Schema-Based Dialogue Management: From Friendly Peer to Virtual Standardized Cancer Patient

by

Benjamin P. Kane

Submitted in Partial Fulfillment of the

Requirements for the Degree

Doctor of Philosophy

Supervised by

Professor Lenhart Schubert
Professor Ehsan Hoque
Professor Aaron Steven White

Department of Computer Science
Arts, Sciences and Engineering
Edmund A. Hajim School of Engineering and Applied Sciences

University of Rochester
Rochester, New York

2024

To my parents

Table of Contents

List of Tables

List of Figures

Biographical Sketch

Benjamin Kane was born in Rochester, New York, U.S.A. He was awarded his B.S. in Computer Science by the University of Rochester – where he was employed as an undergraduate research assistant under the mentorship of Professor Lenhart Schubert – in 2019. He continued into his doctoral studies at the University of Rochester, advised by Lenhart Schubert with additional advising from Professor Ehsan Hoque and Professor Aaron Steven White. In 2021, he was awarded the Donald M. and Janet C. Barnard Fellowship by the University of Rochester, as well as his M.S. in Computer Science.

The following publications were a result of work conducted during his studies:

- **Benjamin Kane** and Lenhart Schubert. We are what we repeatedly do: Inducing and deploying habitual schemas in persona-based responses. *EMNLP*, December 6-10, 2023, Singapore.

- **Benjamin Kane**, Catherine Giugno, Lenhart Schubert, Kurtis Haut, Caleb Wohn, and Ehsan Hoque. Managing Emotional Dialogue for a Virtual Cancer Patient: A Schema-Guided Approach *IEEE Transactions on Affective Computing*, January 2023.

- Kurtis Haut, Caleb Wohn, **Benjamin Kane**, Thomas Carroll, Catherine Giugno, Varun Kumar, Ronald Epstein, Lenhart Schubert, and Ehsan Hoque. Validating a virtual human and automated feedback system for training doctor-patient communication skills. *Affective Computing and Intelligent Interaction (ACII)*, September 10-13, 2023, Boston, USA.

- Georgiy Platonov, **Benjamin Kane**, and Lenhart Schubert. Generating Justifications in a Spatial Question-Answering Dialogue System for a Blocks World. *Reasoning and Interaction (ReInAct 2021)*, October 4-6, 2021, Virtual.

- **Benjamin Kane**, William Gantt, and Aaron Steven White. Intensional Gaps: Relating veridicality, factivity, doxasticity, bouleticity, and negraising. *Semantics and Linguistic Theory (SALT) 31*, May 7-9, 2021, Virtual.

- Mohammad Rafayet Ali, Taylan Sen, **Benjamin Kane**, Shagun Bose, Thomas Carroll, Ronald Epstein, Lenhart Schubert, and Ehsan Hoque. Novel Computational Linguistic Measures, Dialogue System, and the Development of SOPHIE: Standardized Online Patient for Healthcare Interaction Education. *IEEE Transactions on Affective Computing*, January 2021.

- William Gantt, **Benjamin Kane**, and Aaron Steven White. Natural Language Inference with Mixed Effects. *The 9th Joint Conference on Lexical and Computational Semantics (*SEM 2020)*, December 12-13, 2020, Virtual.

- **Benjamin Kane**, Georgiy Platonov, and Lenhart Schubert. Registering Historical Context for Question Answering in a Blocks World Dialogue System. *Text, Speech, and Dialogue*, September 8-11, 2020, Virtual.

- Georgiy Platonov, **Benjamin Kane**, Aaron Gindi, and Lenhart Schubert. A Spoken Dialogue System for Spatial Question Answering in a Physical Blocks World. *SIGDIAL*, July 1-3, 2020, Virtual.

- Zahra Seyedeh Razavi, **Benjamin Kane**, and Lenhart Schubert. Investigating Linguistic and Semantic Features for Turn-Taking Prediction in Open-Domain Human-Computer Conversation. *Interspeech*, September 15-19, 2019, Graz, Austria.

- Gene Louis Kim, **Benjamin Kane**, Viet Duong, Muskaan Mendiratta, Graham McGuire, Sophie Sackstein, Georgiy Platonov, and Lenhart Schubert. Generating Discourse Inferences from Unscoped Episodic Logical Formulas. *1st Int. Workshop on Designing Meaning Representations (DMR), at the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, Aug 1, 2019, Florence, Italy.

- Zahra Seyedeh Razavi, Lenhart Schubert, **Benjamin Kane**, Mohammad Rafayet Ali, Kimberly Van Orden, and Tianyi Ma. Dialogue Design and Management for Multi-Session Casual Conversation with Older Adults. *Workshop on User-Aware Conversational Agents (User2Agent), at the 24th Int. Conf. on Intelligent User Interfaces (ACM IUI 2019)*, March 17-20, 2019, Los Angeles, USA.

Acknowledgments

My doctorate research has been characterized by extensive collaboration, and I would not have been able to get to where I am today without the peers and mentors that I've had along the way.

I have had the rare opportunity to receive mentorship from not one but three amazing advisors. First, I would like to thank Prof. Lenhart Schubert, my primary advisor who supported my academic endeavors from the very beginning. He is a veritable treasure trove of knowledge from an era of Artificial Intelligence research increasingly lost to time, and I feel fortunate to have been a receptacle for his pearls of wisdom. On top of this, he has continued to devote his full energy towards the mentorship of myself and other students, and his tireless enthusiasm to stay on top of new research in the field is a trait that I aspire to.

Prof. Ehsan Hoque has an extensive background in researching human communication and soft skills, and this shows in the many lessons that I've personally gleaned from him while being a part of the ROC HCI lab. Through collaborating with his lab, I've felt my own presentation skills and confidence as a researcher improve substantially. I greatly appreciate his encouragement over the years, as well as his drive to focus on truly impactful projects.

Thank you to Prof. Aaron Steven White for pushing me to explore new directions and acquire new skills over the course of my program. His impressive ability to perceive connections between diverse concepts in linguistics, statistics, and computer science

– combined with a tasteful aesthetic sensibility and general savviness – have made a strong imprint on my own research career.

Thank you as well to Prof. Scott Grimm for piquing my interest about lexical semantics, for the helpful feedback along the way, and for kindly stepping in as my committee chair.

I would also like to provide thanks to the many peers and colleagues with whom I have collaborated during my program. To Will Gantt, whose inquisitive and incisive mind has lead to many great conversations and collaborations over the years. To Kurtis Haut, whose gregarious personality made him a joy to work with, and a better leader for the SOPHIE project than one could ask for. To Masum Hasan, for his sharp insights and for encouraging me to think on the frontiers of emergent technology. To Georgiy Platonov, whose collaboration helped steer the initial direction of my doctoral research. To Zahra Razavi, and to Rafayet Ali, for being my earliest collaborators and the ones to first show me the ropes as an undergraduate researcher. To the other members of the Schubert NLP lab for creating a warm and erudite environment – Lane Lawley for his amazing sense of humor, Gene Kim for his approachability as the Lisp guru of the group, and Hannah An for her support up to the very end. And to any other collaborators that I did not name here, without whom this work would not have been possible.

I appreciate the efforts of the UR Department of Computer Science faculty and staff to foster a friendly and supportive community, and the many friends that I've made within the department who have made the process enjoyable.

Finally, I'd like to thank my family (and pets), who have supported me at every step of the way and who provided me with a place to decompress during times of difficulty.

