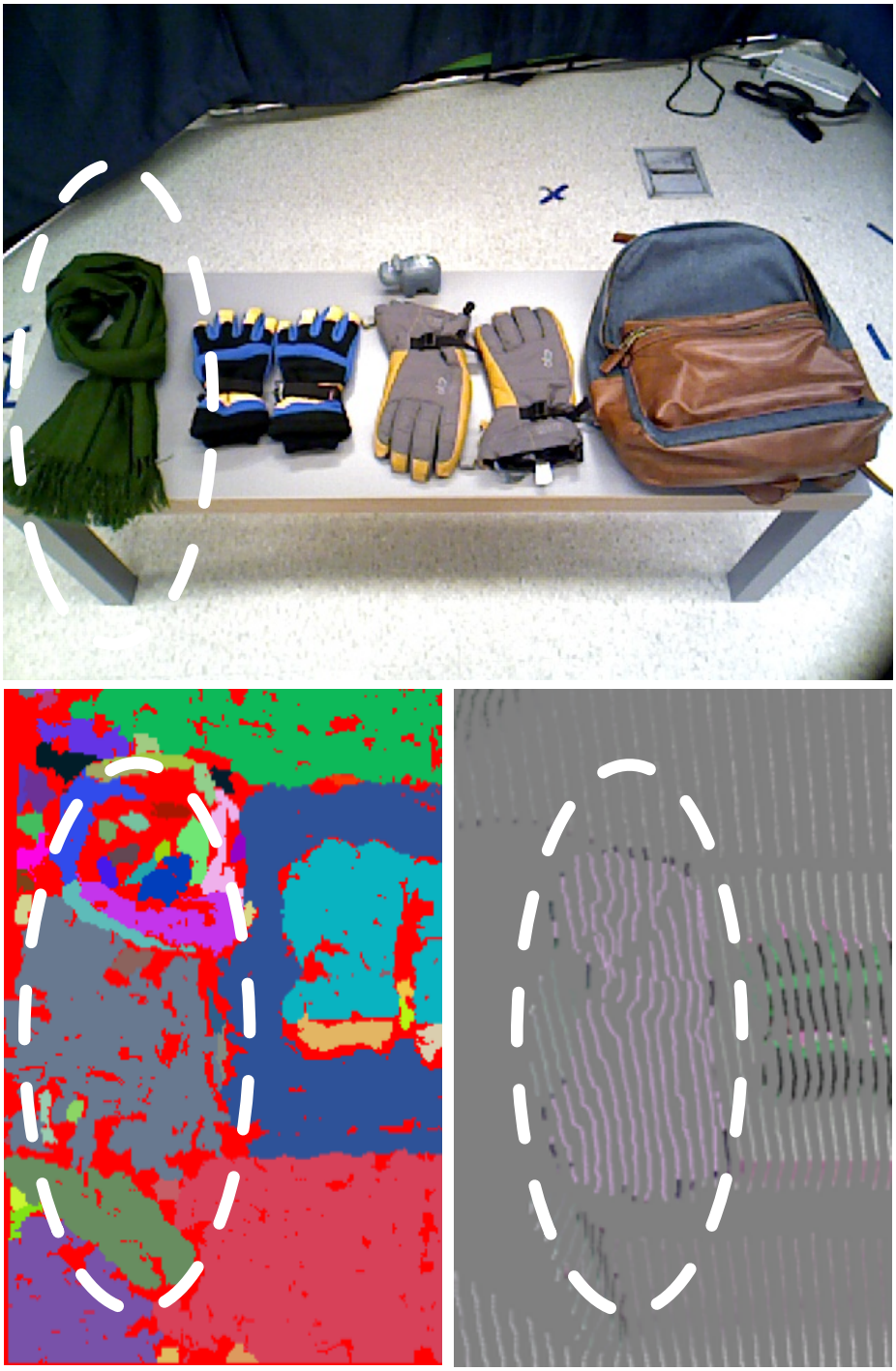


Figure 4.9: For a deformable object (Top: green scarf within the dotted white oval), PCL's region growing erroneously segments the scarf into multiple distinct regions (Bottom Left), whereas an Amazon Mechanical Turk worker is able to correctly segment and annotate the scarf (Bottom Right).

There were a total of 20 false positive instances, which is a 15.15% false positive rate. However, 30% of the false positives (6 object instances) were objects that the crowd recognized that were out of scope for the task (e.g. one worker broke down a "tissue box" into constituent "box" and "tissues"; another worker identified a "cap" that was hidden behind a bowl).

Finally, when asked to find a particular object (e.g., "Pick up the cup"), it is possible that the scene has multiple objects of the same type. If there are other instances of said object, we imagine the robot will prompt the end user for a clarifying request (e.g., "Pick up the cup nearest the fridge"). We do find that workers annotate all objects of a particular class (i.e., all cups or all bowls).

From this case study, we can see that workers are able to recognize objects even without direct prompting, though direct object mentions boost the ability to perceive objects. Workers also seemed to recognize objects better the second time they saw a scene, including objects that were not part of the ground truth. This suggests EURECA could help robots to get a more general understanding of their in-home environments.

4.7 Limitations and Future Work

We show that EURECA effectively leverages non-expert crowd workers to annotate 3D scenes in as little as 36s per object for individual workers, and as little as 26s per object when workers collaborate in teams. However, collaboration currently only works in parallel and builds on the "soft-locking" idea seen in [68]. Future work can explore ways of having multiple workers select the same object without conflicts, making the workflow fast enough for natural and continuous interactions with robots.

During our collaboration tests, we did not observe any social loafing behaviors in our tests. We were focused on the functionality of the end-to-end system, and any loafing effect was minimal enough that it did not prevent a significant improvement in the performance of groups over individuals. With further instrumentation, future work can study worker behavior in detail. While this is an interesting problem, studying it sufficiently is beyond this dissertation's scope.

Furthermore, even when preserving privacy by removing RGB information and downsampling the point cloud by keeping only 10% of initial points, workers are still able to correctly identify common household items with high precision and recall. However, in addition to the scenario we saw in the case study where it is hard to delineate objects into constituent ones (e.g. the saucepan), unique objects could prove problematic for EURECA.

Unique objects, such as say a clay dinosaur, would be hard to identify from just the point cloud alone. Indeed, clutter and other scene properties (e.g., camera capture angle) can significantly affect the ability for anyone, computer or human, to both perceive and select objects in 3D scenes.

Our goal was to demonstrate that crowds could be used to segment and annotate objects in real time in "tractable" scenes, which we explored using a common 3D vision dataset in the literature that contains images feasible for highly-trained vision systems to recognize with reasonable accuracy. As a result, EURECA's strength lies in dealing with objects that are familiar to the average worker. Future work could explore how to overcome these bounds by devising workflows that selectively relax privacy constraints.

Finally, EURECA's ability to deal with novelty makes our approach especially relevant to mobile robots. As these robots enter new environments, the likelihood of them encountering unknown and novel objects increases. For these settings, the robot can place on-demand requests to EURECA. We could then take advantage of the class of point-tracking algorithms to map the already-annotated region as the robot moves around its environment. Furthermore, EURECA's fill algorithm can be used to incrementally update this annotated region as more of the object is uncovered (e.g., occlusions disappear as the robot moves around). Future work may address how to introduce approaches that reduce latency and further reduce the amount of human time required for the real-time annotations.

4.8 Conclusion

In this chapter, we present EURECA, a mixed-initiative, hybrid intelligence system that leverages non-expert crowds of human contributors to help robots identify, segment, and label objects in 3D point clouds in near real-time. This makes it possible to deploy robots that operate reliably in real-world settings from day one, while collecting training data that can help gradually automate these systems over time.

Chapter 5

Support in the Form of Guidance: A Novel Interface for Multi-Domain Conversation Disentangling

In this chapter, we explore what happens when interactional support *guides* user annotations for a task that is not only difficult, but also requires domain knowledge.

Specifically, people with domain expertise in computer science and non-experts in the form of online crowdworkers will be disentangling conversations from Internet Relay Chat (IRC) logs. The output conversations are critical to training machine learning (ML) algorithms so that these algorithms can effectively provide disentanglement in chat applications. Our ultimate goal is to explore how providing visual contexts during the annotation phase can make these annotations adaptable to multiple domains, making it easier and more effective to train ML classifiers, which in turn can be used to train chatbots and other applications.

Conference: *A Large-Scale Corpus for Conversational Disentanglement*, ACL, 2019.

Note: Only the initial part of the work discussed in this chapter was published at ACL. The rest of the chapter is unpublished work.

Coauthors: Andrew M. Vernier, Yiming Shi, Zihan Li, Jonathan K. Kummerfeld, Mark S. Ackerman

Collaborations: IBM Research

5.1 Motivation

Intelligent systems in digital environments bring us closer to achieving the longstanding goal of artificial intelligence and human-computer interaction: systems that can interact with human beings using natural language. However, finding human-human conversational data to train these systems remains a core challenge to fully automating these systems; in fact, millions of lines of human-human dialog exist on the Internet for free in the form of Internet Relay Chat (IRC) messages. However, current state-of-the-art systems face a critical challenge since they cannot easily disentangle conversations that require domain expertise. If we improve the ability for AI systems to disentangle these IRC logs into constituent conversations.

Not only can these disentangled conversations help train chatbot models, but also they can be used to train disentanglement models. These models can then be used to make log history easier for people to read through, such as allowing individual channel owners to quickly get disentanglement working for their own logs.

An important step, then, is to leverage human intelligence to annotate this IRC data that requires domain knowledge. An example of how multi-party, multi-turn conversations can be entangled, or overlap, is seen in Figure 5.1.

Our previous work in this space [60] first started as an internship project at IBM Research. We took a rule-based learning approach to automatically disentangle multiparty, multi-turn conversations taking place in Internet Relay Chat (IRC) on the #Ubuntu channel. Each rule would identify structure and patterns in the data and create a contextual thread of importance in the IRC log, such as keywords in each conversation thread. For a new utterance, rules were used to identify to which conversation that utterance belongs; in this way, the combination of the rules with the keywords-creation helped to disentangle different threads.

However, this process requires high-effort on behalf of the expert. The more rules that are created, the more complex potential edge-cases can be, or the higher chance utterances can be filtered out / otherwise not included. It is often the case that experts always have to keep updating their heuristics because new conversations can bring in edge-cases that the expert might not have thought about. For instance, Figure 5.2 shows ambiguity that is often involved in disentangling conversations that require domain knowledge.

This initial, rule-based learning approach for this task quickly becomes complex and full of edge-cases. As a result, there can be issues with precision and recall. Rather than relying on heuristics to capture high-quality annotations, instead, it is better to create interactional slingshots and rely on ML algorithms to help augment the expert's effort.

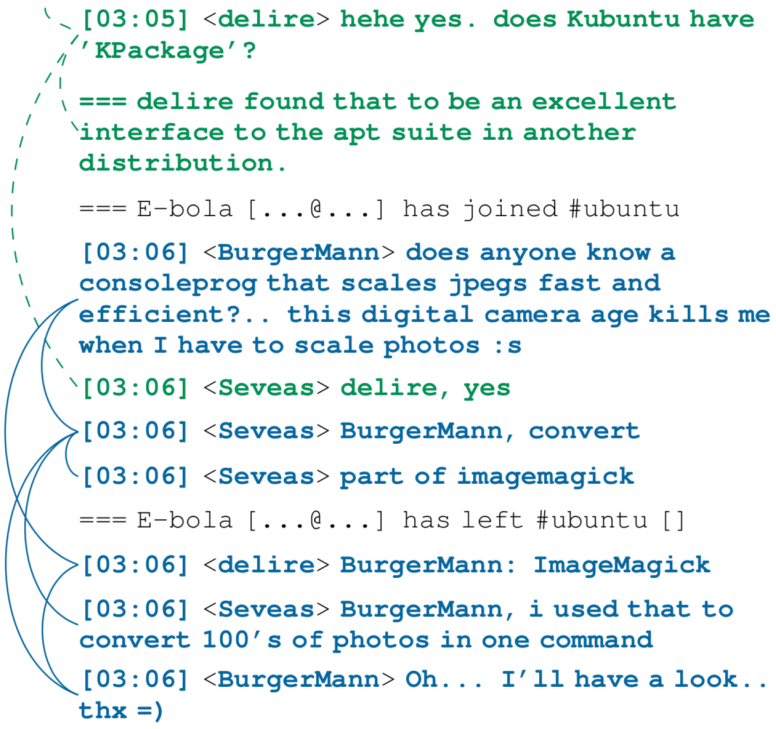


Figure 5.1: A sample log from the #Ubuntu IRC channel, earliest message first. The curved lines represent two different conversations happening at the same time. Notice that the username `delire` is speaking in both conversations to separate users.

The shortcomings of this expert-driven rules creation approach led to the creation of a larger team effort that looked at overcoming these issues with a more powerful AI approach. As part of this larger team effort, a hand-created dataset of more than 77k messages (a dataset 16x larger than previous similar datasets combined) was used to train ML models which then labeled a data set of the remaining 37 million messages [60].

What if the expert wishes to switch the domain or the channel? As effective as the results are, it is clearly inefficient and infeasible for experts to manually annotate and hand-label 77k messages for each new channel for which they wish to set up automated support. There is a huge need for more efficient methods to allow individual channel owners to make automation work on their own data. Can we create a system that can enable rapid annotation of large corpuses from new IRC channels with minimal effort from channel owners?

```
    [21:29] <MOUD> that reminds me... how can I use
    CTRL+C/V on terminal?
    [21:29] <MonkeyDust> MOUD ctrl ins pasts
    [21:29] <nacc> MOUD: it depends on your
    terminal application, in gnome-terminal ...
->  [21:30] <MOUD> -.-

    [17:35] <Moae> i have to remove LCDproc ...
    [17:38] <Madsy> Moae: sudo make uninstall &&
    make clean? :-)
    [17:39] <Madsy> Open the makefile and see what
    the targets are.
->  [17:40] <Madsy> Moae: Don't message people in
    private please. It's ...
    [17:42] <Moae> Madsy: sorry
    [17:42] <Moae> Madsy where i have to launch the
    command?
```

Figure 5.2: This conversation snippet shows annotation ambiguity that arises in the IRC messages. The message from MOUD could be a response to either `MonkeyDust` or `nacc`. In the same vein, the message from `Madsy` could be a part of this conversation or to another one entirely.

In this project, we develop a hybrid intelligence system whose ultimate goal is to enable an AI system that has been trained to disentangle conversations in domain A (or channel A) to be able to disentangle conversations in domain B (or channel B). The labeled data that the AI system is trained on exists for domain A. To get to this step, however, we first explore what it means to create a novel interface for multi-domain conversation disentanglement, specifically with interactional slingshots (referred to with the acronym "IS" in the rest of this chapter) that *guide* user annotations.

5.2 Related Work

As seen in Section 2.1.3, related work in this area of research has historically focused on the disentanglement task itself, and less on the interface development. Our main focus for this project is to create an effective interface that, via interactional slingshots, can help annotators disentangle text from multiple domains.

However, in general, there exist several tools that automatically preannotate text to help annotators with a range of NLP tasks. These interfaces include GATE Teamware [13], WebAnno [116], and AlvisAE [87]. Perhaps the closest interface to ours is from Klie et al. [54], who introduce a domain-agnostic human-in-the-loop approach for Entity Linking

tasks based on their previous INCEpTION [53] interface. This interface enables annotators to suggest new annotations, rejections, etc., at any time in their annotation process. Klie et al.'s work is similar to ours in that they also study a variant of the "No IS" and "IS" study that we describe later in this chapter.

SLATE [59] is the most direct inspiration for our work. SLATE was built as a flexible annotation tool that supports three annotation types: applying categorical labels, writing free text, and linking portions of text. The latter annotation type is the inspiration for the "Link" mode, which is described in Section 5.5. Our annotation interface borrows concepts from SLATE, but unlike SLATE, is fully-online, and uses mouse input for selections.

5.3 Task Definition

We consider the same overall conversation disentangling task as seen in [60], which we describe again here. An Internet Relay Chat (IRC) shared channel contains a group of users that communicate with each other about a topic, often technical in nature. Each message in this chat is timestamped. Sometimes, these users use *directed* messages, in which they mention the users to whom they are replying; otherwise, they just reply, which we then consider an *undirected* message. For the annotation task, there are two major flavors: **Link**, where annotators label the previous message to which this message is a responding or referring to. This provides a *graph* structure in which messages are nodes and edges indicate that messages are responses to each other; and, **Conversation**, where annotators create ongoing conversational *snippets* to which they then assign each message. A conversation can be inferred from the graph structure, but not vice-versa.

5.4 Data

Our data consists of 16 files with varying levels of disentangling difficulty that were sampled from four different channels: `Stripe`, `Rust`, `Ubuntu-Meeting`, and `MediaWiki`. Stripe and Rust contain jargon, MediaWiki contains mostly bug reports, and Ubuntu-Meeting contains mostly meeting jargon. Each file contains 200 lines that were manually annotated by us. However, we subsample for 34 lines from these files for the annotation task in order to make the task time manageable for our studies.

Adjudication for Gold Standard

For the gold standard dataset creation, we had four total annotators: Each of the 16 files were annotated in the Link task by two unique authors using the SLATE interface [59], and a third author then *adjudicated* between these annotations to improve quality. During this adjudication step, there was no indication of which author had been given which annotation, and there was an option to choose another annotation entirely. These adjudicated annotations serve as our gold standard.

Summary Stats

Table 5.1 shows relevant summary stats from the set of annotation files, grouped by domain. Even though each file comprises 34 lines, there is a large variability in characteristics. For instance, the shortest conversation time span (time elapsed from Line 1 to Line 34) was 4.00 minutes for the `ubuntu-meeting.0.100` file, whereas the longest time span contained in the files was 1204.05 minutes for the `mediawiki.1.100` file. Similarly, we notice that the Stripe and Ubuntu-Meeting channels contain a mix of directed and undirected messages, but Rust and MediaWiki are heavily biased towards undirected messages. Another feature of variability for these files exists in the "Number of Users Directly Addressed" column. For those users directed their messages at others, this column shows the number of unique users that were addressed by such a user. For instance, for the `stripe.0.100` file, one user addressed eight unique users, whereas for the `mediawiki.0.200` file, there was only one user for one directed message.