Abstract

Generalist conversational agents are a longstanding goal for dialogue researchers; however, this goal remains elusive owing to the often divergent roles of robust "analogical" reasoning and rigorous formal reasoning in computer dialogue. This dissertation proposes a general dialogue management framework, Eta, for creating conversational agents using an explicit *schema* representation that subsumes both modes of behavior. The schemas used by Eta represent *expected* or prototypical dialogue events, and can be used to dynamically guide dialogue through incremental matching of schemas to Eta's observed dialogue context. Deploying an agent to a particular domain requires only the creation of a set of schemas and the integration of modular, portable pattern transduction methods; the latter allows for the flexible integration of both symbolic methods and large language models in processes such as interpretation, reasoning, planning, and generation.

I demonstrate the generality of this approach by presenting a chronology of case studies of conversational agents created using Eta across three highly diverse domains: beginning with a friendly peer for social skill assistance; then turning to a spatially situated collaborative agent in a physical "blocks world" domain; and finally I present a virtual standardized cancer patient for end-of-life communication practice, representing the most elaborate application of Eta to date. I conclude the dissertation by shifting to an empirical investigation of how prototypical knowledge about cognitive attitudes – such as that contained within dialogue schemas – is reflected in natural language itself, laying the groundwork for more precise methods of inference about such event knowledge.

Contributors and Funding Sources

This thesis was supervised by a dissertation committee consisting of Professors Lenhart Schubert, Ehsan Hoque, Aaron Steven White, and Scott Grimm. All material presented herein is the result of work done independently by the author, with the exceptions of the following collaborations.

In Chapter 5, Section 5.4 borrows from material originally published as Ali et al. (2020) and Razavi et al. (2019), which were collaborations with Mohammad Rafayet Ali, Zahra Razavi, Raina Langevin, Abdullah Al Mamun, Reza Rawassizadeh, Tianyi Ma, Kimberly van Orden, Lenhart Schubert, and Ehsan Hoque. Section 5.6 was originally published as Kane and Schubert (2023).

Chapter 6 borrows from material originally published as Platonov et al. (2020) and Kane et al. (2020), which were both collaborations between myself, Georgiy Platonov, and Lenhart Schubert.

In Chapter 7, Sections 7.4 and 7.5 borrow from material originally published as Haut et al. (2023) and Kane et al. (2023), respectively. These works were collaborations with Kurtis Haut, Catherine Giugno, Caleb Wohn, Varun Kumar, Tom Carroll, Ronald Epstein, Lenhart Schubert, and Ehsan Hoque.

Chapter 8 borrows from material originally published as Kane et al. (2021), as well as material currently in preparation. Both are collaborations with William Gantt and Aaron Steven White.

This work was supported by the NSF NRT Fellowship DGE-1449828, the DARPA CwC grant W911NF-15-1-0542, the NSF EAGER Award IIS-1940981, the NSF Award

IIS-1750380, the NSF Award BCS-1748969, and the University of Rochester Department of Computer Science. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of above named organizations.

1 Introduction

Natural language dialogue, in either spoken or written form, is the primary method by which humans transmit information about their underlying mental states in order to collaborate effectively on joint activities. It is therefore unsurprising that a sizable body of research on human and computer collaboration concerns the creation of *virtual conversational agents* – systems that can collaborate with a user principally through natural language interaction. Such systems can assist a user in completing everyday tasks; can tutor a user or answer questions about some topic; can automate costly communication training programs, among many more valuable applications.

Our ability to simulate truly human-like conversational agents is still nascent, requiring one to solve such difficult problems as theory of mind; massive-scale knowledge acquisition across various domains; design of advanced algorithms for perception and interpretation; transforming auditory or imagistic inputs to latent symbolic or sub-symbolic representations; and robust reasoning and planning methods utilizing those representations and knowledge. Indeed, it is for this reason that natural language dialogue is often considered an "AI-hard" task: at least as hard as the hardest problems in AI.

Nevertheless, some of these problems can, and should, be abstracted away in the pursuit of practical conversational agents. What remains is the problem of *dialogue management*: how an automated system should *mediate* between its interpretations of the utterances of interlocutors, its own private mental state (goals, obligations, memories, knowledge, etc.), the context of the dialogue (facts that are assumed to be "common ground" between participants), and exogenous events that it observes in the world. Along these lines, Traum and Larsson (2003) define a dialogue manager as consisting of the following (minimal) responsibilities:

1. Updating the dialogue context on the basis of the system's interpretations of communication.
2. Using the dialogue context (as well as knowledge or memories possessed by the system) to guide interpretation of communication or observed external events.
3. Interfacing with domain-specific reasoners or resources (e.g., a planner, ontology, or knowledge base) to coordinate both dialogue and non-dialogue behavior.
4. Deciding what action or utterance the system should perform next, and when it should perform it.

Numerous systems for dialogue management have been developed over several decades, and have varied widely in the generality of the virtual conversational agents that they instantiate. The great majority of conversational virtual agents to date have been *specialist* agents: Through the use of suitable task representations, plan recipes, and/or pattern-matching rules, they were able to achieve a high degree of competency in narrow domains. Seminal specialist agents included SHRDLU – a conversational agent allowing natural language interaction in a physical blocks world (Winograd, 1972) – ELIZA – a chatbot that played the role of a Rogerian psychotherapist (Weizenbaum, 1966) – and TRIPS – an assistant for solving logistics problems in an island simulation (Ferguson and Allen, 1998).

Specialist conversational agents are often sufficient for many practical applications, and thus remain an active area of research with numerous architectures and techniques having been proposed to enable task-specific dialogue (Zhang et al., 2020b; Jacqmin et al., 2022). However, in many more open-ended domains, where a conversation may veer unpredictably in various topical directions, it becomes more critical for an agent to have *generalist* as opposed to *specialist* capabilities. Recent advances in large language models (LLMs) and the expansion of available computational power have led to a renewed attention toward generalist agents. For instance, transformer models trained on a combination of natural language and diverse task representations have been able to demonstrate basic competency across a variety of natural language tasks (Reed et al., 2022; Bubeck et al., 2023). However, while these models appear to have impressively generalist *linguistic* capabilities, they fall short of achieving *functional* competency; particularly within tasks involving planning or reasoning (Mahowald et al., 2023; Valmeekam et al., 2023). Furthermore, autoregressive language models have been shown to exploit superficial linguistic cues while solving problems, suggesting that their performance in closed task evaluations is likely an overestimate of their performance in open-ended collaborative dialogues (Levy et al., 2023; Du et al., 2023).

The challenge in progressing from specialist agents to generalist agents can be better understood by considering a hypothetical dialogue agent as implicitly mapping from the space of possible "situations" to primitive actions (including dialogue actions, but also other cooperative actions in physically situated dialogues). As domains become more general, the complexity of the function implementing this mapping scales exponentially with the number of possible variables that describe a situation (Ginsberg, 1989) – precluding many of the search-based planning and reasoning methods that are successful in narrow domains. Therefore, in more general domains, this mapping needs to be decomposed into *expectations* about possible situations, allowing for tentative dialogue plans to be constructed through analogy between the observed situation and an expected situation. The apparent successes of LLMs as generalist agents can be attributed in part to their robust analogical capabilities, often compared to the "System 1" in the dual-system theory of cognition (Yao et al., 2023; Kambhampati et al., 2024); the apparent failures of LLMs can similarly be attributed to their lack of a "System 2" capable of more deliberate reasoning on the basis of explicit world models.

In this dissertation, I aim to demonstrate that the missing link between these two modes of behavior is precisely an expressive knowledge representation for expected patterns of dialogue situations, coupled with automated mechanisms for extrapolating dialogue actions from these expectations. Specifically, I propose the use of structured prototypical event representations henceforth referred to as *schemas* in the tradition of psychologists who, nearly a century ago, first theorized the role of such prototypical knowledge in human behavior (Piaget and Cook, 1952; Bartlett, 1932). Intuitively, these schemas (which may be learned or designed by domain experts) represent a hierarchical clustering of the full space of dialogue situations in a particular domain, allowing for analogy from the current situation to the closest matching prototype. This enables an agent to reflexively act by simply carrying out the expected actions in the matched schemas, as well as to engage in goal-driven planning based on the goals and conditions associated with the matched schemas – thus providing a useful generalization of plans (Turner, 1994) as well as other related concepts such as event or discourse scripts (Schank and Abelson, 1977; Poesio and Traum, 1997).

This dissertation introduces a *schema-based* dialogue framework – Eta – that can be deployed in a wide variety of tasks or domains through the creation of *dialogue schemas* and the integration of modular adapters/plugins for performing pattern transduction in the process of extrapolating appropriate actions from schemas (e.g., for mapping natural language inputs to an underlying semantic representation, or vice-versa).

I present a compilation of case studies in which Eta is used to develop conversational agents in three separate applied domains – each demanding a very different set of capabilities and providing unique insights towards the further development of Eta. The LISSA conversational agent – a virtual human for social skills practice – served as the genesis of the Eta dialogue manager, and subsequently served as a test of Eta's abilities to handle casual, open-domain conversation across a broad set of topics. The DAVID conversational agent was designed to support situated question-answering and collaborative tutoring dialogues in a physical "blocks world" domain, requiring the coordination of multimodal perception, deep semantic understanding, spatial and temporal reasoning, and planning. The SOPHIE conversational agent – a virtual standardized patient that allows doctors to practice end-of-life communication scenarios – is the most recent, and most elaborate, application of Eta. The challenges in this domain range from managing the mixed-initiative and goal-directed nature of conversation, to being able to understand the wide variety of responses that a user might provide, to detecting complex attributes such as empathy in the user's responses, to generating emotionally appropriate reactions. As these case studies illustrate, the Eta dialogue management framework is, to our knowledge, unique in its ability to balance fluency in miscellaneous domains with control of the structure and goals of dialogue.

A secondary topic that I explore in this dissertation concerns the semantic interface between natural language and particular forms of schematic knowledge associated with events – such as inferences pertaining to beliefs, desires, and intentions of agents. While in dialogue management we are typically concerned with complex events, even atomic events at the lexical scale appear to associate with prototypical semantic knowledge (Dowty, 1991; Heim, 1992; Anand and Hacquard, 2013, 2014). For example, given an event of a speaker *ordering* a hearer to perform some action, it is *typical* that the hearer consequently intends to perform that action, *modulo* context that may counteract this expectation (such as if the speaker has no relevant authority, or if the hearer is particularly disobedient). Perhaps surprisingly, the patterns of prototypical inferences associated with particular event predicates appear to be associated with constraints over the types of syntactic constructions that these predicates may appear in – suggesting that natural language itself is, in some sense, structured around this prototypical knowledge. I discuss the MegaIntensionality project – an effort to crowdsource lexical-scale annotations of prototypical belief, desire, and intention inferences associated with English verbs – and a subsequent analysis that aims to uncover the lexicosemantic components that give rise to syntactic variation in natural language. Not only is this analysis of theoretical import, but a model of such prototypical inferences might also be used to augment natural language inference (NLI) models, allowing for more precise dialogue management in frameworks such as Eta where tracking belief, desire, and intention is of utmost importance.

1.1 Chapter Descriptions

Chapter 2 begins by situating my work in the context of prior approaches to dialogue management, comparing and contrasting them with the approach presented in this dissertation.

Chapter 3 provides background on essential theoretical constructs upon which my work is premised – namely, episodic logic and dialogue schemas.

Chapter 4 provides a complete technical description of our dialogue management framework, Eta, and the dynamic schema-based planning algorithm that it uses to guide dialogues. Section 4.1 describes the components of the dialogue state used by Eta, while Section 4.2 describes the parallel processes of the Eta architecture that operate over this dialogue state. Section 4.3 describes the use of *transducers* – plugin-like functions that can transduce tree representations into new representations – within the Eta architecture, allowing for portability between domains, adaptation to new domains, and flexible integration of various NLP techniques.

Chapter 5 discusses the development of the LISSA conversational agent – a virtual human for social skills practice in casual open-domain conversation that served as the origin point of the Eta framework. The schema and transducer design of the system is discussed, and results from initial user evaluations are summarized. Section 5.6 presents an experiment that finds that, when integrated with LLMs, systems like LISSA can generate more engaging and personable responses using *habitual schemas* – a particular type of schema encoding knowledge about habitual agent activities.

Chapter 6 presents the DAVID conversational agent – a spatially-aware collaborative agent situated in a physical blocks world domain. I discuss the schema and transducer design for spatial question-answering sessions, as well as a novel method for registering historical states of the dialogue and reasoning about temporal queries (Section 6.4), and present results from experiments with both spatial and historical question-answering. I also discuss an extension of this agent to *concept tutoring* dialogues, where the agent attempts to teach the user some spatial concept through the collaborative construction of examples (Section 6.6).

Chapter 7 presents the SOPHIE conversational agent, which is the most elaborate and impactful application of Eta to date. SOPHIE is a standardized virtual patient capable of affective and goal-directed conversation in a medical domain, allowing medical practitioners to practice end-of-life communication scenarios and receive dynamic feedback. The schema and transducer design enabling this application are discussed. I present the results of a pilot experiment showing that SOPHIE can help users improve in some medical communication abilities (Section 7.4) and a follow-up evaluation of the conversation transcripts from this experiment (Section 7.5). Finally, I discuss recent improvements to the SOPHIE system through integration of the schema-based approach with an LLM (Section 7.6).

Chapter 8 takes a turn away from the Eta framework, and towards an exploration of how the expected cognitive attitudes implicit in dialogue management, and often explicit in dialogue schemas – such as beliefs, desires, and intentions – are related to the structure of natural language itself. We describe the collection of a lexicon-scale dataset of belief, desire, and intention inferences (Section 8.3), and then present a clustering model that we use to derive a taxonomy of English predicates based on several lexicon-scale inference datasets (Section 8.4). Finally, we conduct an exploratory analysis of the semantic components underlying patterns in these inference judgments, and their relation to syntactic features (Section 8.5).

2 Related Work

Several prior branches of research have proposed general task-agnostic frameworks for dialogue management. In this chapter, I discuss several of these approaches and discuss their strengths and limitations relative to our framework.

2.1 Recipe-based Dialogue

Early approaches to dialogue management emerged out of formalisms developed for discourse analysis. One such formalism – SharedPlans – was created to model discourses involving joint collaborative tasks between two or more agents (Grosz and Sidner, 1986, 1990; Pollack, 1990; Lochbaum, 1998). In the SharedPlans formalism, discourse participants collaboratively build and augment a joint plan by contributing information about their beliefs, wants, and individual plans through utterances. In doing so, each agent expands a tree of intentions using *recipes* – i.e., steps for achieving a particular intention – with each intention corresponding to a particular section of the discourse.

2.1.1 COLLAGEN

The SharedPlans formalism was directly implemented in the COLLAGEN dialogue manager (Rich et al., 2001). The overarching goal of the project was to create a general

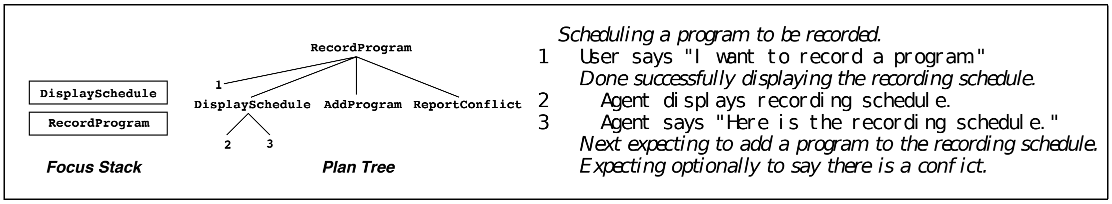

Figure 2.1: An example of a COLLAGEN dialogue state, showing the three components of the SharedPlans tripartite model (focus stack, plan tree, and discourse structure).