File	Agreement	Duration (min)	Users	Users per Hour	Number of Users Directly Addressed	Avg. Words per Message	Directed Messages	Undirected Messages	Messages per Hour
stripe.0.100	0.771	21.20	9	26	[1, 1, 1, 1, 8]	22	19	15	97
stripe.1.100	0.813	29.50	9	19	[1, 1, 1, 1, 6]	24	13	21	70
stripe.2.100	0.893	90.98	7	5	[1, 1, 4]	19	17	17	23
stripe.0.200	0.771	18.05	5	17	[1, 1, 1, 2, 4]	22	23	11	114
rust.0.100	0.808	30.35	9	18	[1, 1, 1, 1, 2]	13	11	23	68
rust.1.100	0.814	152.37	11	5	[1, 1, 1, 1, 1, 2]	11	8	26	14
rust.2.100	0.758	84.47	8	6	[1, 1, 1, 1]	14	5	29	25
rust.0.200	0.808	78.62	8	7	[1, 1, 2]	11	11	23	26
ubuntu-meeting.0.100	0.731	4.00	7	105	[1, 1, 1, 2, 2, 2]	9	14	20	510
ubuntu-meeting.1.100	0.656	7.00	15	129	[1, 1, 2, 3, 3]	5	11	23	292
ubuntu-meeting.2.100	0.735	6.00	4	40	[1, 1]	10	15	19	340
ubuntu-meeting.0.200	0.731	6.00	9	90	[2, 2]	12	6	28	340
mediawiki.0.100	0.858	58.52	10	11	[1, 1, 1]	12	3	31	35
mediawiki.1.100	0.685	1204.05	10	1	[1, 1]	16	2	32	2
mediawiki.2.100	0.710	15.15	7	21	[1, 1, 2]	8	4	30	135
mediawiki.0.200	0.858	18.40	5	17	[1]	8	1	33	111

Table 5.1: Summary stats broken down by individual files.

5.5 Interface Conditions

All the client and server-side software is written in JavaScript and uses off-the-shelf libraries. As mentioned in Section 5.3, there are two major ways of doing the conversation disentanglement task: "Link" mode, where annotators find the previous message being replied-to (resulting in graphs with nodes and edges), and "Convo" mode, where annotators separate the entangled chat into constituent conversations. Since we are interested in adding guidance support to the annotators' interactions with the interface, we have two further variations: No IS mode, where the annotators do not receive any interactional slingshot support (so these are the baseline conditions), and IS mode, where annotators receive interactional support.

Unlike the SLATE interface, which is a custom-built tool that utilizes the command line [59], our interface is fully web-based. Figures 5.3, 5.4, 5.5, and 5.6 show our interactive annotation interface, built using the Meteor framework. Depending on the task condition ("Link" or "Convo") and whether or not interactional slingshots are present ("No IS" vs. "IS"), annotators can interact with different features.

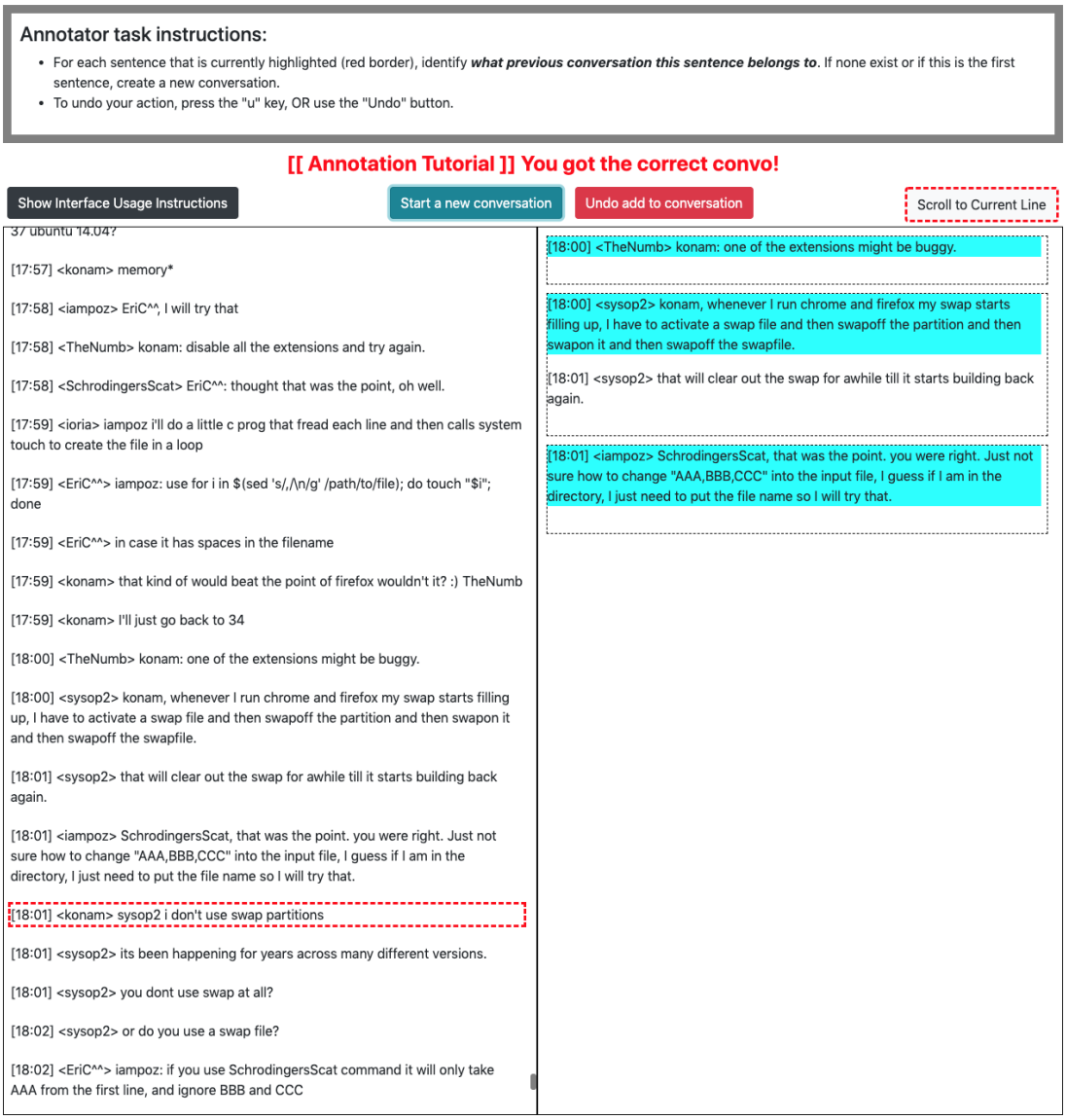


Figure 5.3: Web-based interactive annotation interface for the Convo-No-IS condition. There are no slingshots afforded for this interface. (Note: these screenshots show each interface at the same point in the Tutorial file.)

In the **Convo-No-IS** condition, seen in Figure 5.3, annotators can scroll through the entangled conversations on the left hand side, and create new conversation "snippets" that show up on the right hand side. Annotators can add sentences by clicking inside each conversation container. The blue-highlighted sentence is the start of the snippet and occupies a "sticky" position. No other interactional slingshots are afforded for this interface.

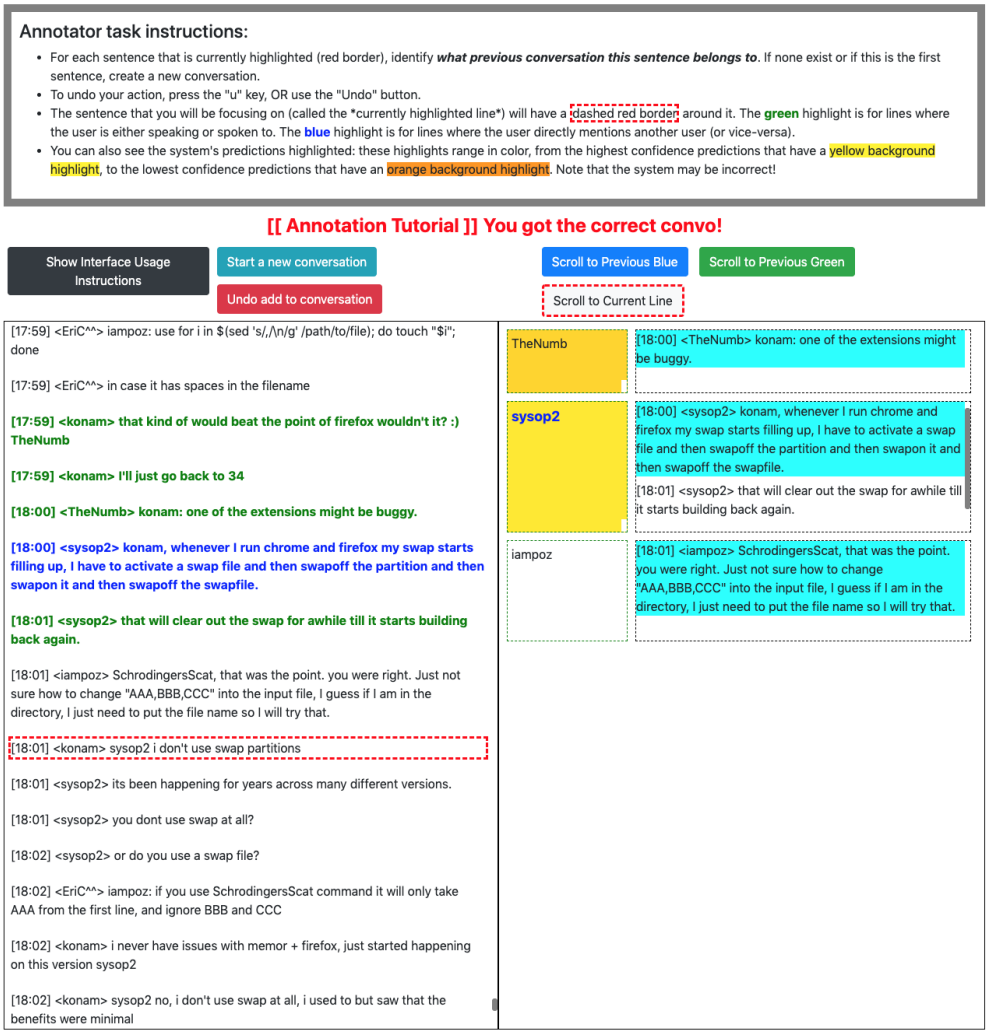


Figure 5.4: Web-based interactive annotation interface for the CONVO-IS condition. The interactional slingshots include username highlighting and system predictions for conversation snippets. (Note: these screenshots show each interface at the same point in the Tutorial file.)

In the **CONVO-IS** condition, seen in Figure 5.4, annotators have access to interactional slingshots: *username displays box*, where the conversation snippets display the users that are currently in that snippet; *system predictions*, which highlight the username displays box based on the system's confidence in the correct conversation prediction, where bright yellow is the highest confidence prediction and dark orange is the lowest confidence prediction; and, *username highlights*, where the current username and any targeted usernames are automatically highlighted in the entangled chats, as well as in the username displays. The highlights are blue if the user is actively speaking at another user, and green if any conversation involves this user.

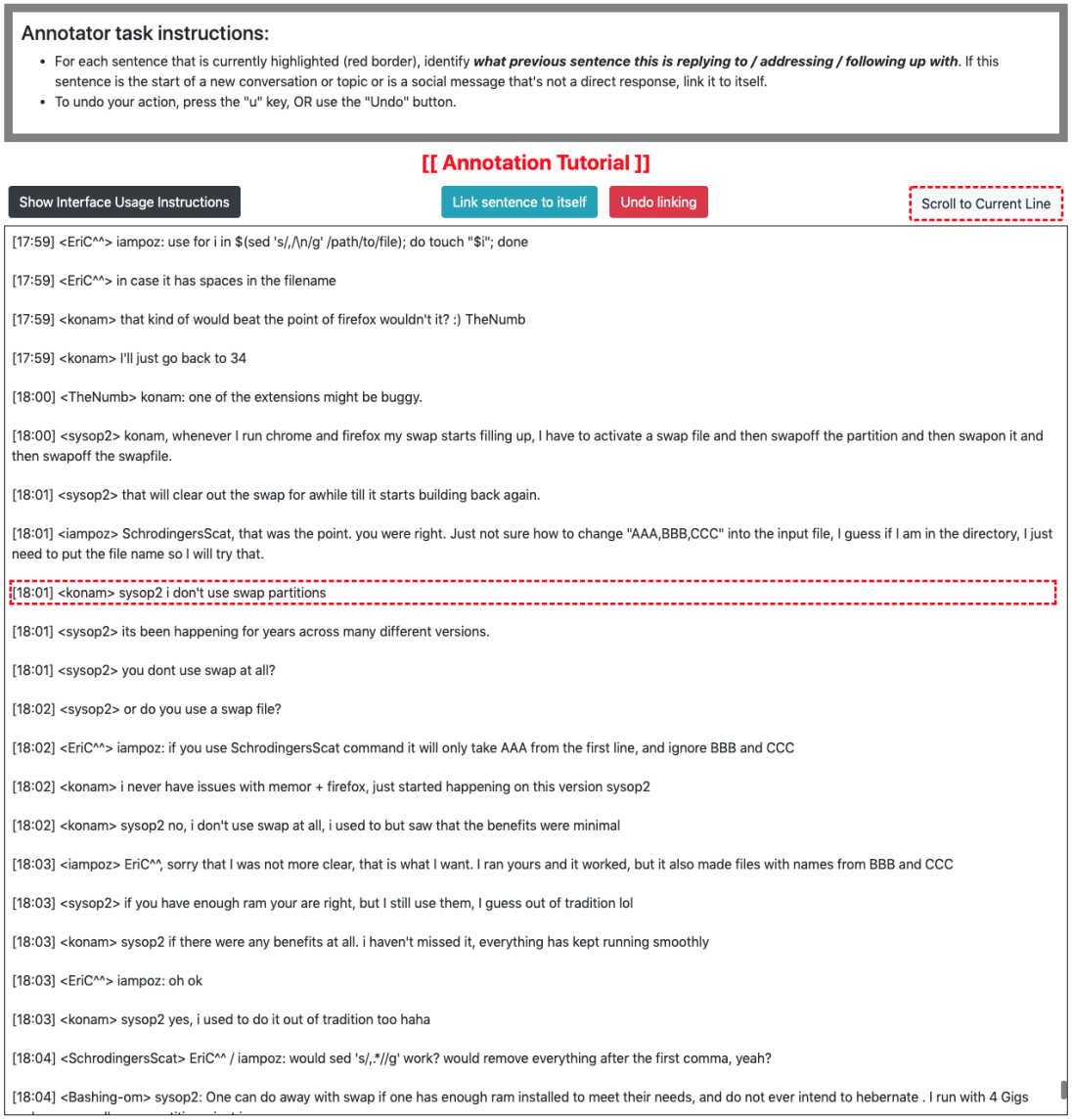


Figure 5.5: Web-based interactive annotation interface for the LINK-NO-IS condition. There are no slingshots afforded for this interface. (Note: these screenshots show each interface at the same point in the Tutorial file.)

In the **LINK-NO-IS** condition, seen in Figure 5.5, annotators can scroll and click on the sentence they think the currently highlighted line (shown in the red border) is responding to. Since this is a baseline condition, annotators do not receive any interactional help from the system. As a result, annotators cannot visually tell to what previous sentence they linked the currently-highlighted sentence.

Figure 5.6: Web-based interactive annotation interface for the LINK-IS condition. The interactional slingshots include username highlighting and system predictions for links. (Note: these screenshots show each interface at the same point in the Tutorial file.)

Finally, in the **LINK-IS** condition, seen in Figure 5.6, annotators once again have access to the three interactional slingshots as in the CONVO-IS condition. The major difference is that, unlike in the CONVO-IS condition, the entangled sentences on the left hand side are highlighted. The system highlights similarly range from yellow (for the highest confidence prediction) to orange (for the lowest confidence prediction).

5.6 Experimental Setup

5.6.1 Hypotheses

There are three hypotheses that we explore in our experimental study, when comparing to the No IS mode:

1. H1: Using IS tools will improve the accuracy for both the Convo and Link modes.
2. H2: Using IS tools will lead to time savings during the annotation.
3. H3: Using IS tools will lead to a speed-up in annotation the further along a file the annotator is.

5.6.2 Annotation Guidelines

We developed annotation guidelines based on our aforementioned previous work:

- In Convo mode, for each currently highlighted sentence, annotators are instructed to identify to what previous conversation this sentence belongs. If none exists or if this is the start of the task, annotators are instructed to create a new conversation snippet.
- In Link mode, for each currently highlighted sentence, annotators are instructed to identify what previous sentence this current sentence is replying to, addressing, or following up with. If no such sentence exists or if the current sentence is the start of a new topic, conversation, or a social message, then annotators are instructed to link this sentence with itself.
- For both modes, we provided examples of annotations for edge cases. Example edge-cases include what to do when a question is repeated, when the user is engaged in multiple conversations, when a user asks new questions that are different from the existing conversation thread, what to do with a series of lines of output, and how to deal with "ubottu," a bot.

5.6.3 Annotators

Human Annotators

The annotators were all fluent English speakers with a background in computer science, a necessary component to understand the technical content present in these domains. Of the 21 total annotators: 17 were doctoral students, 1 was an undergraduate student, 1 received their undergraduate degree, 1 received a Master's degree, and 1 received a doctoral degree. Of the four author annotators: 1 was pursuing a doctoral degree, 1 was pursuing a Master's degree, 1 graduated undergraduate, and 1 was a postdoc. All adjudication was performed by a postdoc, who is a native English speaker.

Machine Models

From our previous work [60], we had trained models from the Ubuntu channel domain. Even though these four channels are out-of-domain for those models, we include the machine's performance in the results, as it is an important baseline and also provides the system suggestions for the IS interfaces.

5.6.4 Pilot Study

We recruited four annotators (all doctoral students doing research in NLP) for a pilot study. Each annotator was randomly assigned one of the four interface conditions and had to annotate both files, for a total of 200 lines of annotation. The pilot study was informative for the construction of our actual study:

- *Task length.* We aimed for a task duration time of about 30 minutes based on previous experience, but noticed that the annotators took a lot longer than 30 minutes (one person took almost 70 minutes on one file).
- *Practice with interface.* We noticed that one of the annotators "played around" with the interface before starting the task in earnest. In the post-task survey, one of the participants suggested that we create a tutorial to make the interface explanations more clear.
- *Task confusion.* Our annotators expressed confusion in how to get started with the task, as the instructions were unclear. All of them said that the ubuntu-meeting file was the hardest one to annotate.

Based on the results and comments from this pilot study, we implemented an interactive tutorial (screenshots from each tutorial is seen aforementioned series of interface figures), added a task instructions section to the interface, added confidence score questions to the post-task survey, and generally improved instructions.