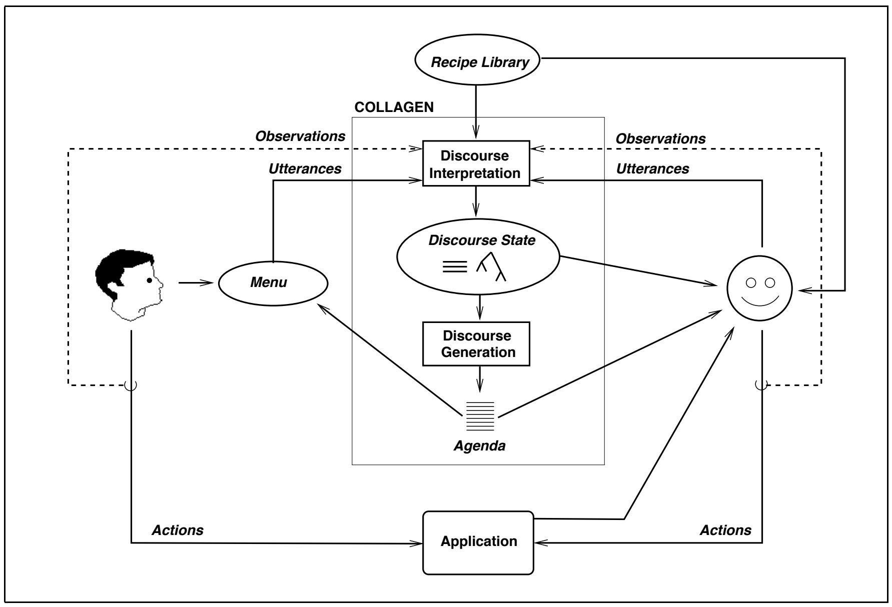

Figure 2.2: A diagram of the architecture of the COLLAGEN dialogue system.

dialogue manager, portable to many different applications, capable of answering open-ended questions about application domains – for instance, *"Why did you/we (not do) ...?"*, *"How do I/we/you do ...?"*, etc. One early application of COLLAGEN, for instance, was a virtual agent acting as an assistant to help a user set up and program a video cassette recorder (VCR). The user could ask questions such as *"What next?"*, and the system might respond with *"Now press Play on your camcorder..."*.

At the core of the SharedPlan formalism, and the COLLAGEN dialogue manager, is a tripartite model of discourse structure consisting of a hierarchical structure of utterances in the dialogue, a "dominance hierarchy" or plan tree of intentions corresponding to dialogue segments, and a focus stack representing the attentional state, with each focus space containing relevant properties, objects, relations, as well as the intention for that dialogue segment. These three components are illustrated in Figure 2.1. The components in this model directly correspond to relations among SharedPlans for dialogue acts – specifically, if some intention dominates another intention, then there's a subsidiary relationship between the SharedPlans corresponding to these dialogue segments. Similarly, partial and complete SharedPlans correspond to partial and incomplete dialogue segments. Plan augmentation, as well as manipulations of the focus stack, are therefore "reduced" to recognizing whether the user's utterance or proposition initiates, completes, or contributes to a dialogue segment.

The discourse state is updated after each utterance/primitive action by an agent (after being parsed into a description logic using standard-at-the-time NLU algorithms) according to an interpretation mechanism proposed by Lochbaum (1998), with extensions to allow for abductively inferring user plans from observed actions, and for handling plan interruptions (Lesh et al., 2001). Each dialogue event is interpreted as either starting a new dialogue segment whose task forms a subtask to the current task (pushing a new goal onto the stack), continuing the current task, or as completing the current task (popping the goal from the stack). Dialogue generation works in the opposite direction as interpretation – given the current dialogue state, it produces an "agenda" of utterances and actions by the agent which would contribute to the current task. The means by which a discourse event might contribute, continue, or complete a task are determined by a *recipe* corresponding to that task. Recipes are action descriptions containing parameters, constraints on parameter values, partially ordered steps, preconditions, and postconditions.

One further innovation of the COLLAGEN system was its plan recognition module used for discourse interpretation, which allowed the system to abductively infer the user's plan based on the user's observed actions (whereas previously the dialogue manager would require the user to explicitly state their intentions prior to acting, preventing natural conversation). This process can be made tractable by leveraging a couple hypothesized properties of collaborative dialogue: the focus of attention on a particular goal or task at any given time, limiting the search space of possible plans, as well as the fact that the plan recognizer only operates on plans that are minimally elaborated (i.e., expanded only as much as necessary to account for observed actions). Given the current plan and the task on top of the focus stack, the plan recognizer can try to extend the plan by applying recipes (which act as production rules) to the task in focus until a set of observed events is matched. If multiple plans are obtained, the system can insert a clarification request to narrow down the possible plan continuations. The overall system architecture combining these components is shown in Figure 2.2.

2.1.2 Notable Applications and Extensions of the COLLAGEN Approach

One notable extension of the COLLAGEN architecture is the health counselling dialogue system designed by Bickmore et al. (2011). This work addressed the problem of extending the generic COLLAGEN system with specialist domain knowledge – in this case, the relational facts and strategies used by health professionals to counsel patients on healthy exercise and eating habits. The system made two significant improvements to the COLLAGEN system. First, the hierarchical planner used by the system – *DTask* – allowed for the use of a declarative *domain ontology* in the planning process, in addition to the task hierarchy, providing the system with domain-specific knowledge and strategies. Second, the recipes used in the task model specified *adjacency pairs* (Clark and Brennan, 1991) (i.e., a question followed by a response), which allowed the dialogue manager to specify system responses conditionally on the context of the user's previous utterance.

Though developed independently to the COLLAGEN system, the RavenClaw dialogue manager (Bohus and Rudnicky, 2009) employs a very similar dialogue model based on hierarchical plan trees to represent the interaction, along with a discourse stack that is modified at runtime. Unique to the RavenClaw dialogue system, however, is an *expectation agenda* capturing what inputs the system expects from the user at any point, and allowing for complex error-handling. Many useful applications were developed using this framework due to the simplicity of engineering the task-specific dialogue trees and the system's support for mixed-initiative dialogue.

2.1.3 Discussion

COLLAGEN and related systems were in many respects groundbreaking, representing the first instance of formal theories of human dialogue (namely, the SharedPlans formalism) being used to create a generic computational dialogue manager. Furthermore, the plan inference system allowed plans to only be minimally elaborated insofar as observed actions are matched to plan actions, giving the system some degree of flexibility. Nonetheless, the system's ability to handle unexpected inputs was limited in practice because plan modification could only be done by chronological backtracking; agents created using plan recipes tended to be restricted to domains where a constrained task model could be devised. Moreover, the semantic interpretations used by the system were limited – user inputs were only interpreted insofar as was necessary to classify the user's utterance as a speech act and identify parameter values, which is insufficient for the general case.

The schemas used in the Eta dialogue framework are comparable to hierarchical recipes in that they include *procedural* knowledge about particular tasks or dialogue events; however, unlike recipes, dialogue planning need not strictly follow hierarchical expansion of schemas, since dialogue is driven primarily by matching the expectations in schemas to observations.

2.2 Plan-based Dialogue

The plan-based approach to dialogue management developed in parallel to the recipe-based framework, with the goal of developing conversational tools in domains that demanded more complicated problem-solving abilities. Whereas the recipe-based approach primarily composes actions based on hierarchical expansion of recipes, the plan-based approach involves systematic search over possible sequences of actions in order to achieve some goal state.

2.2.1 Speech Act Planning

The earliest systems for planning-based dialogue were primarily an application of the STRIPS planning formalism (Fikes and Nilsson, 1971) to classical theories of speech acts in dialogue. A STRIPS problem defines a set of operators, or actions, which operate on states of the world (sets of logical propositions) – these operators can specify preconditions that must to be true in order for the operator to be applied and effects that modify the state of the world after application of the operator. A STRIPS plan consists of a chain of operators that, when applied to some initial state, transform the state into a desired goal state. Typically, this basic formalism is extended with *hierarchical planning*, i.e., the ability to decompose operators into several sub-actions or sub-goals.

The classical theory of speech acts advanced by Searle (1969) posited several categories of necessary and sufficient conditions for the successful performance of any speech act. For instance, a successful act of *informing* a person about some fact requires that the speaker believes the fact, the speaker wants the hearer to believe the fact, and that the hearer consequently believes the fact. A seminal work in dialogue theory by Cohen and Perrault (1979) formalizes these conditions in the STRIPS formalism, allowing dialogue to be modelled as a sequence of speech acts that operate on individual cognitive states of dialogue participants, such as beliefs and intentions, represented using a first-order modal logic.

A tractable algorithm for dialogue management based on this plan representation was developed by Allen and Perrault (1980). For a collaborative goal-oriented dialogue to proceed, an agent will often have to infer the plans, beliefs, and goals of the other agent before continuing their own plan. This inference process is modelled as a search through a set of partial plans, each represented as tuples $\pi = \langle \Delta_{\texttt{alt}}, \Delta_{\texttt{exp}}, r \rangle$, where:

- $\Delta_{\texttt{alt}}$ is an *alternative* plan graph, created by chaining *plan inference rules* from an observed action by the other agent.
- $\Delta_{\texttt{exp}}$ is an *expected* plan graph, created by chaining *plan construction rules* from an expected goal of the other agent.
- $r \in \mathbb{R}$ is a rating for the partial plan (i.e., the current pair of plan graphs), initialized to 1 and updated using a set of heuristics.

Plan inference rules consist of 'if-then' rules that are used by an agent to form beliefs about the interlocutor's intents. For example, if `S BELIEVE A WANT P` and `P` is a precondition of some `ACT`, then `S` will by default infer that `S BELIEVE A WANT ACT`. Similar rules exist for reasoning about action bodies and effects. *Plan construction rules* are used by an agent to construct their own plan (as opposed to inferring beliefs of the interlocutor) and are essentially the reciprocals of plan inference rules. For example, if `S WANT ACT` and `P` is a precondition of `ACT`, then `S` will infer that `S WANT P`.

The underlying intuition behind the plan inference algorithm is that planning can be made tractable by simultaneously forward chaining from an observed action and backward chaining from an expected goal, finding pairs that 'meet in the middle' with unifiable actions or subgoals, and prioritizing the expansion of partial plans which are more likely to efficiently lead to a unification. At each step of the plan inference algorithm, the partial plan with the highest rating (according to multiple heuristics) is selected, and expanded according to the plan inference rules. The resulting plan graph is executed by the agent.

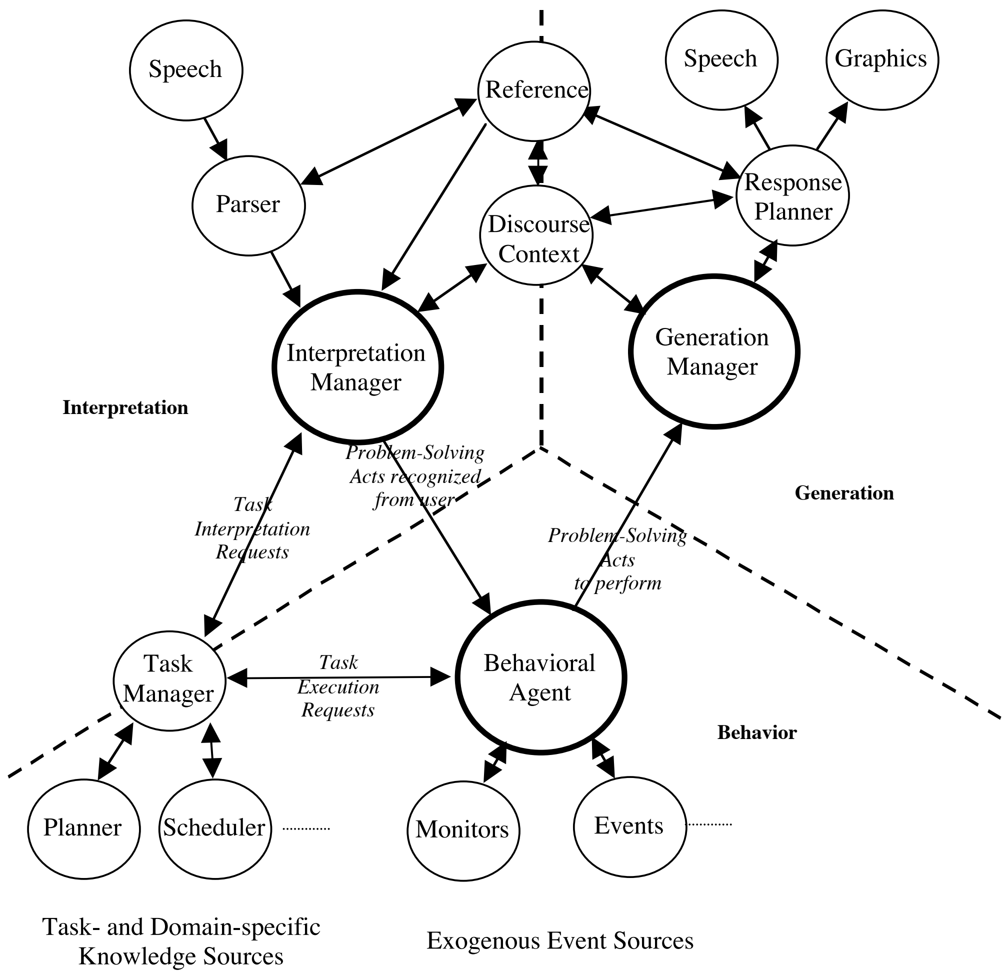

Figure 2.3: The overall architecture of the TRIPS dialogue manager.

2.2.2 The TRIPS system

The TRIPS system (Ferguson and Allen, 1998; Allen et al., 2001) is a modular dialogue management architecture built on the collaborative speech act planning framework, and was intended to integrate the abstract problem-solving capabilities of this framework with domain-specific behavior and task management – as demonstrated in a hypothetical emergency response task on a simulated island.

The architecture of TRIPS, as outlined by Allen et al. (2001), is shown in Figure 2.3 and briefly described in the following section. The various sub-components of the TRIPS system are based upon the Abstract Problem Solving Model, a shared planning model containing a set of problem-solving objects including *objectives* (i.e., goals), *solutions* (i.e., proposed sequences of actions intended to achieve an objective), *resources* (i.e., any domain objects used in a solution), and *situations* (i.e., the world settings in which solutions are being created). These objects are operated on using a set of collaborative problem-solving actions including both plan modifications (e.g., creating an objective, selecting or evaluating a solution, etc.) and speech acts (e.g., describing a plan).

The Discourse Context contains rich information about: (a) Salient discourse entities previously extracted by the interpretation module for reference resolution; (b) The structural representation of the immediately preceding utterance, to support ellipsis resolution or clarification; (c) The current turn-holder; (d) The discourse history consisting of speech-act interpretations of utterances so far, with indication of which have been grounded; (e) Current discourse obligations, typically used by the generation manager to form responses to the user's last utterance.