5.6.5 Study Setup

Each of the 16 participants were randomly split into one of the four task conditions, and received four files from the three channels that they had to annotate. Each file had 34 lines that needed to be disentangled. To mitigate any learning effects (which could make performance on later files appear better) and to reduce task fatigue (which could make performance on later files appear worse), the order of the files shown was randomly assigned to be one of four possibilities. This means that each annotation file was 1st, 2nd, 3rd, and 4th only once for each of the four interface conditions: Convo-No-IS, Convo-IS, Link-No-IS, and Link-IS.

Before accessing the task, however, each participant had to first pass a tutorial. The tutorial's aim was to help the annotators better understand both the task, as well as the interface. If the participant got an annotation wrong, an alert would show up and show the correct answer, which the participant had to click before continuing. For the participants in the Convo-IS and Link-IS conditions, the tutorial also had system suggestions highlighted. Since some of these suggestions were incorrect, we hoped that the participants could learn that not all system suggestions, even if confident, were automatically right.

After six lines of practice, participants could move to the main task. Once all four files had been annotated, participants filled out a post-task survey that asked participants to self-rate their confidence level of how well they thought they did the task, from a scale of 1 (least confident) to 5 (most confident). The survey also had open-ended questions, which we address in the Discussion section.

5.6.6 Task Measures

For both tasks, we measure timing and accuracy.

For the Convo task, we consider two broad measures: **(1) Exact Match** (Precision, Recall, and F1): these measures are calculated from the number of perfectly matching conversations, excluding conversations with only one message. As described in [60], this is a challenging metric, but as with there, we include it here because it directly measures the

partcipants' ability to perfectly extract conversations.; and (2) **One-to-One Overlap (1-1)**, which measures the percentage overlap when conversations from two annotations are optimally paired up using the max-flow algorithm (the same measure was used in [60]). Higher values indicate greater overlap between participant annotations and gold standard annotations.

For the Link task, we consider **precision, recall, and F1 score** when comparing with the gold standard. Since links provide a graph structure, we infer sets of messages (conversations) and run the aforementioned Convo task metrics for this as well. (In the results tables, conversation measures that were inferred from the Link graph structures will say "link" in the "Task" column heading.)

5.7 Results: Quantitative Analysis

We break the results into two broad categories: **Quantitative**, where we observe the impact the different task and interface conditions had on the output measures, and **Qualitative**, where we examine the annotations themselves to better understand the task and characterize annotator output.

5.7.1 Outlier Participants

There were two outlier participant data points—both in the IS conditions—so we show analysis with and without adjusting for these outliers.

The first outlier, Participant 5, not only had the lowest scores and abnormally fast time to completion for all files, but also they contacted the authors after completing the tutorial saying that they "didn't quite understand the tutorial" and that they weren't sure how to progress in the task. The adjustment for this outlier was to recruit another participant, Participant 17, who was given the same task.

The second outlier, Participant 16, performed the worst on their first file and contacted the authors saying that they were frustrated with the UI on the first file and had "little to no idea what's going on" and that the first file "was hard"; however, their performance on their next three files was seemingly unaffected by their frustration with the UI. The adjustment for this outlier performance was to give this participant a redo with another file from the same domain as their first file.

Participant	Task	Condition	Matched P	Matched R	Matched F1	1-1	Time on task (seconds)
Machine	-	-	53	55	53	78	-
1	convo	no IS	54	60	55	78	364
2	convo	no IS	70	67	68	93	338
3	convo	no IS	38	46	41	73	472
4	convo	no IS	38	42	39	74	287
(Original)			**50**	**53**	**51**	**80**	**365**
5*	convo	IS	15	10	12	60	157
17	*convo*	*IS*	*81*	*78*	*80*	*95*	*827*
6	convo	IS	35	34	35	77	262
7	convo	IS	65	65	65	87	267
8	convo	IS	56	52	54	91	280
(Original)			**43**	**40**	**41**	**79**	**242**
(Modified)			**59**	**57**	**58**	**87**	**409**
9	link	no IS	37	35	36	86	482
10	link	no IS	54	54	54	85	340
11	link	no IS	44	47	45	73	272
12	link	no IS	38	42	39	77	324
(Original)			**43**	**44**	**43**	**80**	**354**
13	link	IS	94	94	94	99	514
14	link	IS	58	59	58	89	354
15	link	IS	50	48	49	84	468
16*	link	IS	63	58	60	79	1171
16	*link*	*IS*	*81*	*73*	*77*	*96*	*1159*
(Original)			**66**	**65**	**65**	**88**	**627**
(Modified)			**71**	**69**	**69**	**92**	**624**

Table 5.2: Summary **Convo metric stats** from the study. (The values from the "Link" task condition are inferred from their graph structure.) Each participant value represents their combined average for their four files. There are two outlier participant data points—both for the IS condition—shown here with an asterisk (*) on the participant ID. We show averages with and without these outliers: Values inside the gray cells are averages for that group of participants. Values inside the yellow cells are averages where we replace the asterisk value with the values in the italics. We explore potential reasons for this in the Discussion section.

For large-scale annotation efforts, we expect these issues to be resolved in annotator training. If domain experts are annotating their own data files, we further expect such kind of confusion to be mitigated, since domain experts would know both why they are doing the disentanglement task, as well as information about their own channel: what the discussions are about and how those discussions flow.

Participant	Task	Condition	Precision	Recall	F1	Time on task (seconds)
Machine	-	-	68	69	68	-
9	link	no IS	75	75	75	482
10	link	no IS	67	67	67	340
11	link	no IS	56	56	56	272
12	link	no IS	76	75	76	324
(Original)			**69**	**68**	**68**	**354**
13	link	IS	85	84	85	514
14	link	IS	78	78	78	354
15	link	IS	74	74	74	468
16*	link	IS	63	61	62	1171
16	*link*	*IS*	80	*78*	*79*	*1159*
(Original)			**75**	**74**	**75**	**627**
(Modified)			**79**	**79**	**79**	**624**

Table 5.3: Summary **Link metric stats** from the study. (The Precision, Recall, and F1 scores here are the Link measures.) Each participant value represents their combined average for their four files. There are two outlier participant data points—both for the IS condition—shown here with an asterisk (*) on the participant ID. We show averages with and without these outliers: Values inside the gray cells are averages for that group of participants. Values inside the yellow cells are averages where we replace the asterisk value with the values in the italics. We explore potential reasons for this in the Discussion section.

5.7.2 Overall Performance

Table 5.2 shows summary results for the Convo measures when comparing across the Convo task mode, and Table 5.3 shows summary results for the Link measures when comparing with the Link task mode. Overall, for both the Convo and Link tasks, we support our hypothesis that the interface with interactional slingshots will lead to more accurate performance, as we see precision, recall, F1, and One-to-One metrics improve, especially for the "harder" channels. However, the time spent on task also increased, so there are no time savings afforded with the IS tools. So, we support our H1, but do not find enough evidence to support H2 (We report significance testing in Section 5.7.3).

Convo Task

When compared with the machine's baseline, for the Convo task, the percent increase in the 1-1 metric for the Convo-No-IS is 2.56%, and the percent increase for the original Convo-IS is 1.28%. After accounting for the outlier, the modified Convo-IS has a 11.54% increase for the 1-1.

Recall that participants who did the Link task have annotations from which conversations can be inferred (by traversing the graph). For these, the increase in 1-1 is 2.56% when using the Link-No-IS, 12.82% when using Link-IS unadjusted for the outlier file, and 17.95% using Link-IS when adjusted for the outlier.

Link Task

For the Link task, when looking at the F1 score, there is no increase when using Link-No-IS when comparing with the machine's performance. However, when using the original Link-IS files, we see a 10.29% increase, and when using the modified Link-IS mode, we see a 16.18% increase. However, the time increase for the Link-IS (orig.) is 77.12% and for the Link-IS (mod.) is 76.27%.

Condition	Matched F1	1-1	Link F1	Task Time (seconds)
Machine	53	78	68	-
All No IS	47	80	68	**360**
All IS (mod.)	**64**	**90**	**79**	517
All Convo (mod.)	55	83	-	387
All Link (mod.)	56	86	74	489

Table 5.4: Averages of the accuracy and time stats for both the No IS (combining Convo-No-IS and Link-No-IS) and IS (combining Convo-IS and Link-IS) conditions.

Combined Tasks

Table 5.4 shows differences between the two No-IS and IS groups, which we get by combining across the two task modes. When compared with the machine's performance, we see that all IS conditions, adjusting for the outliers, provides the greatest benefit in accuracy performance: Matched F1 is improved by 20.75%, 1-1 by 15.38%, and Link F1 score by 16.18%. When comparing across the two tasks for both conditions, we see that all the Link annotations provide slightly higher Matched F1 and 1-1 scores than thsoe seen in all the Convo annotations.

5.7.3 Significance

Table 5.5 shows p-values for different condition groups after performing one-tailed unpaired *t*-tests, as our null hypothesis is that there is no increase in the Match (Convo) F1, One-to-One, Link F1, and Task time measures given our IS interventions. Since we are performing multiple tests, we apply a correction using the Holm-Bonferroni Method [44] to control for the potentially-higher probability of introducing false positive errors. Because we are conducting multiple hypothesis testing of different metrics on the same two populations, this correction can help account for any family-wise errors. However, we note that this is probably an overly aggressive correction because we know that the three accuracy measures—Match F1, 1-1, and Link F1—are strongly correlated.

Even with the correction applied, as we can see, there is a statistically-significant difference for the Link F1 score when using the Link-IS mode versus using the Link-No-IS mode, once outliers are adjusted for. Similarly, there is a statistically-significant difference in Task Time when using Convo-No-IS and Convo-IS (original), along with Convo-IS (original) versus Link-IS (original).

Table 5.6 shows p-values when combining across the interface conditions. We see a statistically-significant difference for the Match F1, 1-1, and Task Time measures when comparing the All No IS group with the All IS (modified) group.

Convo No IS	Convo IS (original)	Convo IS (modified)	Link No IS	Link IS (original)	Link IS (modified)	Match F1	1-1	Link F1	Task Time
X			X			0.280	0.478	-	0.431
			X	X		0.060	0.132	0.141	0.017
			X		X	0.026	0.020	**0.007**	0.019
X	X					0.229	0.449	-	**0.012**
X		X				0.280	0.096	-	0.291
	X			X		0.045	0.096	-	**0.001**
		X			X	0.019	0.160	-	0.060

Table 5.5: Unpaired One-tailed *t*-test p-values, corrected with the Holm-Bonferroni method. Statistically significant values are bolded. "Original" refers to values that contained the two outliers, whereas "Modified" refers to values that replaced those outliers, as described in the Results section.

All No IS	All IS (orig.)	All IS (mod.)	Match F1	1-1	Task Time
X	X		0.263	0.239	0.160
X		X	**0.033**	**0.008**	**0.020**

Table 5.6: Unpaired One-tailed *t*-test p-values for combined conditions, with a correction applied using the Holm-Bonferroni Method. "Original" refers to values that contained the two outliers, whereas "Modified" refers to values that replaced those outliers, as described in the Results section.

5.7.4 Channel Breakdown for Accuracy

Now that we looked at how annotator performance was across all the domains for the different conditions, let's look at the different channels in more detail. Table 5.7 shows a detailed breakdown of annotator performance for each file when combining the two interface conditions (No IS and IS). Bolded values show the condition that performed the best for that particular measure.

In this table, we see that the machine's performance beats aggregated human performance on 4/16 (25%) files for the "Convo" task, and on 5/16 (31%) for the "Link" task. (Three of these files (19%) are common to both task conditions.) For the remaining files, human performance beats that of the machine's.

In Table 5.8, we can see that, when aggregating across files for each channel, human performance always beats machine performance for "Link" mode, and all-but-one case for "Convo" mode.

File	Task	Prec	Rec	F1	Matched P	Matched R	Matched F1	1-1
stripe.0.100	machine	**88**	**88**	**88**	**100**	**100**	**100**	**100**
	convo, orig.	-	-	-	50	50	50	71
	convo, mod.	-	-	-	93	88	90	93
	link	81	81	81	76	76	76	96
stripe.1.100	machine	79	79	79	71	71	71	91
	convo	-	-	-	92	86	89	97
	link	**87**	**87**	**87**	**100**	**100**	**100**	**100**
stripe.2.100	machine	**77**	**77**	**77**	71	83	77	91
	convo	-	-	-	**75**	**84**	**79**	**93**
	link	76	76	76	67	75	70	88
stripe.0.200	machine	74	74	74	67	67	67	**97**
	convo	-	-	-	59	50	54	90
	link	**82**	**82**	**82**	**75**	**84**	**79**	96
rust.0.100	machine	67	66	67	20	25	22	56
	convo, orig.	-	-	-	37	38	37	84
	convo, mod.	-	-	-	70	**75**	72	91
	link	**90**	**87**	**89**	**75**	**75**	**75**	**99**
rust.1.100	machine	**85**	**85**	**85**	60	50	55	79
	convo	-	-	-	40	34	37	75
	link	80	80	80	**82**	**75**	**78**	**94**
rust.2.100	machine	**59**	**59**	**59**	**100**	**100**	**100**	**100**
	convo	-	-	-	80	88	84	97
	link	58	58	58	70	75	72	90
rust.0.200	machine	71	71	71	50	50	50	94
	convo	-	-	-	75	75	75	94
	link	**80**	**80**	**80**	**100**	**100**	**100**	**100**
mediawiki.0.100	machine	50	60	50	33	25	29	62
	convo, orig.	-	-	-	38	26	30	56
	convo, mod.	-	-	-	**69**	**63**	**66**	**81**
	link	**72**	**72**	**72**	61	57	59	**81**
mediawiki.1.100	machine	59	59	59	20	33	25	71
	convo	-	-	-	**63**	**67**	**65**	84
	link	**67**	**67**	**67**	42	50	45	**85**
mediawiki.2.100	machine	67	67	67	25	20	22	56
	convo	-	-	-	**50**	**50**	**50**	72
	link	**68**	**68**	**68**	**50**	40	45	**85**
mediawiki.0.200	machine	**79**	**79**	**79**	35	50	33	50
	convo	-	-	-	**54**	**63**	**58**	**78**
	link, orig.	43	43	43	0	0	0	41
	link, mod.	77	77	77	38	30	34	75
ubuntu-meeting.0.100	machine	67	67	67	**100**	**100**	**100**	**100**
	convo, orig.	-	-	-	13	25	17	65
	convo, mod.	-	-	-	38	50	42	81
	link	**78**	**78**	**78**	50	50	50	94
ubuntu-meeting.1.100	machine	35	35	35	0	0	0	44
	convo	-	-	-	**17**	**17**	**17**	**85**
	link	**58**	**58**	**58**	0	0	0	68
ubuntu-meeting.2.100	machine	**82**	**82**	**82**	**100**	**100**	**100**	**100**
	convo	-	-	-	0	0	0	57
	link	59	59	59	0	0	0	50
ubuntu-meeting.0.200	machine	53	50	51	0	0	0	50
	convo	-	-	-	0	0	0	69
	link	**74**	**68**	**71**	**25**	**17**	**20**	**75**

Table 5.7: Summary metric stats broken down by individual files. The machine's performance beats aggregated human performance on 5/16 files (31%) for the "Link" task, and on 4/16 (25%) files in the "Convo" task.

Channel	Task	Condition	Prec	Rec	F1	Matched P	Matched R	Matched F1	1-1
Stripe	link	no IS	79	79	79	66	74	69	91
		IS	**83**	**83**	**83**	**93**	**93**	**93**	**99**
	convo	no IS	-	-	-	79	84	81	96
		IS, orig.	-	-	-	58	51	54	79
		IS, mod.	-	-	-	80	70	74	90
	machine	-	80	80	80	77	80	79	95
Rust	link	no IS	72	72	72	**83**	**83**	**83**	**99**
		IS	**81**	**80**	**81**	80	79	79	93
	convo	no IS	-	-	-	58	61	59	90
		IS, orig.	-	-	-	58	56	57	85
		IS, mod.	-	-	-	75	75	75	89
	machine	-	71	70	71	58	56	57	82
MediaWiki	link	no IS	60	60	60	23	20	21	68
		IS, orig.	65	65	65	53	54	53	79
		IS, mod.	**82**	**82**	**82**	**72**	**69**	**70**	**96**
	convo	no IS	-	-	-	56	57	56	75
		IS, orig.	-	-	-	46	45	45	70
		IS, mod.	-	-	-	62	64	63	83
	machine	-	64	66	64	28	32	27	60
Ubuntu-Meeting	link	no IS	63	62	62	0	0	0	63
		IS	**71**	**70**	**70**	38	33	35	81
	convo	no IS	-	-	-	6	13	8	58
		IS, orig.	-	-	-	8	8	8	80
		IS, mod.	-	-	-	21	21	21	**88**
	machine	-	59	59	59	**50**	**50**	**50**	74

Table 5.8: Summary metric stats broken down by channel. When aggregating across all files for a channel, human performance always beats that of the machine's for "Link" mode, and all but one case for "Convo" mode.

5.7.5 Channel Breakdown for Errors

Table 5.9 shows a breakdown of all correct and incorrect annotations for the No IS, IS, and Machine annotations. The table shows the eight combination; for instance, for Table 5.9(c), the value of 12 indicates that, across all four files, only the IS condition was correct that many times (No IS incorrect and Machine incorrect). Table 5.9(e) shows the cumulative stats: across all 16 files (total of 544 messages), the IS condition was the only one correct for 39 messages, or 7.17% of the time, and the only one incorrect for 10/544, or 1.84% of the time. Similarly, the machine prediction annotations were the only correct ones 16/544, or 2.94% of the time, and was the only incorrect one almost 21.32% of the time (116/544).

An important result that arises from this error breakdown per channel is that voting as a means to improve accuracy would not help. The errors are correlated enough (e.g., when a machine makes a mistake, it is likely that the human also did the same) that voting across the different methods would likely not yield large benefits, and would be similar to using the one best option.

5.7.6 Time Per Annotation

On top of total task time that we see in the prior tables, we also look at how annotators did with respect to time per annotation for both the No IS and IS conditions for each channel. Figure 5.7 shows results for the `stripe` channel; Figure 5.8 is for `rust`; Figure 5.9 is for `ubuntu-meeting`; and finally, Figure 5.10 is for the `mediawiki` channel. Looking at these figures, our hypothesis H3—where we talk about time savings the further along the annotator is in a file—can be rejected. It does not appear to be the case that the context that the annotators are building is helping them to annotate the document faster as they move along.

Machine Correct / Machine Incorrect	IS Correct	IS Incorrect
No IS Correct	66 / 31	3 / 8
No IS Incorrect	10 / 6	4 / 8

(a) `stripe`

Machine Correct / Machine Incorrect	IS Correct	IS Incorrect
No IS Correct	48 / 38	3 / 9
No IS Incorrect	15 / 9	6 / 9

(b) `rust`

Machine Correct / Machine Incorrect	IS Correct	IS Incorrect
No IS Correct	44 / 29	3 / 9
No IS Incorrect	11 / 12	5 / 25

(c) `ubuntu-meeting`

Machine Correct / Machine Incorrect	IS Correct	IS Incorrect
No IS Correct	38 / 18	1 / 24
No IS Incorrect	20 / 12	1 / 22

(d) `mediawiki`

Machine Correct / Machine Incorrect	IS Correct	IS Incorrect
No IS Correct	196 / 116	10 / 50
No IS Incorrect	56 / 39	16 / 64

(e) Totals

Table 5.9: Breaking down correct and incorrect annotations for the No IS, IS, and Machine annotations. The table shows eight combinations of where No IS, IS, and Machine could have been correct and incorrect. For instance, for (c), a value of 12 indicates that the No IS and Machine were both incorrect, but the IS condition was correct for 12 of the 34 annotations. (e) shows a total across all of the file annotations (544 lines). Instances where only the IS condition was correct amount to 39/544, or 7.17% of the total annotations, and 10/544 (1.84%) where only IS was incorrect.

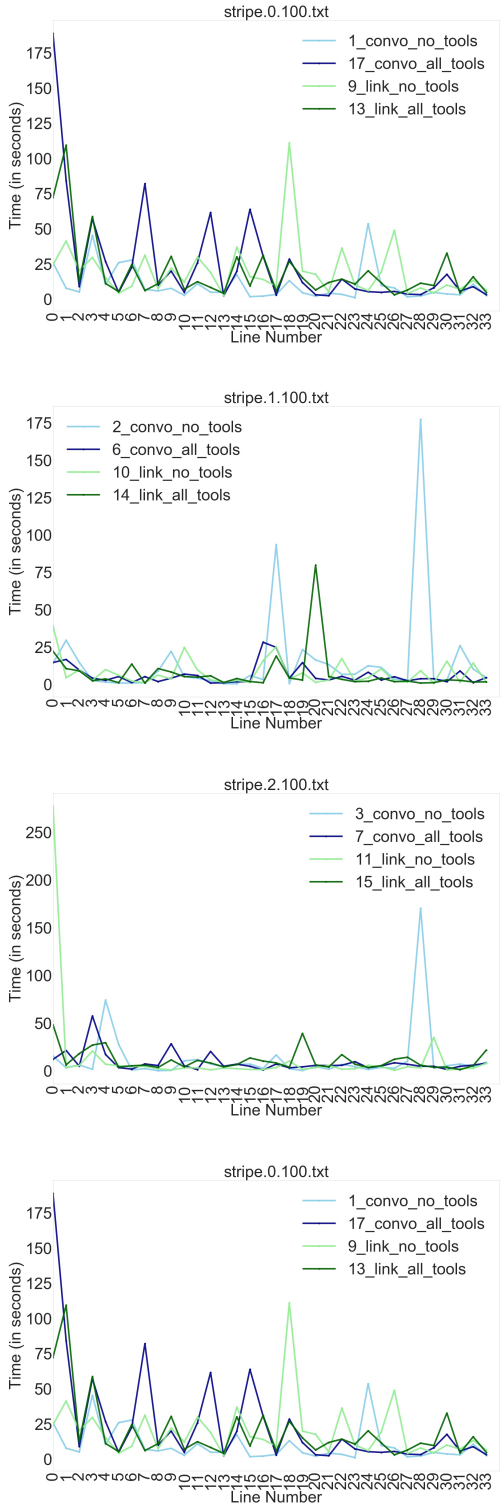


Figure 5.7: Time per annotation for all channels: `stripe`

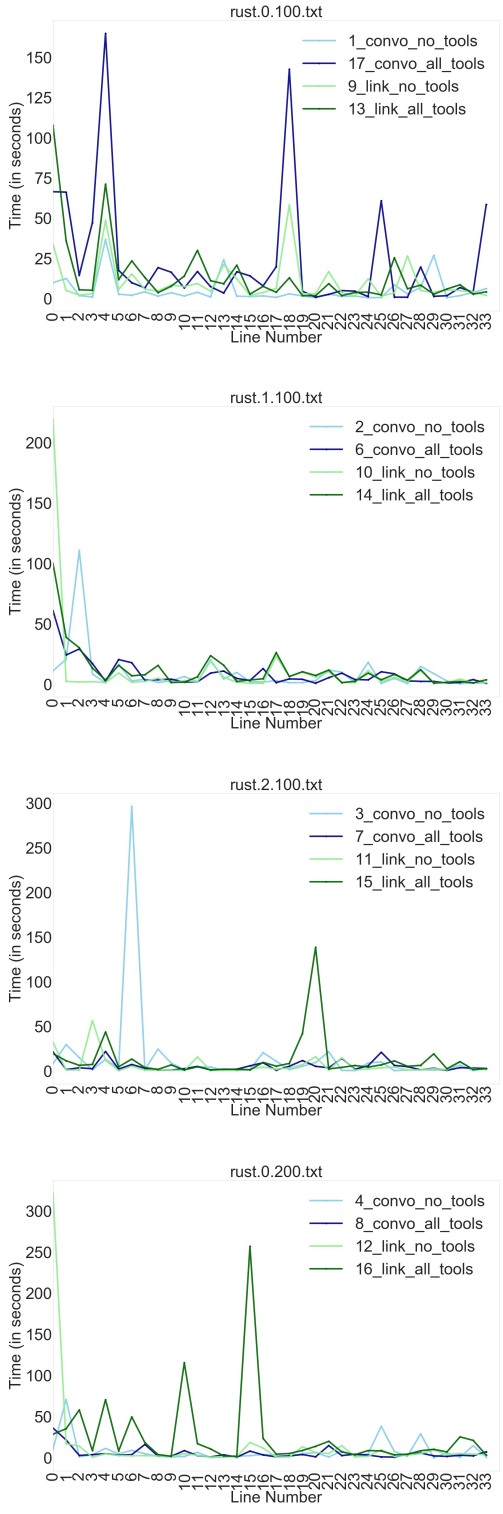


Figure 5.8: Time per annotation for all channels: `rust`

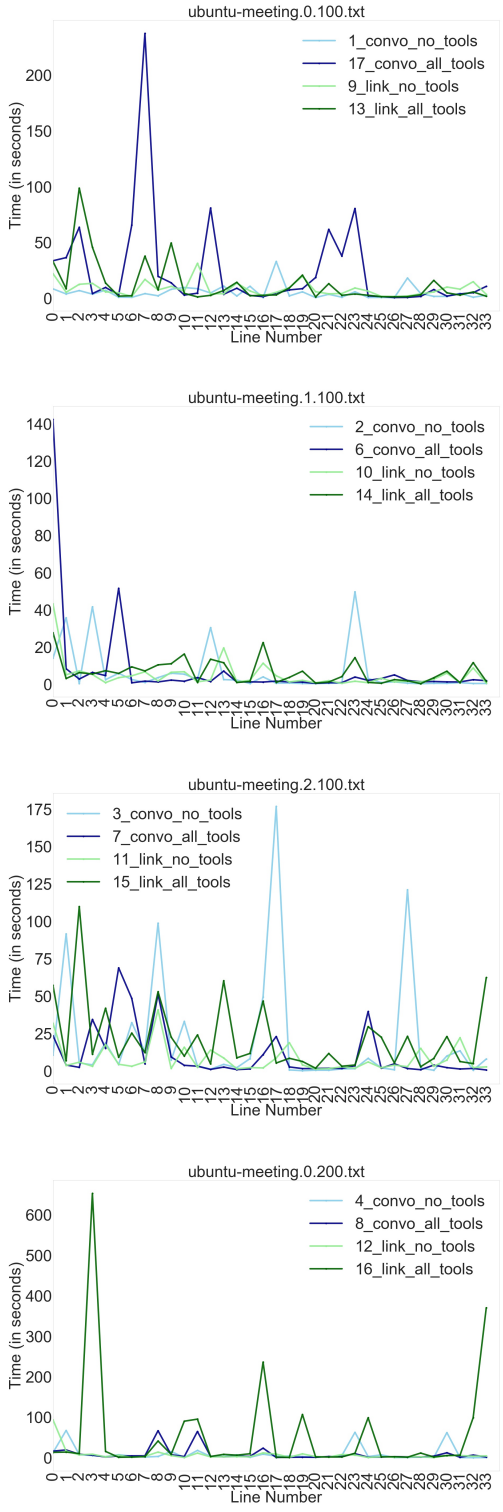


Figure 5.9: Time per annotation for all channels: `ubuntu-meeting`

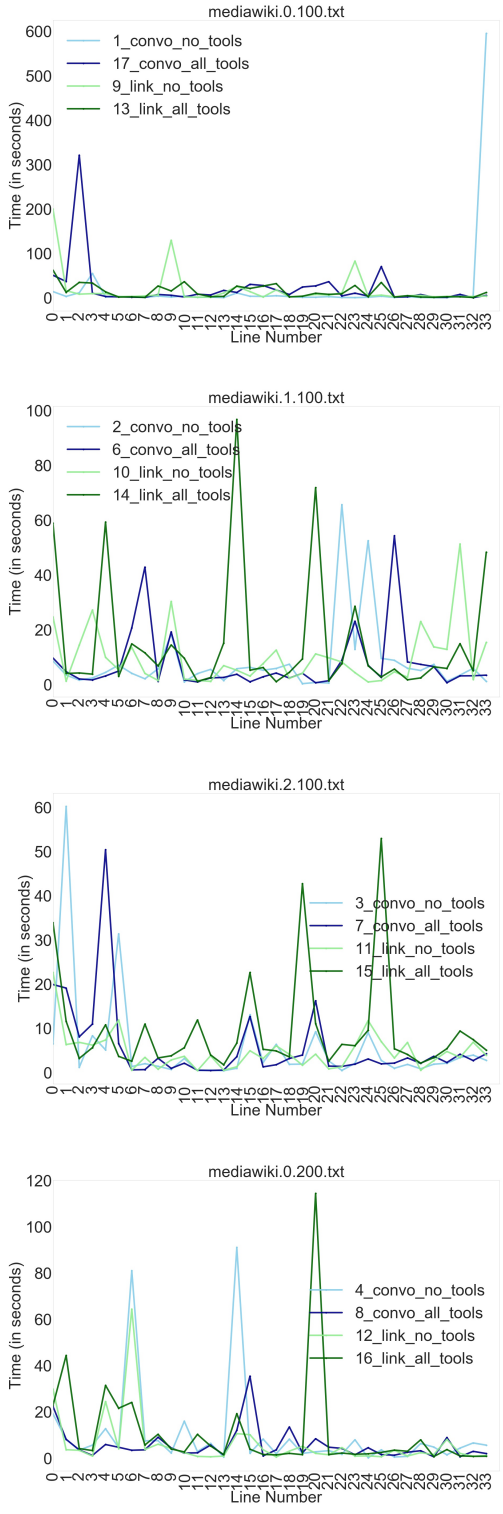


Figure 5.10: Time per annotation for all channels: `mediawiki`

5.8 Results: Qualitative Analysis

5.8.1 How much jargon is present in the files?

Part of what makes conversational disentanglement a difficult task is the amount of technical concepts being discussed by the users. We annotated each line in all 16 files to see whether jargon[1] was present in the file, and what impact, if any, this had on the output annotation accuracy measures. Table 5.10 shows the percent jargon that is present in each annotation file. We can see that the `Stripe` and `Rust` channels have the most jargon, whereas `Ubuntu-Meeting` and `MediaWiki` do not have as much. What is different about the latter two channels is that it contains more Bot and bug report activity than the first two files. For instance, although "ubuntu-meeting.1.100" does not contain technical jargon, it does contain a way that people vote during the meetings (using a slash with the letter 'o'). "ubuntu-meeting.0.200" is mostly filled with bug report information (URLs, links to the reports themselves).

When comparing the per-file accuracies that we see in Table 5.1, we notice that the machine's accuracy is higher for those channels that contain more jargon, whereas annotator performance is lower. This flips around for the two channels that contain less amount of jargon.

5.8.2 Annotator Feedback

Task Difficulty

Conversation disentanglement is a difficult task, as it requires domain knowledge and being able to juggle the context of multiple simultaneous conversations, especially if the users use undirected messages. We found that, in all cases of annotation tasks—for the gold standard annotation, pilot study, and the 16-person study—at least one annotator was completely confused with this task (one out of four, one out of four, and two out of 16, respectively, for the three scenarios). In P16's case, they were confused on the first file, but figured out the task on the second file, whereas for P5, they were confused for all four

[1]We define "jargon" as content that is endemic to a particular channel. This covers technical content, but also channel-specific items such as how people vote in the ubuntu-meeting channel.

File	% Jargon
stripe.0.100	65
stripe.1.100	35
stripe.2.100	59
stripe.0.200	67
	57
rust.0.100	59
rust.1.100	47
rust.2.100	67
rust.0.200	56
	57
ubuntu-meeting.0.100	65
ubuntu-meeting.1.100	15
ubuntu-meeting.2.100	47
ubuntu-meeting.0.200	53
	45
mediawiki.0.100	56
mediawiki.1.100	76
mediawiki.2.100	21
mediawiki.0.200	35
	47

Table 5.10: Percentage of lines in each annotation file that contains technical jargon. (I.e., # of jargon sentences / # of total sentences in file)

files. P5 messaged the author prior to starting the task saying "I was a little bit confused what the task is [...] I don't quite get the tutorial." P16 also messaged the author saying, "I think I need to do [the first file] again. I have little to no idea what's going on [...] Ok, I did the rust one, I think that was maybe better."

After reading through all of the annotators' feedback, we uncover the following themes that can shed light on why this task is particularly difficult:

- Understanding task rules: 3/16 (P8, P15, P16) annotators mentioned that understanding the task's rules was difficult, and that they only figured out how to proceed by continuing to do the annotations. P8 notes the edge-cases that can occur with this task, and commented, " It was initially hard to understand the task and what the rules of the task were. Questions like 'if this isn't the start of the conversation, but it's the first red-dotted snippet of the file, is that a new conversation?' caused some confusion."

- Parsing technical jargon: 6/16 (P1, P3, P4, P6, P7, P10) annotators mentioned that they were not familiar with the technical jargon that was being used. A few annotators resorted to using Google to figure out what some of the terms meant. P3 said that "I was unfamiliar with many terms, so did some [G]oogling to make sense of the conversations."
- Picking up the conversational context: 8/16 (P1, P4, P6, P7, P10, P13, P15, P17) annotators mentioned how difficult it was to get started on new conversations and to pick up the conversational context when new topics emerged.
- One-to-many responses: 3/16 (P2, P11, P14) annotators noted that the disentangling task became difficult when trying to figure out what messages a particular user is responding to, especially if that user was speaking with multiple others. For instance, P14 wrote, " Identifying multiple threads of conversation involving one person at the same time was difficult," a sentiment also echoed by P2, who wrote, "Also hard was when conversations overlapped because sometimes they are carrying on a conversation that happened messages ago and with the same username answering multiple threads, it could be difficult to keep track of the conversations they belong to."
- Bots created additional confusion: 4/16 (P9, P13, P14, P17) annotators said that they were unsure where Bot messages fit into the annotation scheme. P5 commented, "When there is (my guesses) a bot in the chat, it is difficult to judge which conversation it belongs to", and P17 wrote, "For example, the conversation that had the bots posting comments mind-flooded me for a while until I was able to figure out the role of the bots and thereby parse out what they were responding to."

Interface Difficulty

Although the IS interfaces help annotators disentangle conversations with higher accuracy measures, we are interested in seeing the overall usability of these interfaces. Adding more interactional elements to the interface—especially those as involved as the interactional slingshots—can make the interface more complicated to use and can increase the annotator's cognitive load. Making sure that the interface captures enough context and is usable, while not taxing the annotator's cognitive load, is an important design consideration for annotation interfaces, and hybrid intelligence systems in general.

While analyzing annotator feedback, we find these commonly-held views regarding the interface, which we split by the No IS and IS conditions:

No IS interfaces comments

- "[I like the] visual representation of separation between the conversations. I also liked that you could refer back to the instructions at any given time." [P1]
- "[C]licking on the conversation card on the right side to add a highlighted sentence to the conversation was awesome." [P2]
- "I like having the instructions ready on hand." [P9]
- "I wished I could 'go back' without undoing, like I wanted to remember which messages I had previously linked to which. [...] I found myself trying to keep the whole thread in my mind so sometimes. Also, sometimes after I got further in the conversation it started making more sense to me so I wanted to go back and check my previous answers in some way." [P10]

IS interfaces comments:

- "I liked the scroll to blue/green/current line options – made the interface more usable. I also liked the names getting added to the left blocks as the conversations were grouped by me. It helped me keep track of who was in which of my conversation groupings." [P6]
- "The yellow and orange suggestions [system predictions, where yellow is most confident and orange is least confident] were somewhat helpful but I didn't feel I could always trust them, but it helped me feel like I had some support." [P13]
- "Yellow highlight also was helpful in general, but when it was wrong, I had to think harder about what is the right answer. Also, among cases that I thought right, but when color is almost-orange, it also required a bit of more thinking, inspecting whether it is right or wrong." [P14]
- "I also didn't really use some of the buttons like for selecting "green" "blue" and the one with the red box. Not sure why they were there." [P15]
- "Being able to find previous messages from a particular user was helpful because people often participated in one conversation at a time. I could just look back and see how they responded to previous messages to get an idea if the current message is a part of that conversation or a different one." [P17]

Emerging Themes

What was interesting about the No IS interface comments was that a couple of annotators mentioned features that they wish the interface had, which were actually part of the IS conditions. For instance, P10's comment refers to the visual context, which is what the IS conversation snippets provide. P12 echoes this sentiment, writing, "A visual mapping of already marked threads would've been nice to keep track of things."