The Task Manager is responsible for acting as a medium between the domain-independent Abstract Problem Solving Model and recognition and execution of actions within a specific domain. The Task Manager's responsibilities include answering queries about objects and their roles in the task at hand; being an interface between the generic problem-solving actions used by the Behavioral Agent (e.g., "create a solution") and the actual task-specific performance of those actions; and assisting the interpretation manager in recognizing user intents when the user performs an action.

The Interpretation Manager uses an incremental chart parser to process user input in an online fashion. As it does so, it produces information used to update the turn-taking status of the dialogue context, as well as identifying intended speech acts by the user and the generic collaborative problem-solving action that the act furthers. Finally, any discourse obligations corresponding to the user's utterance are added to context. Since interpretations may be conditional on situational context, the identification of speech acts is governed by rules based on the Abstract Problem Solving Model and potentially queries made to the Task Manager. For example, interpretation of the utterance *"the bridge over the Genesee is blocked"* could be interpreted as the problem-solving act of identifying a problem with the goal of replanning, or as the introduction of a new goal to reopen the bridge. One of these two interpretations would be chosen by querying the task manager about whether there exists a plan using the bridge already, and whether making the bridge available is a reasonable high-level goal to adopt.

The Generation Manager is responsible for high-level response planning, while the Response Planner is responsible for lower-level response generation. The former system uses abstract problem-solving goals from the Behavioral Agent and discourse obligations from Context to produce plans for the system's responses. The latter system can produce surface-level responses using a number of different strategies: superficial template-based generation, a grammar, or output selection and coordination.

The Behavioral Agent is in some respects the "core" module of the system, as it governs the system's problem-solving behavior by balancing several aspects of the dialogue: the interpretation of the user's inputs, the system's own goals and obligations, and external events observed by the system. In each case, the Behavioral Agent is responsible for making choices about how much initiative the system should take in its response, according to its prioritized goals and obligations. For instance, if the user's utterance initiates creating a new objective, a low initiative response by the system might be adopting a new problem-solving obligation to find a solution, while a high initiative response might be computing a solution and proposing it to the user.

2.2.3 Collaborative Problem Solving (CPS) Dialogue Shell

Subsequent work on the TRIPS dialogue manager involved abstracting away the problem-solving model from the particular planning domain used by the original TRIPS system. The culmination of these efforts was the COGENT dialogue shell (Galescu et al., 2018), which is a minimal TRIPS-like system, in the sense that it excludes any TRIPS modules with domain-specific properties, such as the Behavioral Agent, Natural Language Generation, and the Domain Ontology. Included within the COGENT framework is the task-independent TRIPS parser (which relies on a general lexicon and ontology and produces logical forms), a Natural Language Understanding module which maps logical forms to *communicative acts*, and a Collaborative Problem Solving agent that maps communicative acts to *abstract communicative intentions* (ACIs). These intentions form the standard format for communication between the domain-independent COGENT system and any custom Behavioral Agent that a designer might create for a particular domain.

2.2.4 Discussion

The planning-based approach to dialogue, augmented with the TRIPS/COGENT architecture, is particularly useful in domains where complex problem-solving tasks are required. Many systems across a diverse set of applications have been designed on top of the COGENT dialogue shell. One particularly notable system built on top of COGENT is a multimodal collaborative agent in the "Blocks World" setting capable of interactively learning spatial concepts through being provided positive and negative examples by users (Perera et al., 2017, 2018).

Thorough planning and plan recognition, however, entail a search problem that can also become computationally intractable in complex scenarios. Moreover, many commonplace conversational interactions – such as asking an acquaintance how they're doing, or about their opinion on recent news reports, or getting to know someone –

are not based on systematic planning, but rather are reflexive, based on learned social obligations or norms related to such requests. Dialogue schemas in the Eta framework typically represent knowledge about *prototypical* dialogue events, enabling a more reflexive and computationally efficient method of dialogue management than plan-based architectures.

2.3 Information State-Based Dialogue

The information state update (ISU) dialogue management framework (Traum and Larsson, 2003) was created to provide a more flexible approach to dialogue management sufficient for simulating virtual humans. Both the recipe-based and plan-based approaches to dialogue require relatively strict adherence to complex action models, limiting their flexibility. Other factors may be more apt in explaining many ‘day-to-day’ interactions. Traum and Allen (1994), for instance, discuss the role of social discourse obligations: an agent asked *“Do you have the time?”* by a stranger likely doesn’t establish a shared plan or reason extensively about the stranger’s beliefs and goals, but rather simply reacts according to a learned social obligation or convention related to such requests.

Moreover, many aspects of ordinary conversation (including the collaborative kind) are difficult or impossible to analyze using these classical theories of dialogue planning. Utterances do not merely represent speech acts, but often comprise of multiple actions at the sub-utterance level related to turn-taking, repairs, backchannels (“yeah”, “I see”, etc.), etc. Furthermore, a classical speech act such as a request may not necessarily correspond to a single utterance, but may consist of a larger *discourse unit*, such as an “adjacency pair” consisting of a proposal and acknowledgement. Such discourse units are often necessary for the successful *grounding* of information, i.e., adding a piece of information, and the mutual understanding thereof, to the common ground (Clark and Brennan, 1991; Clark, 1996). In fact, the role of common ground in the previous theories has typically been left implicit and minimal – consisting of the observation of speech acts by both participants, and the reflexive inferences about beliefs and intentions which follow from that.

The ISU approach to dialogue posits a structured formal representation of the "conversational score" including both information that's been grounded by participants (i.e., added to common ground) and ungrounded "contributions" or discourse units (Poesio and Traum, 1998). A dialogue manager therefore consists of the following elements:

1. An **information state** consisting of informational components – participants, common ground/context, discourse structure, previous move(s), obligations and commitments, beliefs, intentions, user models and individual plans, etc.

2. A set of **dialogue moves** that update the information state, and rules for recognizing/realizing performance of these moves.

3. A set of **update rules** governing how the information state is updated, given the conditions of the current information state and any observed dialogue moves. Update rules have a set of preconditions specifying conditions on the values of particular components of the information state (possibly including variables to be unified), and a set of effects to be applied to particular components of the information state (possibly using any variables bound by the preconditions).

4. An **update strategy** for deciding which rule(s) to apply from a set of applicable ones. This can range from straightforward strategies such as choosing the first applicable rule, to more complex strategies such as choosing rules according to assigned probabilities.

A simple example of this approach is shown in Figure 2.4, with a set of update rules being shown on the left, and the analysis of a question-answer pair being shown on the right. This example consists of a very simple information state containing sets of private and shared beliefs, private agendas (stacks of intentions), shared knowledge of the previous move, and a stack of "questions under discussion" (QUD). In step (1), the only applicable rule is `selectAsk`, which selects an *ask* speech act as the expected next move given the agenda of the turn-holding agent (in this case, a computer system). As the agent makes the question utterance, the `integrateSysAsk` rule in step (2) pops the agent's agenda stack, and pushes the question that was asked onto the QUD stack. Under this context, any utterance the user makes is assumed to be an *answer* speech act, and if the utterance passes certain domain checks (such as relevance to the question), the propositional content of the answer is integrated into common ground in step (3) via `integrateUserAnswer`. Finally, since the propositional content in common ground resolves the question under discussion, the QUD stack is popped in step (4) using `downdateQUD`.