Another interesting theme that emerged from only the No IS interface comments is that a few annotators mentioned the ability to *undo* more than just one line at a time would have been helpful. P9 expresses this by saying that they "forgot about undo-ing because it's a different interface. It's like in a word doc, you know how an undo works and you can do it multiple times. This one, I kind of forgot or lost that intuition." Similarly, P2 states, "Sometimes, I changed my mind about whether an earlier sentence should go in a different conversation, and so had to backtrack and undo work to fix that spot – it didn't interfere much with this task, but I could see that being annoying on larger scale projects." Is is interesting to note that the IS interface comments did not mention undo-ing their annotations, but instead referred to the thought process they had to go through before doing an annotation.

An emerging theme from the IS interfaces is that the system highlights provide support, but annotators had to think more before trusting them. P13 and P14 both express similar sentiments, as seen above. Two annotators also mentioned that the color scheme used in the system highlights made it hard to distinguish between different shades that were close together in color.

5.9 Repeating the Task With Non-Expert Annotators

Thus far, we have explored what it means to do this task with annotators with domain knowledge, with some annotators even having expertise. But, what would change if we relaxed this constraint and conducted the experiment with non-expert annotators? Would the interactional slingshot guidance help non-experts overcome their knowledge gaps?

We repeat the previous study, but this time with non-expert annotators recruited from the Amazon Mechanical Turk platform [5]. To our knowledge, this task has not been performed with non-expert annotators.

5.9.1 MTurk Study Setup

For the study, we doubled the length of the interactive tutorial to be 12 lines of annotation. For each file task and interface condition, we recruited 2 unique workers from AMT who had at least a 98% HIT approval rating, for a total of 128 unique workers (16 files x 4 conditions x 2 workers per condition = 128 workers). We randomly split the workers into four groups, again mirroring the four different interface conditions, but this time, each worker only does the interactive tutorial and annotates their assigned task file. After they complete the annotation, workers fill out a post-task survey after which they can submit the HIT. For the post-task survey, we asked workers to evaluate how well they think they did on the task, as well as asked for general comments on the task and interface. We paid an effective rate of $15 per hour for this task (each HIT was worth $3.75).

5.9.2 MTurk Study Results

Figure 5.11 shows the results from this study. When averaging across all files in each domain, we see that annotators get higher conversation precision, recall, F1, and one-to-one scores for the Convo task using the LINK-IS condition for the `rust` domain. For the `mediawiki` domain, annotators perform better than the machine for 1-1 with the LINK-IS condition. However, the machine's performance bests that of the non-experts for all the Link mode tasks.

We note that there exists a lot of variability in this data given that there were only two crowd workers per file, but still believe that these results show promise. Across all the conditions, there are MTurk workers who could hold either the domain knowledge or expertise required to do this task. When removing those workers who spend fewer than 100 seconds on the task (as we consider them outliers), we see further improvements, as non-expert performance matches or beats that of the machine in *three* domains (refer to Figure 5.11b). When breaking the results down into individual worker performance, we notice a few workers matching or performing better than experts.

What this indicates to us is that this task, although difficult, could still benefit from pairing machines, non-experts, and experts together. For instance, there are clearly workers that do not perform well or are outliers, but there are others who might do well if we filtered for them. This can involve giving workers a "pre-task" with a set of annotations to better gauge what their performance on the real task might be.

Predicting workers' performance is outside the scope of this thesis, but holds promise as a future work direction, though there has already been research work that explores this topic [42, 49, 100]. In fact, the approach discussed in [93] could show benefits in selecting workers that are skilled at conversation disentangling tasks.

Channel	Task	Cond	Link P	Link R	Link F1	Conv P	Conv R	Conv F1	1-1	Time
Stripe	link	no IS	48	48	48	23	24	23	65	534
		IS	63	63	63	53	53	53	79	545
	convo	no IS	-	-	-	47	43	45	81	394
		IS	-	-	-	55	49	52	74	242
	machine		**80**	**80**	**80**	**77**	**80**	**79**	**95**	-
Rust	link	no IS	61	61	61	20	23	21	68	554
		IS	51	51	51	42	38	39	62	549
	convo	no IS	-	-	-	32	34	33	79	260
		IS	-	-	-	**62**	**65**	**63**	**84**	284
	machine		**71**	**70**	**71**	58	56	57	82	-
Media-Wiki	link	no IS	40	40	40	22	24	23	61	327
		IS	58	58	58	21	21	21	**67**	567
	convo	no IS	-	-	-	22	26	23	63	360
		IS	-	-	-	14	13	12	57	363
	machine		**64**	**66**	**64**	**28**	**32**	**27**	60	-
Ubuntu-Meeting	link	no IS	57	56	57	6	10	8	61	594
		IS	47	46	47	35	35	35	67	502
	convo	no IS	-	-	-	28	31	29	72	212
		IS	-	-	-	25	29	27	64	587
	machine		**59**	**59**	**59**	**50**	**50**	**50**	**74**	-

(a) Averages including outliers

Channel	Task	Cond	Link P	Link R	Link F1	Conv P	Conv R	Conv F1	1-1	Time
Stripe	link	no IS	48	48	48	23	24	23	65	534
		IS	63	63	63	53	53	53	79	545
	convo	no IS	-	-	-	47	43	45	81	394
		IS	-	-	-	55	49	52	74	242
	machine		**80**	**80**	**80**	**77**	**80**	**79**	**95**	-
Rust	link	no IS	61	61	61	20	23	21	68	554
		IS	56	56	56	43	39	40	67	614
	convo	no IS	-	-	-	37	39	38	82	287
		IS	-	-	-	**71**	**74**	**72**	**93**	314
	machine		**71**	**70**	**71**	58	56	57	82	-
Media-Wiki	link	no IS	45	45	45	29	32	30	67	416
		IS	58	58	58	21	21	21	**67**	567
	convo	no IS	-	-	-	22	26	23	63	360
		IS	-	-	-	14	13	12	57	363
	machine		**64**	**66**	**64**	**28**	**32**	**27**	60	-
Ubuntu-Meeting	link	no IS	57	56	57	6	10	8	61	594
		IS	53	52	52	40	40	40	**74**	565
	convo	no IS	-	-	-	21	25	22	72	260
		IS	-	-	-	33	39	36	64	753
	machine		**59**	**59**	**59**	**50**	**50**	**50**	**74**	-

(b) Averages after removing outliers

Figure 5.11: Average across all files for the non-expert MTurk worker study. Although machine performance beats that of the non-experts in all four of the "Link" mode conditions, crowd worker performance matches or beats that of the machine's in three domains: `rust`, `mediawiki`, and `ubuntu-meeting`. This shows that non-expert contributors can be beneficial for this conversation disentanglement task.

5.9.3 Crowdworker Feedback

Since we did not collect information from the MTurk workers that would correlate their post-task survey with their Worker ID, we unfortunately cannot directly map comments to the participant IDs. However, we present worker comments to better understand how they perceived both the task and the interface.

Task Perception

As expected, worker responses for how they felt about the task ranged from "VERY EASY" to "nothing was easy, all of it was hard." What is surprising is how confident most workers seemed to be in their annotations. In the post-task survey, about 44% of workers said that they were confident they got a majority of the annotations correct, with 23% saying they were confident they got most of the answers correct. A further 26% were only half-confident in their responses. Only 2% of workers said that they were completely unconfident in their responses. Clearly, worker perception of their effort is that this task is doable, even though it requires domain expertise.

Worker comments also track with those left by the expert annotators: at the beginning, knowing what to do with the task was difficult, but after a couple of lines of annotations, the task became clearer. Finally, workers also had the same questions of ambiguity with respect to the server / bot logs, and also wished for clearer instructions.

Interface Perception

Most of the workers mentioned that the interface was easy to use and that they liked using it. One worker, for the No IS interface, wrote, "I found the task to be a bit hard. There wasn't enough visuals in the interface to show me which sentences I selected, how many times I selected them, and which sentences I linked them to." Said another worker, "The suggestions were a helpful starting point as well as how easy you made it to identify the individual speakers."

What is interesting, however, is that none of the comments mentioned anything about second-guessing or doubting the suggestions made by the machine, something that the expert annotators had mentioned. That is, crowd workers appreciated the highlighting features and the system predictions ("prediction coloring was helpful"), but did not say anything about potentially distrusting those suggestions.

5.10 Discussion

In this section, we expand upon some observations made during the Results analysis sections.

5.10.1 Hypotheses

We revisit the three hypotheses from Section 5.6.1:

- H1: Using IS tools will improve the accuracy of both the Convo and Link task modes. Based on the improvement in performance using IS tools, as seen in Table 5.4, we can *support* this hypothesis.
- H2: Using IS tools will lead to decrease in annotation time. We do not see any time improvements for IS tool usage, so we *do not confirm* this hypothesis.
- H3: Using IS tools will lead to an annotator to go faster the further along a file they are. As we see in the figures mentioned in Section 5.7.6, there is no indication that annotators get faster as they move along. As a result, we also *do not confirm* this hypothesis.

5.10.2 Jargon vs. Accuracy

As mentioned before, the machine seems to do better in the `Stripe` and `Rust` files than in the `Ubuntu-Meeting` and `MediaWiki` files. One potential explanation for why we believe this occurs is because people might not be familiar with jargon, and so might take more time and/or be more confused when reading these messages. Since our machine models have been trained on messages from the #Ubuntu channel, these models might be better at disentangling in those channels whose messages resemble those from the #`Ubuntu` channel. However, if people are less distracted or confused by the jargon, they can better disentangle the conversations based on other relevant context, such as that found in the natural language around the jargon itself. If we can better profile and preanalyze the entangled text, perhaps we can create human-machine pairings to create even better accuracy outcomes.

5.10.3 Annotator Feedback

In Section 5.8.2, we saw a few themes emerge from the post-task surveys. Here, we address potential ways to ameliorate annotator concerns.

Task Difficulty

- Understanding task rules: One way to lower the barrier to entry for this task can be to make the interactive tutorial longer, or to have annotators do a tutorial on more than just one sample file, so that the 'rules" become clearer. Because there are a lot of edge-cases that can come into play, another option is to make the rules easier (e.g., if there are certain edge cases that do not impact the disentanglement outputs, then having more flexibility with that rule can reduce the number of rules annotators need to remember through this process.)

- Parsing technical jargon: A few annotators mentioned not having familiarity with the jargon seen in the text. One way to help familiarize annotators can be to provide examples or a dictionary where annotators can look up what these unfamiliar terms might mean. In a large-scale annotation effort, having the ability for annotators to ask follow-up questions with the channel owner can also be helpful.

- Picking up the conversational context: This particular theme is hard to immediately overcome, as gaining context involves spending some time with the conversations to start figuring out topics, number of threads, etc. One way to make this effort easier is to ask annotators to focus on just one conversation; that way, they can scan only for the particular conversation they've been assigned and will not need to keep track of other threads. Another approach, as suggested by P6, is to provide short summaries of the files before annotators start the task, to better help their mental-model formations. As P6 wrote, "I wish I could've had a sense of a general topic of the conversation (e.g., this is about meetings or Ubuntu or whatever) just to have a better sense of the terminology to look for as I was about to start the task."

- One-to-many responses: Similar to the context theme, it is difficult to easily disentangle messages from users that speak to many other users. A way to mitigate this burden can be to better highlight related messages not just based on the users, but also the content within those messages.

- Bots created additional confusion: Annotators were confused as to how to treat Bot messages. Although we had examples in the instructions for what to do with "ubottu," an Ubuntu bot, it is clear that annotators could have benefitted with more examples.

Interface Difficulty

Annotators using the IS interface had positive views towards the interactional slingshots present in them, but the usability of the interface could still be improved. In particular, we restricted the ability to undo an annotation to be for the most-recent-annotation only, rather than have the ability to undo any prior annotation. This was a design decision to remove freedom from the annotator's side (to avoid ill-effects that arose when annotators had more freedom, as seen in Chapter 3. However, since these annotators have domain knowledge, perhaps relaxing this constraint can lead to better outcomes. Moreover, improving instructions and providing video examples can also help annotators better navigate the interface.

Cognitive Load

Even given difficulties faced, we find that our interface helps annotators to better disentangle files in channels that are harder in nature (e.g., MediaWiki). In the harder tasks, there are more undirected messages, fewer users, and more errors made by the machine suggestions; nevertheless, we find higher accuracies. One potential explanation for this is that, although people took more time to do the annotations, perhaps this slowdown helped annotators think more critically about the task itself.

This leads us to posit that the expert annotators were dealing with a case of managing their cognitive load [15]: intrinsic cognitive load of the task itself being difficult, and extrinsic cognitive load from how the interface was guiding their interactions. Not only is the task itself difficult, but the annotators had to use the interactional slingshot support to evaluate whether the guidance was accurate or not, and then make an annotation. P16 expresses a sentiment shared by a few other annotators, namely the issue of trusting the machine's suggestions. They write, "I think the highlighting was done well, that helped a lot. The highlights actually I found to be wrong most of the time. I honestly relied on the time stamps, those disentagled [*sic*] it for me more than anything else."

Managing the two intrinsic and extrinsic sources of cognitive load can also help explain why the experts found less success in the jargon-heavy files, since now the annotators have to expend more effort to understand the different threads and content taking place in the file. This makes them more prone to errors, something that can also be explained by thinking about this slowdown in the Expert Reversal effect [50]. Experts have the externally-provided guidance from the interactional slingshots, but also pull into working memory their domain knowledge. By relying on their domain knowledge store, the expert annotators were slowed down more when they had to double check the system's predictions. As a result, the effort to combine these two knowledge structures causes cognitive overload and can also cause slowdowns.

On the other hand, there was no indication from the non-expert annotators that they were being slowed down by the external cognitive load, apart from the start where most workers were unsure what to do. As one of the key points of Chapter 2 in *How People Learn* states, "Experts notice features and meaningful patterns of information that are not noticed by novices" [14]. Because non-expert annotators do not know the subject matter as well, perhaps they are less affected by the system predictions being erroneous, and so their cognitive load was less impacted as they do this novel task.

A design point to take away from this is that, if experts are working on the task, interactional slingshots that are providing support in the form of guidance could throttle back their support to be less-guidance driven, and more along the nudging type of support.

5.10.4 Time Spent On Task vs. Accuracy

To examine whether annotators performed better on the task if they spent longer on the document, we plotted time on task versus 1-1. Figure 5.12 shows the scatterplots for "Convo" mode, and Figure 5.13 shows the scatterplots for "Link" mode.

For the "Convo" task, it is interesting to see a moderate correlation inverse correlation between time spent in the Convo-No-IS condition to the 1-1 score for the `ubuntu-meeting` and `mediawiki` domains. However, in the Convo-IS condition, we see a relatively weaker correlation between time spent in this task condition versus the Convo-No-IS condition. One reason for this positive correlation for the two harder domains is that, though the annotators take more time in the Convo-IS condition, they are doing the task more carefully.

For the "Link" task, in the Link-No-IS condition, there is effectively no correlation between time spent on task and 1-1 score. However, for the Link-IS condition, there is a slight positive correlation for the `rust` domain, but there is not enough evidence to state that spending longer on the file leads to higher annotation scores.

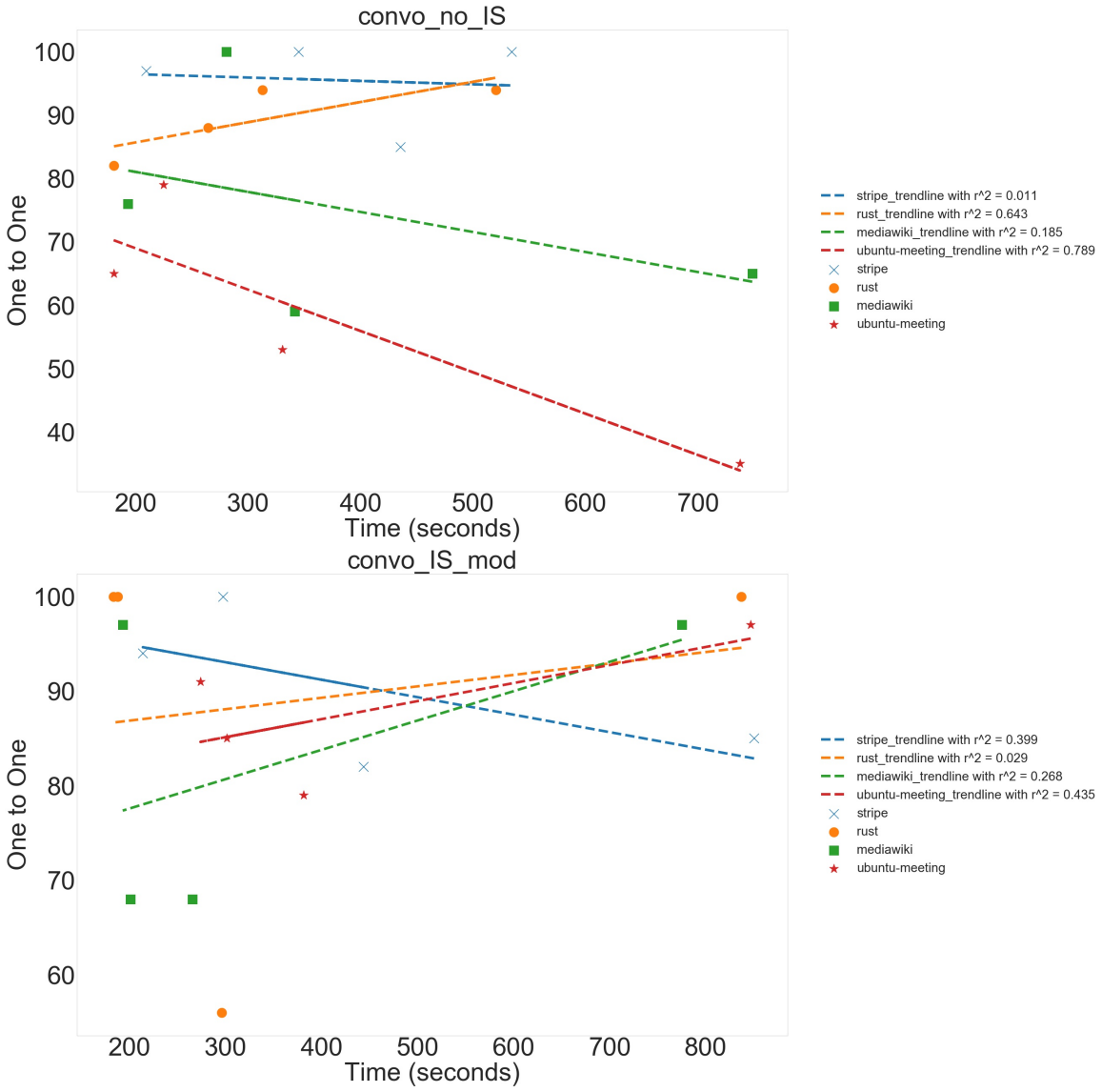


Figure 5.12: Time vs. 1-1

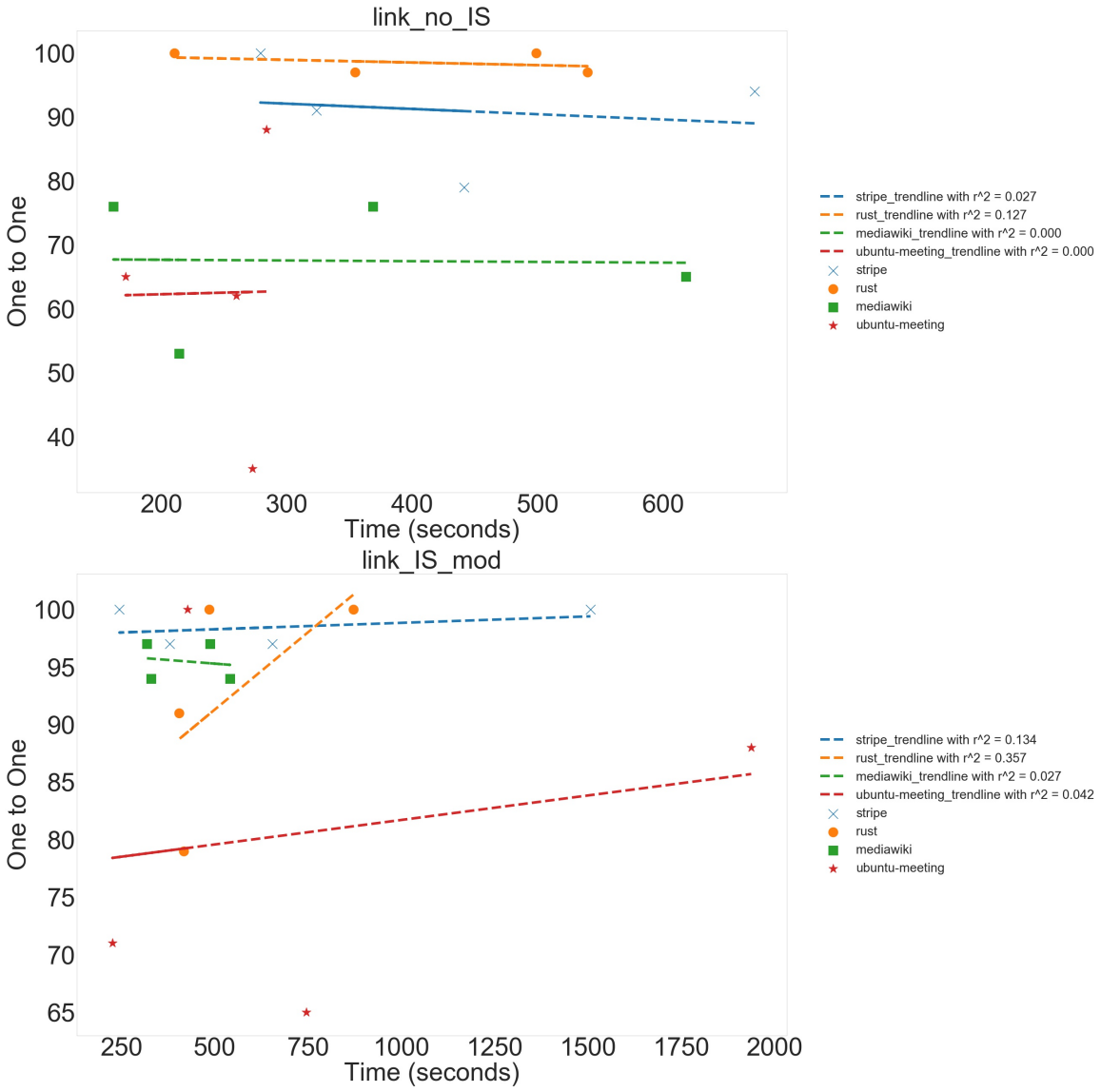
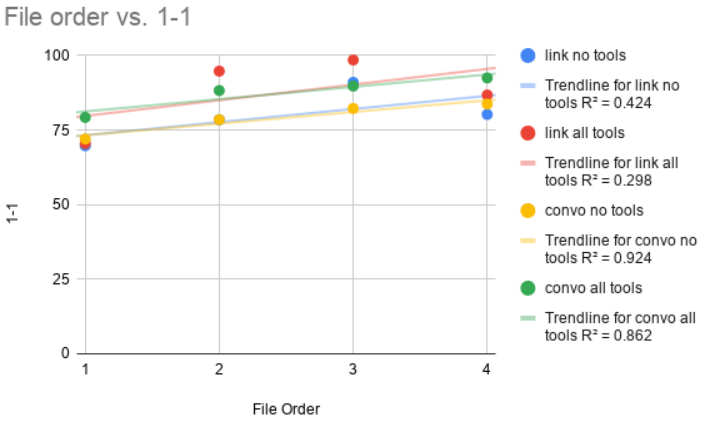


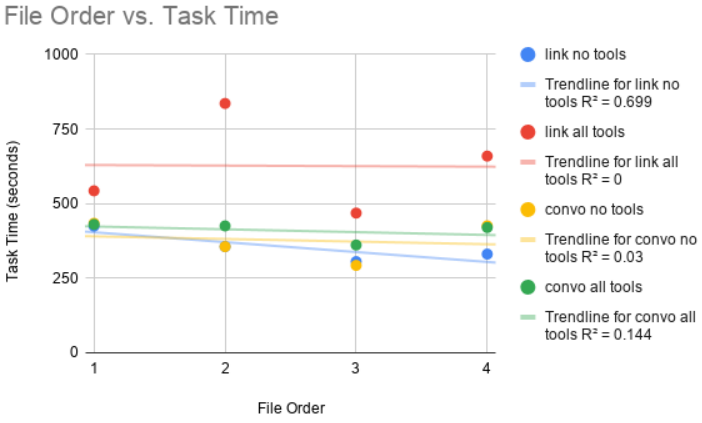
Figure 5.13: Time vs. 1-1, cont'd.

5.10.5 Are There Any Learning Effects?

Even though we randomize the order of files seen by the annotators such that, for each condition, a file appears in that position only once, we are still interested to see if there are any learning effects taking place. Figure 5.14 shows the effect of file order on accuracy (1-1) and time (Task time). For 1-1, we see that, for "Convo" mode, annotators seem to have higher accuracies the more time they spend doing annotations. However, for "Link" mode, that correlation is weak. For Task time, there is no correlation between subsequent annotations and how long they spend doing the task.



(a) 1-1



(b) Task Time

Figure 5.14: File order versus accuracy and time measures. For 1-1, we see that, for "Convo" mode, annotators seem to have higher accuracies the more time they spend doing annotations. However, for "Link" mode, that correlation is weak. For Task time, there is no correlation between subsequent annotations and how long they spend doing the task.

5.10.6 Link Mode vs. Convo Mode For Disentangling Chats

Elsner and Charniak, in the Future Work section in [30], write the following:

> Although our annotators are reasonably reliable, it seems clear that they think of conversations as a hierarchy, with digressions and schisms. We are interested to see an annotation protocol which more closely follows human intuition. One suggestion (David Traum, personal communication) is to drop the idea of partitioning entirely and have annotators mark the data as a graph, linking each utterance to its parents and children with links of various strengths.

Although our "Link" mode does not specify a strength to the links, we can still do a comparison between the "Convo" and "Link" modes. Based on our results, we can confirm their intuition that using the "Link" task leads to higher quality annotations.

5.11 Conclusion

By using interactional slingshots to guide user interactions for a task that requires domain knowledge and is difficult across many dimensions, we show that we get more accurate annotation output than using an interface without guidance. Upon further breakdown of the annotator performance, we note that there are cases where the machine outperforms the human annotations, and discuss implications of this. For instance, annotators outperform the machine in domains where there is less jargon, as well as domains where the conversation characteristics more complex (e.g., fewer directed messages, fewer users, more bots). Cognitive load plays a role in the annotator experience, and we find that expert annotators do not fully trust the suggestions provided by an automated system.

Even with interactional slingshots, annotators find the task difficult, with ambiguity that can be overcome with future studies, along with some difficulty that is not often easy to overcome. For instance, based on annotator feedback, we find that the startup cost to developing context before starting the conversation disentangling is almost-always high. Future work directions can look into how to ameliorate this and provide context-loading speedups. Moreover, because we find a signal that non-expert annotators are able to do this task successfully, future work can explore pairing up non-expert, expert, and machine annotators to leverage each group's strengths for even higher quality annotations.

Chapter 6
Conclusion and Future Directions

This dissertation shows that adding support structure to user interactions—in the form of interactional slingshots—can make annotators more efficient in data annotation settings. Specifically, we have proposed and explored the following thesis statement:

> *By providing support structure that nudges, assists, and guides user interactions, it is possible to create hybrid intelligence systems that enable more efficient (faster and/or more accurate) data annotation.*

The interactional slingshot workflows that we introduce can serve as the basis for thinking about how to provide support to user interactions in hybrid intelligence systems. In this chapter, we briefly restate our evaluation of that thesis statement, and discuss some implications of this work, for data annotation and beyond.

6.1 Research Questions: A Summary of Findings

As shown in the thesis, we believe that adding support in the form of nudging, assistance, and guidance form the beginnings of interactional slingshots that help users annotate data more efficiently and accurately. We believe these forms of support to be three instantiations of interactional slingshots that can help user interactions in hybrid intelligence systems.

6.1.1 Support as Nudges for Collective Conversational Memory

[RQ1]: *For a task that relies on extracting latent mental models from each annotator, what challenges arise when providing support as a form of* nudging*?*

Predicting what information is going to be useful in future interactions for conversations is a challenging task to explicitly capture, even though humans do this curation naturally and subconsciously. By providing nudging support that reminds crowd workers of a fixed time horizon for their notes-creation, we introduce Mnemo, a tool and workflow that allows non-expert crowd workers to collectively select, summarize, and annotate conversational content.

We show that, by nudging non-expert annotators' user interactions, we are able to tease out their latent mental models for how they create notes from conversations. These worker notes can be aggregated to better predict facts that will be relevant in the future, leading to more accurate results. Across 10 dialogs, workers generated 500 notes for collective memory, with individual precision and recall being 44% and 42%, respectively. By aggregating across worker performance, we exceed 90% recall with just five workers. Though our methodology shows that aggregating over different crowd workers' notes can boost recall, we find that precision suffers.

We claim that nudging as a form of support cannot overcome the annotation freedom given to these crowd workers, which can lead to output annotation variability. That is, nudges remind workers to keep in mind the fixed time horizon for the facts that they create in their own words; however, because nudging support does not restrict the "own words" of these workers, the notes that are created cover a multitude of time spans, leading a loss in precision as compared with a gold-standard set of notes.

However, we characterize worker errors and show that they are often not true errors: the loss in precision is not because of superfluous notes created by the workers, but is because of different categories of worker notes. To improve precision and recall, we add a more-explicit nudge at the start of an interaction, leading to a 37% increase in precision and 44% increase in recall. By associating content with utterances within a dialog, we enable automated approaches to recall facts with a non-expert curated knowledge base of important notes, enabling our approach to scale to any length of conversation history.

6.1.2 Support as Assistance for Grounding Natural Language Object References in 3D Scenes

[RQ2]: *For a task that involves dealing with 3D spatial ambiguity, how can interactional slingshots that provide support in the form of* assistance *work with annotators?*

Automated approaches' ability to automatically ground and identify natural language references to objects in 3D scenes fails when these approaches encounter never-before-seen or diverse environments. This creates a barrier to creating and deploying such systems, including autonomous robots, in the wild. To overcome this problem, we introduce EURECA, the first mixed-initiative hybrid intelligence system that leverages real-time crowdsourcing to bridge the gap between understanding visual scenes and natural language references to objects in them.

With support in the form of assistance, we show that EURECA enables non-expert crowd workers to annotate 3D points clouds more accurately and with less time when compared with a baseline interface. Because interactional slingshots like automated filtering and filling of additional points boost each interaction the user has with the data, we show that this leads to a substantial reduction of annotation time, from 85 seconds without interactional slingshots to 58 seconds with. EURECA also facilitates collaboration between sets of online crowd workers, and with just three workers' efforts being assisted and coordinated, EURECA achieves high precision (84%) and recall (92%) with a latency of 26.5 seconds per object.

To ascertain whether our system helps bridge the aforementioned lack of real-world deployability, we test EURECA with several case studies approximating real-world conditions. These tasks include: successfully getting crowd workers to segment a deformable object (scarf) among a scene full of multiple deformable objects (gloves, backpack, toy); including RGB color information in the scene; and, enabling a Fetch robot to pick up a previously-unknown spray bottle based on crowd worker annotations. We show that assistive slingshots make it possible to deploy robots that reliably operate in real-world settings, all the while collecting training data that can help to gradually automate these systems in the future.

6.1.3 Support as Guidance for Multi-Domain Conversation Disentanglement

[RQ3]: *For a task that is difficult and requires expertise, how effective are interactional slingshots that provide support by* guiding *interactions*?

Millions of lines of human-human dialog exist in the form of Internet Relay Chat (IRC) logs, yet automated methods are incapable of easily disentangling these overlapping messages. Expert users can disentangle these IRC logs and build channel-specific disentangling models, but each domain requires effort that does not scale with size of the logs. We introduce MDCD, an interface that provides guided support to expert and

non-expert users to disentangle IRC logs across multiple domains. We show that people with and without domain knowledge are able to use our interactional slingshot-equipped tools to annotate conversational data more accurately than when compared with a baseline. We study this conversational disentanglement task with two task modes across four domains: "Link," where annotators create a reply-to graph structure, and "Convo," where annotators separate messages into constituent conversations.

When the slingshots guide user interactions, annotators achieve higher performance on all four of the channels when averaged across all files. Compared with the machine's performance, the interfaces with guidance improves Exact Match F1 score by 21%, One-to-One by 15%, and Link F1 score by 16%. (See Section 5.6.6 for technical detail regarding these measures.) We find that the guided interfaces improve annotation performance on those channels that are more difficult for annotators. Furthermore, we conduct a study with non-experts recruited from Amazon Mechanical Turk and find crowd workers are capable of outperforming the machine with guidance on three of the four channels, depending on the particular task.

We discuss how guidance as the form of support can be more effective for non-experts and experts alike, although with important differences. For non-experts, because their lack of domain knowledge renders them less likely to notice when the machine is incorrect, they can rely more easily on the machine suggestions and benefit from guidance. However, expert annotators do tend to notice mistakes made by the machine, and so take longer to complete the task, potentially making it harder for them to trust the machine.

6.2 Reflections and Future Directions for Annotation-Related Interactional Slingshots

We comment on some interesting future directions that can tackle challenges that still remain in providing support for user interactions in hybrid intelligence systems.

6.2.1 Support Modalities Across Multiple Dimensions

Prior work shows that leveraging human intelligence in human-in-the-loop, active learning, and human computation can lead to much improved algorithms. In this dissertation, we focus on the human side and have argued that three natural ways of adding support structure to user interactions in hybrid intelligence system are support as nudging, assisting, and guiding. There is an implicit question within this support arrangement,

Task Complexity	Interaction Complexity	Annotator Expertise	AI System's Context	Support Modality
Low	Low	Non-expert	None	Nudging
Medium	High	Non-expert	Partial: can select clusters, even if they are incomplete	Assisting
High	Low-to-Medium	Non-expert and Expert	Full: can predict links and conversation snippets, but trained in only one domain	Guiding

Table 6.1: Important dimensions that underlie annotation tasks and their respective support modalities. E.g., As seen in the Nudging case, when the system does not have much context for how to do the task, the ensuing support mode becomes less intrusive.

namely: Why are these three support modalities useful for annotation tasks? We claim that these three modes are natural divisions that occur when thinking about providing support to users, not just for interactions but providing support in general. In the way that a schoolteacher might nudge, assist, or guide a student as they learn new material, so do these support mechanisms help users as they learn new interfaces.

What is important for deciding when to use these support mechanisms is to think about important dimensions that underlie annotation tasks. For this dissertation work, as seen in Table 6.1, there are four that we focus on: task complexity, interaction complexity, annotator expertise, and context embedded in the AI system. Different support might be required depending on how complex the task is or how complex the actual interactions with the system are, and similar considerations exist for the other two dimensions.

As an example, for the memory curation task, the AI system does not know anything about solving the task and lacks context necessary for saving future-relevant notes, and so it cannot provide assistance or guidance. The context for solving this task lies wholly with the human annotators, as does the expertise. However, AI methods can aggregate over worker inputs, so Mnemo's computational tools (slingshots) offer support only in the form of nudging annotators into curating future-relevant notes.

However, for the 3D point cloud segmentation task, interactions become more complex (they take place in a 3D space after all), the natural language grounding expertise lies with the human, but the AI system does contain context, which is how to partition the point clouds based on existing computer vision information (the *a priori* clusters). However, it cannot solve the entire task of associating objects with the natural language references, so the interactional slingshots in this case not only need to help non-experts

overcome complex data transformations in 3D space, but also leverage their expertise in recognizing what objects exist in the scene. For this task, nudging itself might not work because these interactions require more powerful support. Assistance becomes collaboration as a means of helping non-experts to overcome the complexity.

What happens, then, when the task itself is complex, requires domain knowledge, and the AI system has full context (but only trained on one domain)? Nudging is not powerful enough, and assisting runs the risk of having annotators constantly correct mistakes made by the AI system (if we approach this annotation in the mixed-initiative way as we did with EURECA). For the conversation disentanglement task, rather than nudging or assisting, guidance seems the most natural fit, since the context embedded in the AI system models can be leveraged to guide user interactions, but the system itself doesn't directly annotate the data since it might be wrong. The expert users have the freedom to choose for whether to rely on the system's guidance or not. If they feel that the system is making mistakes, they can still rely on the other interactional slingshots to receive boosts for doing their annotations.

6.2.2 What Support Modality Works Best for What Type of Task?

Each support modality brings with it a varying level of intrusiveness with respect to the user's interaction with the system, and ultimately, their data annotations. Nudging is least intrusive, assistance is most intrusive (since the system actually annotates the data after the user is done), and guidance is in the middle. The more intrusive the support modality, the more cognitive load being placed on the annotator, since now they have to consider whether the system was accurate or not.

Furthermore, there is yet another dimension that exists when human and machine collaborations exist, which is the matter of *trust*. Do the humans trust the machine's annotations? We have seen with EURECA that non-expert annotators learn to work with the automated filter and fill methods that modify the users' initial selections. But, in the conversation disentanglement task, expert users often doubt the validity of the machine suggestions, whereas non-experts do not do so as frequently.