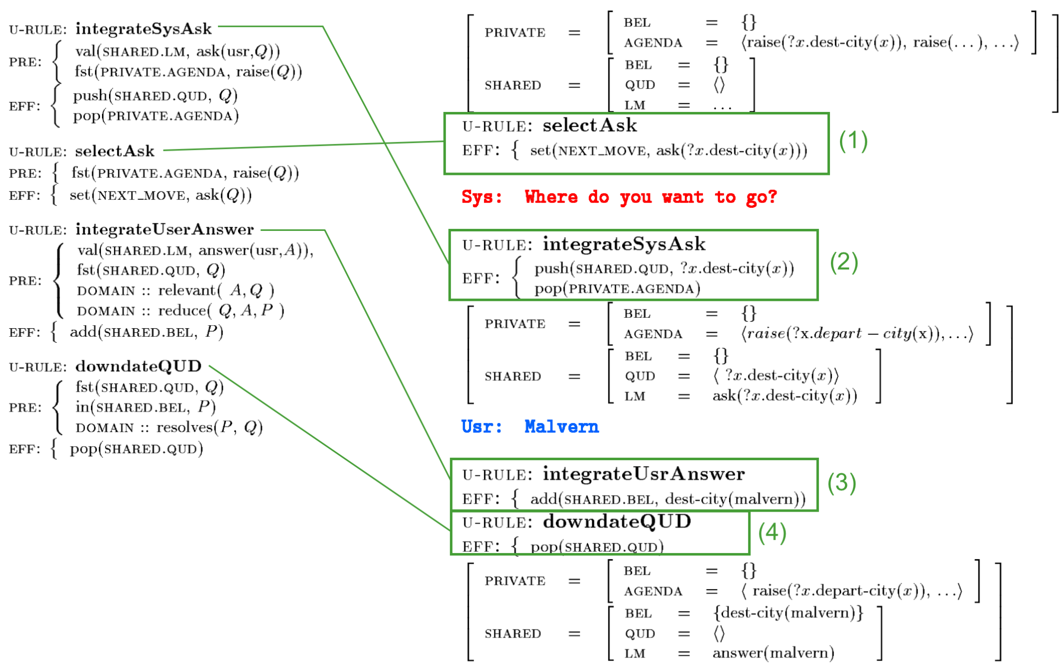

Figure 2.4: An information state is updated according to a set of update rules as a dialogue proceeds (Traum and Larsson, 2003). System utterances are shown in red, and user utterances in blue. Application of update rules are shown in green.

2.3.1 Notable Applications and Extensions of the ISU Approach

Several notable virtual human applications and dialogue management systems have made use of the ISU approach.

The 'How was your day?' (HWYD) system (Pulman et al., 2010) was designed to create a supportive and empathetic system capable of discussing open-ended work-related topics with a user. The system maintains an information state containing all of the dialogue acts and named entities detected by a natural language understanding pipeline, as well as conversation goals. If a new entity is recognized and introduced by the user input, the DM updates the information state with a goal to talk about it – the generation module uses a variety of strategies to generate empathetic responses, including sentiment analysis, templates, and a knowledge base. The underlying dialogue manager is notable for several novel contributions including the use of verbal and nonverbal backchannels by the system to keep the conversation engaging, and the ability to dynamically replan behavior if the user interrupts.

The "NPC Editor" software (Leuski and Traum, 2011) was designed to provide a convenient interface for dialogue designers to customize agent behavior using the ISU approach. It employs a statistical retrieval system to generate appropriate responses – at a high level, designers can specify semantic frames (i.e., a set of slot-value pairs), and a set of system utterances. Given the semantic frame representation of the user's input, the system selects a response utterance by computing language models for the frame and utterance – $P(F)$ and $P(W)$ respectively – scoring each utterance by Kullback–Leibler divergence between the two probability distributions ($D_{KL}(P(F) \ || \ P(W))$), and choosing the minimum scoring utterance. The KL-divergence is computed by treating the translation between semantic frame and the utterance as analogous to a standard cross-language information retrieval task. The NPC Editor framework can be combined with an information state to guide dialogue; for instance, the INOTS (Immersive Naval Officer Training System) virtual human – a virtual avatar for training army officers with interpersonal communication skills – was created using the NPC Editor software in conjunction with a branching dialogue state representation.

The FLoReS (Forward Looking, Reward Seeking) dialogue manager was created to overcome a limitation of prior information state systems – their dependence on predictable dialogue flows (Morbini et al., 2013). The FLoReS system addresses this issue (while still allowing for the ease of customization that characterized precursors) by combining several methods of dialogue reasoning as well as a reward-seeking algorithm for determining dialogue policy. Specifically, the information state is extended with local subdialogue networks (i.e., operators) for specific conversation topics. As events are received, the system chooses the operator that maximizes the expected reward given a particular information state, computed by propagating rewards over a graph of possible future dialogues. The resulting dialogue manager has been used successfully to create virtual agents in mixed-initiative tasks, such as holding healthcare counselling dialogues with military personnel (Rizzo et al., 2011).

2.3.2 Discussion

One strength of the ISU approach is the ease in which dialogue systems can be designed for various counselling and conversational practice domains, with agent behavior readily customizable by non-experts. Furthermore, aspects of this approach are amenable to various methods of integration with statistical NLP techniques, such as probabilistic topic retrieval or reward-seeking policy decisions for choosing dialogue policies, which can allow for more robust behavior than a rule-based system can. However, the natural language representations used by the systems are in general fairly limited. The NPC Editor program used to create the INOTS dialogue manager, for instance, uses only slot-value frames as semantic representations, which cannot capture many important elements of human language (e.g., quantification). Another potential drawback of such systems is that, as the number of information state update rules increases, the various interactions between these rules and their overall effects on the dialogue become more difficult to anticipate.

The ISU approach shares many similarities with the Eta framework presented in this dissertation; Eta defines several parallel processes that each update a shared dialogue state based on the output of transduction methods, much like the information state update rules in the ISU approach. However, Eta performs these updates on the basis of an explicit dialogue schema representation, in some sense synergizing the advantages of the ISU approach with the advantages of the plan-based approaches discussed previously.

2.4 Deep Learning-Based Dialogue

Since the advent of deep learning, many recent dialogue systems try to improve the system's ability to generate robust and realistic responses by leveraging statistical models trained on massive dialogue corpora. Although such systems fall mostly outside of the scope of this review due to their lack of dialogue management capabilities, I discuss some systems that were able to achieve robust behavior through either specialized training procedures or specialized architectures.

2.4.1 Chatbot Models

The Blenderbot chatbot model (Roller et al., 2020; Shuster et al., 2022) was able to achieve robust, topically broad conversational abilities through a combination of specialized Transformer architectures and particular conversation-oriented training datasets. Roller et al. (2020) compare three model architectures for the dialogue system: a **Generator** model that implements a standard Seq2Seq Transformer architecture; an attention-based **Retriever** model that uses the dialogue history as context to select the next dialogue utterance from a candidate set (typically, all possible training set responses)[1]; and a **Retrieve and Refine** model that combines the previous two models – a retrieval model is first used to select a candidate response, which is then paraphrased by the generator model.

Each model is pre-trained using 1.5B training examples derived from online message board discussions. The authors fine-tune with several datasets: the *ConvAI2* dataset where crowd-workers have casual "getting to know you" conversations; the *Empathetic Dialogues* dataset where one crowd-worker describes a personal situation and the other responds empathetically; the Wizard of Wikipedia dataset where crowd-workers hold conversations while utilizing wikipedia facts from a randomly assigned topic; and the *Blended Skill Talk*, where a "guided" participant is able to select utterances from chatbots trained on the previous three datasets. Using a crowdsourced transcript comparison protocol for evaluation, the authors find that using the Blended Skill Talk dataset, along with minimum length beam-search decoding with repetition penalties, produces the highest-scoring dialogues.

The Athena dialogue system (Harrison et al., 2020) was another effort to create a chatbot with topically-broad casual conversational abilities; rather than following an end-to-end approach, as in the case of Blenderbot, this system was based on a novel approach that "dispatches" response generation capabilities to multiple diverse sub-modules, and also grounds responses using a topic detector and entity recognition.