Because of this added cognitive load and the extra dimension of trusting the system, it is possible that the annotator is slowed down, leading to inefficient data annotation. We claim that, if annotation speed is the most important consideration, then assistance helps. This is because the AI system knows how to partially solve the task, so the user's interactions can benefit from the system's assistance. However, a drawback to this pairing is that accuracy could suffer: people end up trusting the machine's annotation output, but this is not helpful if the computer models are not accurate themselves.

On the other hand, if accuracy matters (let's say, for a complex task such as conversation disentanglement), we claim that guidance is an optimal pairing. It might be difficult for annotators to trust the AI suggestions fully, but given guidance from those suggestions and other interactional slingshots, the annotators can be in a better position to make more accurate annotations. This is at the cost of speed, as we have seen, since the extra cognitive load can slow the experts down. If both speed and accuracy matter for the task, a combination between all three—non-experts, experts, and machine—might be required. We speculate on this might entail in the next section.

6.2.3 Setting and Curating Contexts to Jump-Start Annotators

One theme that consistently arose from annotators across the hybrid intelligence systems discussed in this dissertation is that of taking a long time to get started on a task. There remains a challenge to devise better slingshots that can reduce this barrier of task entry for the annotator. One approach can be to enable annotators to collaborate with each other; that is, if more than one annotator works on the same task, a dialog could be established, say via chat, that can help the annotators share knowledge and more quickly curate the context needed to solve the problem.

6.2.4 Dynamically Changing Support Structures

The interactional slingshots used in this thesis were all static in nature. That is, they are the same whether for experts or non-experts. Future work directions can explore what it means for slingshots to be dynamically updated based on task and annotator characteristics. For instance, given the well-studied phenomena of cognitive overload and the Expertise Reversal effect [50] for experts (where guidance can end up negatively impacting expert learners) interactional slingshots can dynamically change the type of support they provide. Rather than support user interactions with just guidance, interactional slingshots can "fade away" over time, or morph into the nudge support.

6.3 Beyond Data Annotation in Hybrid Intelligence Systems

We have seen how the four dimensions of task complexity, interaction complexity, expertise of the annotator, and the context held by the AI system lead to different support modalities, but this was still centered around data annotation tasks. If we move into applications that are beyond data annotation, how does the interplay between these dimensions play out? What new dimensions come to the fore?

For one, even though data annotation has multiple dimensions, hybrid intelligence systems still have to face with two that are endemic to human and machine collaborations: how trust can be developed between the two parties, and how intrusive the AI system is. The more intelligence that is behind the AI system, the more important trust becomes. If the AI system can solve part of the task (not just data annotation), then burden is being placed on the user of the hybrid intelligence system to then trust the AI system. Trusting the machine's ability to solve the problem becomes harder if the user of the system is an expert, and especially becomes tougher the more intrusive the type of support that is being provided by the hybrid intelligence system. Otherwise, providing support to expert users that are distrustful of the support can lead to suboptimal outcomes.

The onus is on the hybrid intelligence system's creators to ensure that whatever support modality they use can help develop trust between the user and the system, as well as make sure that the support's intrusiveness will not slow the users down. Identifying ways of doing this are left as studies for future work. If trust is established, more powerful forms of support can be deployed, including assistance in which the human and machine take turns solving the task at hand.

6.3.1 Hybrid Intelligence Systems Task Taxonomy

As a helpful comparison point, we comment on slingshot support as it pertains to four generic categories of hybrid intelligence system tasks, based on the taxonomy devised by Dellermann et. al [24]: *recognition* tasks, where the primary objective is to recognize objects, images, or natural language; *prediction* tasks, where the aim is to predict future events based on previous data such as stock prices; *reasoning* tasks, where the focus is to understanding data, for instance, by building mental models; and, *action* tasks, where the objective is to conduct a certain kind of action by an agent (human or machine). Based on this taxonomy, this dissertation's projects can be classified as annotations for a prediction task (Mnemo), recognition task (EURECA), and reasoning task (MDCD).

For these four task categories, the three interactional slingshot support approaches that we present—nudging, assisting, and guiding—can still be helpful, especially because the dimensions that underlie data annotation tasks also exist in other tasks. However, for each of these categories, other dimensions can become more important, changing the importance of the support being provided.

In the remainder of this chapter, we provide some motivating examples of tasks that exemplify the task taxonomy, and reason about how interactional slingshots could perform in such scenarios.

6.3.2 Motivating Examples

Example: Slingshot Support for Context-Curation Tasks

Curating context is a dimension whose importance that we have already seen for data annotation tasks. Specifically, if the AI system has context that it can bring to the annotation problem, the type of slingshot support that can be afforded to the user can become more powerful. The more context the AI system has, the more towards guidance and assistance the hybrid intelligence system can move.

What about context-curation for non-data annotation tasks? We propose that providing interactional slingshot support to user interactions for such tasks can make it possible to create more complex hybrid intelligence systems, and imagine what this would mean for two tasks: exploratory data analysis and learning in classroom environments.

1) Exploratory Data Analysis So far, the interactional slingshots presented in this dissertation provide workflows that are either non-expert plus machine, or expert plus machine. It would be interesting to evaluate a workflow that can combine experts, non-experts, and machines with the interactional slingshots that can nudge, assist, and guide the various annotators in this workflow. A position paper [40] that we submitted to the Human-Centered Machine Learning workshop shows one such configuration for the task domain of data mining.

Mining massive datasets can benefit from human input, but current approaches require making tradeoffs between overburdening end users or under-informing the system – algorithms become more accurate given more training data, but requiring more exemplars takes significant user effort. We suggest an approach that engages non-expert and semi-expert crowds as a supporting "interface layer" between end users and data mining systems. Leveraging human intelligence will allow systems to answer new types of

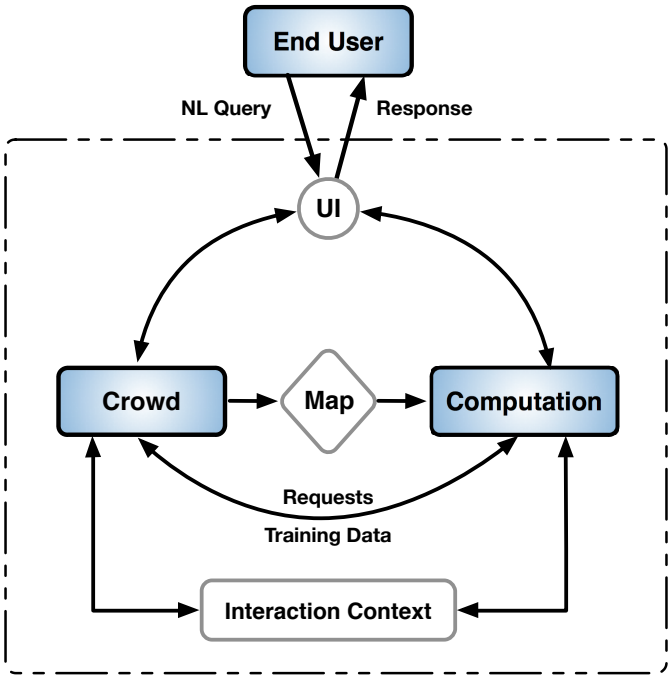


Figure 6.1: A proposed hybrid intelligence system for exploratory data analysis. The system comprises all the elements inside the dotted rectangle. The End User uses natural language queries to interact with the system. The Crowd helps the End User with the data analysis by supporting vague or subjective queries. The UI provides the crowd with analysis tools. By providing *guided* interactional slingshots, we can engage human groups in the analysis process and rely on the system to coordinate those efforts in different ways.

queries (e.g., vague or subjective ones) and generate richer example sets for user-specified patterns. Using crowdsourcing to parallelize this task makes it possible to provide training data to the system in nearly real-time. This allows the system to learn from crowd-generated examples of user-provided instances within the span of a single query.

We predict that the interactional slingshot support most well-suited for this type of recognition task is the *guided* approach. Because this is exploratory analysis, the end user will have an idea of what kind of analysis to do with the data (as they have expertise), and a hybrid intelligence system can guide such data interactions by providing slingshots, for example, that automatically suggest certain data operations. For the crowd workers, guidance can also help them to better understand how to manipulate data, in case they don't already have domain knowledge.

Because end users may construct human-understandable queries, even if algorithms may not understand, new dimensions arise for this hybrid intelligence system: novelty and coordination. Users can construct queries where they describe patterns they wish to find in the data, which means that slingshot mechanisms should guide non-experts

to better understand the novel user queries. Because the crowd can clarify with examples for the end user requests, guided support now must support the ability to coordinate between the end user, crowd, and algorithmic layers. Making available the ongoing interaction context to both the crowd and the data analysis can help the crowd get a better idea of what was queried before, and how it is related to the current query. As depicted in Figure 6.1, this query-response-refine process allows the crowd's insights to become an integral part of the data analysis workflow. The machine's ability to curate context combined with the human's ability to describe data regions of interest can lead to helpful crowd-powered data mining system that leverages slingshot support.

2) Learning in Classroom Environments Context-curation in the classroom environment is also a task that is an exemplar for interactional support, but one that contains a dimension that we did not explore in this thesis: namely, *learning*. Unlike tasks in the annotation domain, tasks in the education domain often require that humans (students) actually learn and obtain new knowledge, not just use the knowledge they already have to annotate data (which was the case with Mnemo and MDCD). However, for such hybrid intelligence systems, slingshot support can still be beneficial, especially with the nudging and assisting supports. For example, education research has shown with overwhelming

Figure 6.2: An example of an AI-based approach that helps students access material from different sources as they learn information.

evidence that explicit and direct instructional guidance are more effective for everyone but experts [19]. Fully-guided instructional support includes lectures, videos, and demonstrations of the problems to be solved. We can imagine hybrid intelligence systems facilitating classroom discussions and helping students by augmenting their classroom activities. Such a system can allow students to leverage context curated from multiple sources—textbooks, online examples, Stack Overflow—and presented to the student as they learn information. An example of an AI-based approach to such a task is planned to be tested at the University of Michigan for the Fall 2020 semester [101]. Figure 6.2 shows a description of their approach, which we imagine can be modified into a hybrid intelligence system.

Moreover, assistive interactional slingshots can take the students' attempts, let's say at coding a problem, and automatically adding and removing errors, but in an instructional way. As Clark suggests, "One of the best examples of an instructional approach that takes into account how our working and long-term memories interact is the 'worked-example effect."' For such hybrid systems, we can leverage computational tools that assist students as they step-through a worked example, thereby leading to better knowledge retention. Once the novice gains more information and expertise, such assistive slingshots can morph into guided slingshots, thereby avoiding the expertise-reversal effect.

Example: Slingshot Support for Accessibility

How can interactional slingshots even begin to tackle tasks that requires recognition, reasoning, and action all at once? An example of such a task having situated interactions for users with motor impairments. Situated interaction leverages a physical environment's context to make communication richer and more efficient between an AI agent and a human user [12]. These interactions leverage gesture and references to physical surroundings, in addition to speech, to make sense of the interaction context. However, these speech- and gesture-based interactions are not always accessible to people with certain types of motor impairments that may reduce their ability to accurately reference an object via gesture, or may result in modified speech patterns.

In a proposed hybrid intelligence system [39], we suggest a direction of work that aims to combine context from multiple interactional sources with collective human intelligence to help overcome these accessibility challenges. As seen in Figure 6.3, it is crucial that the system obtain the ability to understand speech and gesture modalities. For instance, current crowd-powered approaches use context from pairwise intersections between these

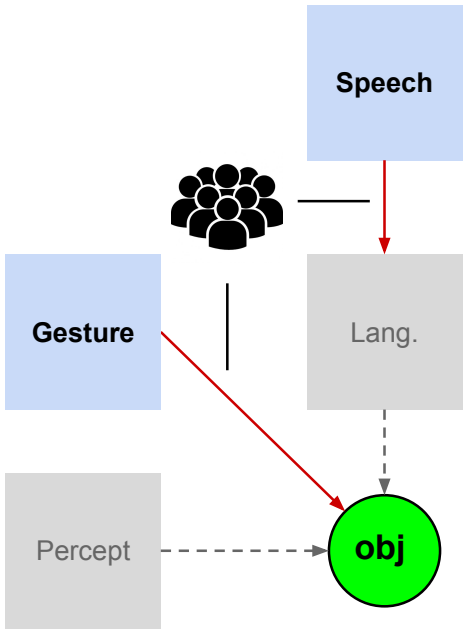


Figure 6.3: General setup of situated interactions. While issues of natural language understanding (Lang.) and perception (Percept) are common, these are the same for motor impaired users as for anyone else, and thus are not our focus here. We believe that crowds provide a powerful and highly available means of addressing challenges in speech and gesture understanding, but new ways to jointly leverage context are needed.

four components to overcome some of these accessibility gaps. If we know that there is a cup in the scene (perception) and the user says "cup" (speech), the system can use that recognized object and interactional slingshots to narrow down the list of object candidates (similar to how language models narrow down candidate words).

Multiple sources of context can help to resolve these complex references, or to further improve people's ability to disambiguate references. However, there remains the key open question of how to present this joint context while avoiding an increase in people's cognitive load, which is an aspect that we encountered in the MDCD project.

Furthermore, there remains an additional open challenge for situated interaction references, as no one has looked at "noisy" gestures for referencing objects in physical environments (though, as an example, Mott et al. [84] looked at this for touch interfaces). This object referencing is itself based on the machine's ability to perceive, which we already explored in EURECA, but not completely resolved, and not when the initial request is low-confidence. This means that slingshots cannot simple nudge user interactions, and need more power, perhaps even assisting users while they perform situated interactions.

However, a drawback with assistance is that, if the system gets the prediction wrong (e.g., the user did not want to touch the part of the interface the system guessed), then the time to complete the task could become too large, frustrating the user and leading to a suboptimal experience.

Example: Slingshot Support for Creative Tasks

Human-AI collaborative systems have had great success when deployed in tasks that require creativity, including for automatically learning about webpage layouts [56], providing support to programmers by reducing coordination costs involved in seeking help [16], using a web browser to facilitate musical performances over a distributed crowd of people [77], and enabling crowd workers to create reusable interactive behaviors easily and accurately [78]. Although these tasks require creativity—a wholly new dimension not seen in the projects discussed in this dissertation—they nevertheless still share some qualities with data annotation tasks.

Namely, tasks that require creativity can benefit from an AI system's context, user expertise, and interaction complexity. We can imagine, for instance, that the different remix behaviors present in [78] can form assistive support, where the system automatically predicts user actions for next steps based on past behavior. Moreover, even though the interactional complexity present in [56] is high, guided support can help an expert user, in this case one that understands CSS and web layouts, to better interact with the tool to automate some functionality. Slingshot supports can provide a net benefit to the user's experience, both if the user is the one being creative (in which case relying on the slingshot support), or if the computer is being creative (such as when neural networks paint in the style of other artists [36]).

6.3.3 Where Slingshot Support Can Fail

The motivation behind this dissertation, which is to provide support to user interactions in hybrid intelligence systems for data annotation, exists because existing AI systems on their own do not cope well with uncertainty, nuance, and complexity. Jarrahi [47] notes that, in decision-making scenarios, humans and AI can collaborate effectively in scenarios that involve uncertainty, complexity, and equivocality. For each category, AI systems and humans bring differing skill sets: for *uncertainty*, humans can make swift decisions in the face of the unknown, and AI can provide access to realtime information to facilitate those decisions; for *complexity*, humans decide where to seek and gather data, and

AI systems can collect, curate, and analyze this data; finally, for *equivocality*, humans can build consensus and rally support, and AI can analyze sentiments and represent diverse inputs. For all three categories, we can see how interactional supports can enable hybrid intelligence systems to facilitate this human-AI collaboration. However, under certain circumstances, we speculate that interactional slingshots still might not be enough to overcome challenges.

One such challenge is something we already explored in this dissertation: when the AI does not have any task-solving context, such as in Mnemo, it becomes difficult for interactional support to be of any other form than nudging. And if the human being is an non-expert, then it can make it difficult for this type of interactional support to actually help the user. Along all of the three axes of uncertainty, complexity, and equivocality, the combination of a non-expert and nudging might not work well.

Another challenge that is difficult for interactional slingshot-based support to overcome is if the task is intolerant of error. The data annotation workflows and systems presented in this thesis perform quality control on the output data annotation, such as aggregating input across all annotators. However, if the task is in a domain where making mistakes can be critical to the task objective, such as in the medical domain, then interactional supports such as nudging, assisting, and guiding might not be enough to guarantee mistake-free outcomes. Future work can explore new support modalities that can overcome this challenge.

In a similar vein, interactional slingshots as presented in this dissertation might not work well if the task type requires extremely low latency. The data annotation tasks that we explored in this dissertation have a relatively high tolerance for time; except for EURECA, where the wall time for completing the end-to-end test with a Fetch robot matters, the time to task completion is not critical to solving the task itself. That is, even though time per annotation matters, it is not fundamental to the solving the task itself. However, for tasks like anomaly detection, latency matters: if the interactional support given to the user's interaction takes longer than the allowed time bounds, not only will the user have a frustrating experience with the system (because the support doesn't arrive in time), but also the support afforded to the user will not help them do the task well.

What this implies is that, although interactional slingshots are well suited to support user interactions across a diverse set of tasks, for the best performance, careful thought needs to go into design decisions that impact what the human will do, what the machine will do, and how slingshots can best facilitate this collaboration.

Bibliography

[1] M. S. Ackerman. The intellectual challenge of cscw: the gap between social requirements and technical feasibility. *Human–Computer Interaction*, 15(2-3):179–203, 2000.

[2] M. S. Ackerman, J. Dachtera, V. Pipek, and V. Wulf. Sharing knowledge and expertise: The cscw view of knowledge management. *Computer Supported Cooperative Work (CSCW)*, 22(4-6):531–573, 2013.

[3] P. H. Adams and C. H. Martell. Topic detection and extraction in chat. In *2008 IEEE international conference on Semantic computing*, pages 581–588. IEEE, 2008.

[4] J. F. Allen, D. K. Byron, M. Dzikovska, G. Ferguson, L. Galescu, and A. Stent. Toward conversational human-computer interaction. *AI magazine*, 22(4):27, 2001.

[5] AmazonMechanicalTurk, 2005. Accessed: 2017-04-04.

[6] R. Baecker, D. Fono, L. Blume, C. Collins, and D. Couto. Webcasting made interactive: Persistent chat for text dialogue during and about learning events. *Human Interface and the Management of Information. Interacting in Information Environments*, pages 260–268, 2007.

[7] M. S. Bernstein, J. Brandt, R. C. Miller, and D. R. Karger. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 33–42. ACM, 2011.

[8] M. S. Bernstein, D. R. Karger, R. C. Miller, and J. Brandt. Analytic methods for optimizing realtime crowdsourcing. *arXiv preprint arXiv:1204.2995*, 2012.

[9] M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich. Soylent: a word processor with a crowd inside. *Communications of the ACM*, 58(8):85–94, 2015.

[10] F. Bessho, T. Harada, and Y. Kuniyoshi. Dialog system using real-time crowdsourcing and twitter large-scale corpus. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 227–231. Association for Computational Linguistics, 2012.

[11] J. P. Bigham, C. Jayant, H. Ji, G. Little, A. Miller, R. C. Miller, R. Miller, A. Tatarowicz, B. White, S. White, et al. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 333–342. ACM, 2010.

[12] D. Bohus and E. Horvitz. Models for multiparty engagement in open-world dialog. In *Proc. of SIGDIAL*, pages 225–234. Association for Computational Linguistics, 2009.

[13] K. Bontcheva, H. Cunningham, I. Roberts, A. Roberts, V. Tablan, N. Aswani, and G. Gorrell. Gate teamware: a web-based, collaborative text annotation framework. *Language Resources and Evaluation*, 47(4):1007–1029, 2013.

[14] J. D. Bransford, A. L. Brown, R. R. Cocking, et al. *How people learn*, volume 11. Washington, DC: National academy press, 2000.

[15] P. Chandler and J. Sweller. Cognitive load theory and the format of instruction. *Cognition and instruction*, 8(4):293–332, 1991.

[16] Y. Chen, S. W. Lee, Y. Xie, Y. Yang, W. S. Lasecki, and S. Oney. Codeon: On-demand software development assistance. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 6220–6231, New York, NY, USA, 2017. Association for Computing Machinery.

[17] H. M. Choset. *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[18] M. J.-Y. Chung, M. Forbes, M. Cakmak, and R. P. Rao. Accelerating imitation learning through crowdsourcing. In *2014 IEEE Robotics and Automation (ICRA)*, pages 4777–4784, 2014.

[19] R. E. Clark, P. A. Kirschner, and J. Sweller. Putting students on the path to learning: The case for fully guided instruction. *American Educator*, 36(1):6–11, 2012.

[20] S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, Z. Popović, et al. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760, 2010.

[21] C. Crick, S. Osentoski, G. Jay, and O. C. Jenkins. Human and robot perception in large-scale learning from demonstration. In *Proceedings of the 6th international conference on Human-robot interaction*, pages 339–346. ACM, 2011.

[22] J. Davies. Science.js. `https://github.com/jasondavies/science.js`, 2011.

[23] G. V. de la Cruz, B. Peng, W. S. Lasecki, and M. E. Taylor. Generating real-time crowd advice to improve reinforcement learning agents. In *Workshops at the 29th AAAI Conference on Artificial Intelligence*, 2015.

[24] D. Dellermann, A. Calma, N. Lipusch, T. Weber, S. Weigel, and P. Ebel. The future of human-ai collaboration: a taxonomy of design knowledge for hybrid intelligence systems. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

[25] J. Deng, J. Krause, and L. Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2013.

[26] K. Desingh, O. C. Jenkins, L. Reveret, and Z. Sui. Physically plausible scene estimation for manipulation in clutter. In *16th International Conference on Humanoid Robots (Humanoids)*, pages 1073–1080. IEEE, 2016.

[27] W. Du, P. Poupart, and W. Xu. Discovering conversational dependencies between messages in dialogs. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[28] A. Dulceanu. Recovering implicit thread structure in chat conversations. *Revista Romana de Interactiune Om-Calculator*, 9(3):217–232, 2016.

[29] M. Elsner and E. Charniak. You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of ACL-08: HLT*, pages 834–842, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[30] M. Elsner and E. Charniak. Disentangling chat. In *Computational Linguistics*, volume 36, pages 389–409, 2010.

[31] M. Elsner and E. Charniak. Disentangling chat with local coherence models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1179–1189, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[32] M. Elsner and W. Schudy. Bounding and comparing methods for correlation clustering beyond ILP. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, pages 19–27, Boulder, Colorado, June 2009. Association for Computational Linguistics.

[33] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard. 3-d mapping with an rgb-d camera. *IEEE Transactions on Robotics*, 30(1):177–187, 2014.

[34] D. Fono and R. Baecker. Structuring and supporting persistent chat conversations. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 455–458. ACM, 2006.

[35] W. Garage. Pr2 interactive manipulation, 2008. Accessed: 2017-04-04.

[36] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[37] A. Golovinskiy, V. G. Kim, and T. Funkhouser. Shape-based recognition of 3d point clouds in urban environments. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2154–2161. IEEE, 2009.

[38] S. R. Gouravajhala, Y. Jiang, P. Kaur, J. Chaar, and W. S. Lasecki. Finding mnemo: Hybrid intelligence memory in a crowd-powered dialog system. In *Collective Intelligence (CI)*, 2018.

[39] S. R. Gouravajhala, H. Kaur, R. Fok, and W. Lasecki. Challenges in making situated interactions accessible to motor-impaired users. In *Proceedings of the 2018 ACM CSCW Workshop on Accessible Voice Interfaces*. ACM, 2018.

[40] S. R. Gouravajhala, D. Koutra, and W. Lasecki. Towards crowd-assisted data mining. In *Workshop on Human-Centered Machine Learning (HCML) at the SIGCHI Conference on Human Factors in Computing Systems*, 2016.

[41] S. R. Gouravajhala, J. Yim, K. Desingh, Y. Huang, O. C. Jenkins, and W. S. Lasecki. Eureca: Enhanced understanding of real environments via crowd assistance. In *AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, 2018.

[42] U. Hassan and E. Curry. A capability requirements approach for predicting worker performance in crowdsourcing. In *9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 429–437. IEEE, 2013.

[43] M. A. Hearst, J. Allen, E. Horvitz, and C. Guinn. Mixed-initiative interaction. *IEEE Intelligent Systems*, 14(5):14–23, 1999.

[44] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[45] E. Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166. ACM, 1999.

[46] T.-H. K. Huang, W. S. Lasecki, A. A. Azaria, and J. P. Bigham. "is there anything else i can help you with?": Challenges in deploying an on-demand crowd-powered conversational agent. In *The Fourth AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2016)*, 2016.

[47] M. H. Jarrahi. Artificial intelligence and the future of work: Human-ai symbiosis in organizational decision making. *Business Horizons*, 61(4):577–586, 2018.

[48] Y. Jiang, C. Finegan-Dollak, J. K. Kummerfeld, and W. Lasecki. Effective crowdsourcing for a new type of summarization task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 628–633, 2018.

[49] H. J. Jung, Y. Park, and M. Lease. Predicting next label quality: A time-series model of crowdwork. In *Second AAAI Conference on Human Computation and Crowdsourcing*, 2014.

[50] S. Kalyuga. The expertise reversal effect. In *Managing cognitive load in adaptive multimedia learning*, pages 58–80. IGI Global, 2009.

[51] H. Kaur, M. Gordon, Y. Yang, J. P. Bigham, J. Teevan, E. Kamar, and W. S. Lasecki. Crowdmask: Using crowds to preserve privacy in crowd-powered systems via progressive filtering. In *Proceedings of the AAAI Conference on Human Computation (HCOMP 2017)., HCOMP*, 2017.

[52] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 453–456. ACM, 2008.

[53] J.-C. Klie, M. Bugert, B. Boullosa, R. E. de Castilho, and I. Gurevych. The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9. Association for Computational Linguistics, Juni 2018.

[54] J.-C. Klie, R. E. de Castilho, and I. Gurevych. From zero to hero: Human-in-the-loop entity linking in low resource domains. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6982–6993, 2020.

[55] T. Kriplean, M. Toomim, J. Morgan, A. Borning, and A. Ko. Is this what you meant?: promoting listening on the web with reflect. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1559–1568. ACM, 2012.

[56] R. Krosnick, S. W. Lee, W. S. Laseck, and S. Onev. Expresso: Building responsive interfaces with keyframes. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 39–47. IEEE, 2018.

[57] B. Kuipers. The spatial semantic hierarchy. *Artificial intelligence*, 119(1-2):191–233, 2000.

[58] A. P. Kulkarni, M. Can, and B. Hartmann. Turkomatic: automatic recursive task and workflow design for mechanical turk. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pages 2053–2058. ACM, 2011.

[59] J. K. Kummerfeld. Slate: A super-lightweight annotation tool for experts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 7–12, July 2019.

[60] J. K. Kummerfeld, S. R. Gouravajhala, J. J. Peper, V. Athreya, C. Gunasekara, J. Ganhotra, S. S. Patel, L. Polymenakos, and W. S. Lasecki. A large-scale corpus for conversation disentanglement. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3846–3856, July 2019.

[61] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1817–1824. IEEE, 2011.

[62] W. Lasecki, T. Lau, G. He, and J. Bigham. Crowd-based recognition of web interaction patterns. In *Adjunct proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 99–100. ACM, 2012.

[63] W. Lasecki, C. Miller, A. Sadilek, A. Abumoussa, D. Borrello, R. Kushalnagar, and J. Bigham. Real-time captioning by groups of non-experts. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 23–34. ACM, 2012.

[64] W. S. Lasecki. On facilitating human-computer interaction via hybrid intelligence systems. In *2019 Collective Intelligence (CI)*, 2019.

[65] W. S. Lasecki and J. P. Bigham. Automated support for collective memory of conversational interactions. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[66] W. S. Lasecki, M. Gordon, D. Koutra, M. F. Jung, S. P. Dow, and J. P. Bigham. Glance: Rapidly coding behavioral video with the crowd. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 551–562. ACM, 2014.

[67] W. S. Lasecki, E. Kamar, and D. Bohus. Conversations in the crowd: Collecting data for task-oriented dialog learning. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[68] W. S. Lasecki, J. Kim, N. Rafter, O. Sen, J. P. Bigham, and M. S. Bernstein. Apparition: Crowdsourced user interfaces that come to life as you sketch them. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1925–1934. ACM, 2015.

[69] W. S. Lasecki, C. D. Miller, A. Sadilek, A. Abumoussa, D. Borrello, R. Kushalnagar, and J. P. Bigham. Real-time captioning by groups of non-experts. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, 2012.

[70] W. S. Lasecki, K. I. Murray, S. White, R. C. Miller, and J. P. Bigham. Real-time crowd control of existing interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 23–32. ACM, 2011.

[71] W. S. Lasecki, K. I. Murray, S. White, R. C. Miller, and J. P. Bigham. Real-time crowd control of existing interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 23–32. ACM, 2011.

[72] W. S. Lasecki, K. I. Murray, S. White, R. C. Miller, and J. P. Bigham. Real-time crowd control of existing interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 23–32. ACM, 2011.

[73] W. S. Lasecki, Y. C. Song, H. Kautz, and J. P. Bigham. Real-time crowd labeling for deployable activity recognition. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 1203–1212. ACM, 2013.

[74] W. S. Lasecki, P. Thiha, Y. Zhong, E. Brady, and J. P. Bigham. Answering visual questions with conversational crowd assistants. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, page 18. ACM, 2013.

[75] W. S. Lasecki, R. Wesley, J. Nichols, A. Kulkarni, J. F. Allen, and J. P. Bigham. Chorus: a crowd-powered conversational assistant. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 151–162. ACM, 2013.

[76] W. S. Lasecki, S. C. White, K. I. Murray, and J. P. Bigham. Crowd memory: Learning in the collective. *arXiv preprint arXiv:1204.3678*, 2012.

[77] S. W. Lee and A. Willette. Crowd in c. In *Proceedings of the 2019 on Creativity and Cognition*, C&C '19, page 425–431, New York, NY, USA, 2019. Association for Computing Machinery.

[78] S. W. Lee, Y. Zhang, I. Wong, Y. Yang, S. D. O'Keefe, and W. S. Lasecki. Sketchexpress: Remixing animations for more effective crowd-powered prototyping of interactive interfaces. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, page 817–828, New York, NY, USA, 2017. Association for Computing Machinery.

[79] R. Lowe, N. Pow, I. Serban, and J. Pineau. The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294, Prague, Czech Republic, Sept. 2015. Association for Computational Linguistics.

[80] R. T. Lowe, N. Pow, I. V. Serban, L. Charlin, C.-W. Liu, and J. Pineau. Training end-to-end dialogue systems with the ubuntu dialogue corpus. *Dialogue & Discourse*, 8(1):31–65, 2017.

[81] E. Mayfield, D. Adamson, and C. Penstein Rosé. Hierarchical conversation structure prediction in multi-party chat. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 60–69, Seoul, South Korea, July 2012. Association for Computational Linguistics.

[82] M. Meilland and A. I. Comport. On unifying key-frame and voxel-based dense visual slam at large scales. In *2013 IEEE/RSJ Intelligent Robots and Systems (IROS)*, pages 3677–3683. IEEE, 2013.

[83] D. Merritt, J. Jones, M. S. Ackerman, and W. S. Lasecki. Kurator: Using the crowd to help families with personal curation tasks. 2017.

[84] M. E. Mott and J. O. Wobbrock. Cluster touch: Improving touch accuracy on smartphones for people with motor and situational impairments. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–14, New York, NY, USA, 2019. Association for Computing Machinery.

[85] V. Narayanan and M. Likhachev. Discriminatively-guided Deliberative Perception for Pose Estimation of Multiple 3D Object Instances. In *Proceedings of Robotics: Science and Systems*, Ann Arbor, USA, June 2016.

[86] S. Osentoski, C. Crick, G. Jay, and O. C. Jenkins. Crowdsourcing for closed loop control. *Proc. of the NIPS Workshop on Computational Social Science and the Wisdom of Crowds, NIPS*, 2010.

[87] F. Papazian, R. Bossy, and C. Nédellec. Alvisae: a collaborative web text annotation editor for knowledge acquisition. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 149–152, 2012.

[88] C. Papazov, S. Haddadin, S. Parusel, K. Krieger, and D. Burschka. Rigid 3d geometry matching for grasping of known objects in cluttered scenes. *The International Journal of Robotics Research*, page 0278364911436019, 2012.

[89] M. D. Peng Dai and S. Weld. Artificial intelligence for artificial artificial intelligence. In *In Proceedings of the 25th AAAI Conference on Artificial Intelligence, AAAI*, 2011.

[90] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[91] A. Richtsfeld, T. Mörwald, J. Prankl, M. Zillich, and M. Vincze. Segmentation of unknown objects in indoor environments. In *2012 Intelligent Robots and Systems (IROS)*, pages 4791–4796. IEEE, 2012.

[92] M. Riou, S. Salim, and N. Hernandez. Using discursive information to disentangle french language chat. 2015.

[93] P. Roit, A. Klein, D. Stepanov, J. Mamou, J. Michael, G. Stanovsky, L. Zettlemoyer, and I. Dagan. Controlled crowdsourcing for high-quality QA-SRL annotation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7008–7013, Online, July 2020. Association for Computational Linguistics.

[94] O. Russakovsky, L.-J. Li, and L. Fei-Fei. Best of both worlds: human-machine collaboration for object annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2121–2131, 2015.

[95] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1):157–173, 2008.

[96] R. B. Rusu. Semantic 3d object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz*, 24(4):345–348, 2010.

[97] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[98] E. Salisbury, S. Stein, and S. Ramchurn. Crowdar: augmenting live video with a real-time crowd. In *Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.

[99] E. Salisbury, S. Stein, and S. Ramchurn. Real-time opinion aggregation methods for crowd robotics. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 841–849, 2015.

[100] M. Sameki, D. Gurari, and M. Betke. Predicting quality of crowdsourced image segmentations from crowd behavior. In *Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.

[101] P. Samson, 2020. Accessed: 2020-08-12.

[102] D. Shen, Q. Yang, J.-T. Sun, and Z. Chen. Thread detection in dynamic text message streams. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, page 35–42, New York, NY, USA, 2006. Association for Computing Machinery.

[103] J. Y. Song, R. Fok, A. Lundgard, F. Yang, J. Kim, and W. S. Lasecki. Two tools are better than one: Tool diversity as a means of improving aggregate crowd performance. In *23rd International Conference on Intelligent User Interfaces*, IUI '18, pages 559–570, New York, NY, USA, 2018. ACM.

[104] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015.

[105] A. Sorokin, D. Berenson, S. Srinivasa, and M. Hebert. People helping robots helping people: Crowdsourcing for grasping novel objects. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2117–2122, Oct 2010.

[106] I. A. Sucan and S. Chitta. "moveit!", 2013. Accessed: 2017-04-04.

[107] Z. Sui, L. Xiang, O. C. Jenkins, and K. Desingh. Goal-directed robot manipulation through axiomatic scene estimation. *The International Journal of Robotics Research*, 36(1):86–104, 2017.

[108] A. Ten Pas and R. Platt. Localizing handle-like grasp affordances in 3d point clouds. In *Experimental Robotics*, pages 623–638. Springer, 2016.

[109] J. W. Vaughan. Making better use of the crowd: How crowdsourcing can advance machine learning research. *Journal of Machine Learning Research*, 18(193):1–46, 2018.

[110] V. Verroios and M. S. Bernstein. Context trees: Crowdsourcing global understanding from local views. In *Second AAAI Conference on Human Computation and Crowdsourcing*, 2014.

[111] L. Von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM, 2004.

[112] D. Vronay, M. Smith, and S. Drucker. Alternative interfaces for chat. In *Proceedings of the 12th annual ACM symposium on User interface software and technology*, pages 19–26. ACM, 1999.

[113] Y.-C. Wang, M. Joshi, W. W. Cohen, and C. P. Rosé. Recovering implicit thread structure in newsgroup style conversations. In *ICWSM*, 2008.

[114] W. Ward and B. Pellom. The cu communicator system. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, volume 562, 1999.

[115] T. Yan, V. Kumar, and D. Ganesan. Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 77–90. ACM, 2010.

[116] S. M. Yimam, C. Biemann, R. E. de Castilho, and I. Gurevych. Automatic annotation suggestions and custom annotation layers in webanno. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 91–96, 2014.

[117] Y. Zhong, W. S. Lasecki, E. Brady, and J. P. Bigham. Regionspeak: Quick comprehensive spatial descriptions of complex images for blind users. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2353–2362. ACM, 2015.