The natural language understanding pipeline of the Athena system strongly relies on Named Entity Recognition (NER) for dialogue management and response generation, which is implemented using a custom statistical NER model. Entities are additionally linked to external knowledge graphs derived from Wikipedia. The NLU component additionally parses the user input and classifies it as a dialogue act, from an ontology of dialogue acts. The dialogue manager uses a rule-based method (relying on inputs

[1] In a sense, this model is a "higher firepower" version of the statistical retrieval system used by the NPC Editor response planner (Leuski and Traum, 2011), with the trade-off of requiring vastly larger amounts of data to use effectively.

from the NLU component, as well as template-based keyword matching) to select one of several response generators to use, depending on the topic and nature of the user's speech act – these response generators (RGs) include a rule-based system for generating utterances related to the avatar's backstory, a news summarization RG, a neural RG trained on Wikipedia knowledge graphs, etc. The generated responses from one or several RGs are ranked, and used to construct the system's utterance.

2.4.2 Generative Agents

Some recent work has attempted to create generalist conversational agent frameworks using end-to-end deep learning approaches. Reed et al. (2022) train a multimodal transformer architecture on a diverse set of tasks, and observe that the resulting agent attains relatively high performance on a variety of tasks, while communicating through natural language.

Park et al. (2023) propose *generative agents* – i.e., agents that perceive, plan, reflect, and act using text streams and generative LLMs. Their overall agent architecture is shown in Figure 2.5, and consists of four primary processes that each involve querying an LLM with particular prompts. To enable this, they maintain a *memory stream* containing a comprehensive record of all observations, plans, and belief states, represented as natural language sentences. In order to make LLM prompting tractable, they retrieve a subset of memories by ranking them using a weighted average of their *recency* – their most recent access time – their *importance* – the impact of a memory as assessed by an LLM – and their *relevance* – their similarity to a particular query memory.

They perform a simulation of their framework in a life simulation video game setting with multiple agents. The memory stream is initially populated with observations of the world state. The process of *reflecting* involves recursively prompting an LLM to synthesize a tree of abstract knowledge, beginning with initial observations of the world. The process of *planning* involves prompting an LLM to create an agenda of actions,

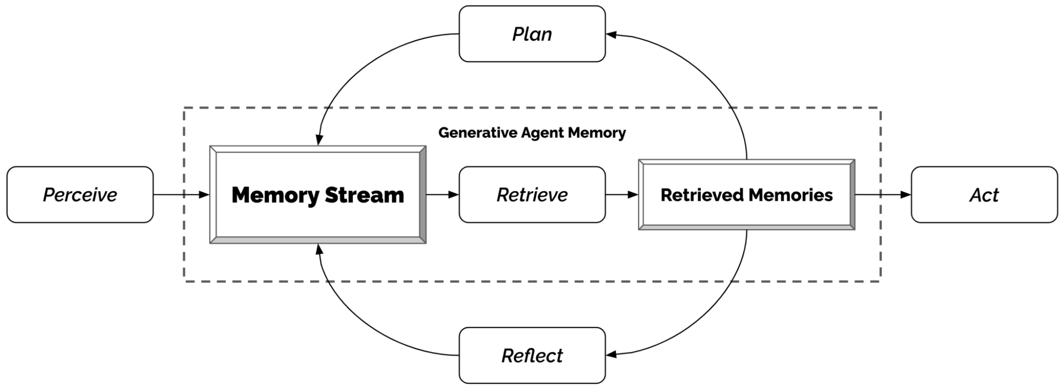

Figure 2.5: The generative agent architecture proposed by Park et al. (2023).

given a summary of the agent's previous day, at multiple levels of granularity – the LLM first provides an agenda for the full day, then recursively breaks down each activity into hour-long actions, etc. At each time step in the simulation, the agent *acts* by prompting the LLM to either act on the current plan action (for instance, generating a dialogue utterance), or reacting with a new, modified plan.

2.4.3 Discussion

Deep learning-based systems and generative agents excel in open-ended chatbot-like applications, owing to their powerful pattern-based analogical abilities, while also showing some promise as generalist frameworks. However, these approaches are often hallucination prone, computationally expensive, and often achieve worse performance in complex tasks where deeper reasoning or planning is necessary (such as the logistical planning handled by the TRIPS system); Roller et al. (2020), for instance, note a tendency of their system to contradict itself or forget a previously mentioned fact[2]. These limitations are notable because they precisely coincide with the strengths of the approaches discussed previously. For example, a plan-based dialogue system with some sort of explicit

[2]It is possible, however, that the latter limitations may be improved by recent advances in long context window models (see e.g. Gu and Dao, 2023).

knowledge store wouldn't contradict itself (provided sound inference rules), is capable of interpreting the semantic contents of the user's utterances to some extent, and can support mixed-initiative tasks with a sufficiently flexible system architecture. Ultimately, it would seem that the goal is to combine the advantages of these types of systems with the robust analogical abilities that deep learning systems exhibit.

The core processes and episodic memory used by Eta – discussed in Chapter 4 – resemble the components in the generative agent framework proposed by Park et al. (2023). Indeed, because of its flexibility in incorporating various pattern transduction methods, Eta is a "generative agent" in the limiting case, with purely LLM-based transduction methods – but in contrast, Eta supports the interpretation of inputs into a deeper logical form, as well as the integration of both symbolic methods and LLM-based methods for pattern transduction, while providing dialogue control via explicit schemas.

2.5 Schema-Guided Dialogue

There is some precedent for systems that use schemas to guide dialogue in collaborative tasks. While these systems were limited in scope relative to the framework presented in this dissertation, they sprung from the same motivation of enabling adaptive dialogue planning based on expectations, as opposed to strict planning or information state updates.

2.5.1 Schemas for Context-Mediated Behavior

An early attempt to address the challenge of handling unexpected events using a schema-based approach to dialogue was due to Turner (1994). The proposed model relies on the notion of *context-mediated behavior*, wherein possible contexts are themselves first-class objects represented as schematic knowledge. More precisely, the author introduces three subtypes of schemas:

- **Procedural schemas (p-schemas)** consist of an actor, a goal, and a set of steps. Each step may be a primitive action, a subgoal, or another p-schema. A p-schema may be specialized or generalized, is interruptable/resumable, and is only expanded as much as necessary.

- **Contextual schemas (c-schemas)** represent specific contexts or situations an agent may encounter, and encode domain-dependent aspects of that context (such as information about responding to events). A c-schema can contain knowledge about features of the context (used to determine when the schema is applicable), actions that are performed when a c-schema is selected, mappings from specific events to goals that should be adapted in response, and links between goals and p-schemas that achieve them in a particular context. This also includes *attention focusing information* that determines which goals an agent should attend to when in a particular context.

- **Strategic schemas (s-schemas)** represent information about an agent's domain-independent problem-solving strategies across contexts. An s-schema can set the agent's *goal importance* information, i.e. order different types of goals by precedence, as well as set an overall *event importance threshold* determining how sensitive the agent is to reacting to unexpected events in general.

Turner (1994) describes an algorithm for using these schemas to dynamically carry out an interaction. This process is diagrammed in Figure 2.6. Only one s-schema is assumed to be active at a time, but multiple c-schemas may be active simultaneously. These schemas are retrieved using the dynamic conceptual memory proposed by Schank and Burstein (1985).

Starting from step (1), the reasoner is initialized with a list of active goals. In step (2), the reasoner focuses attention on a particular goal using the s-schema's *goal importance* information, and the c-schema's *attention-focusing* information. In step (3), a p-schema is selected conditionally on the chosen goal and the *action selecting information* in the

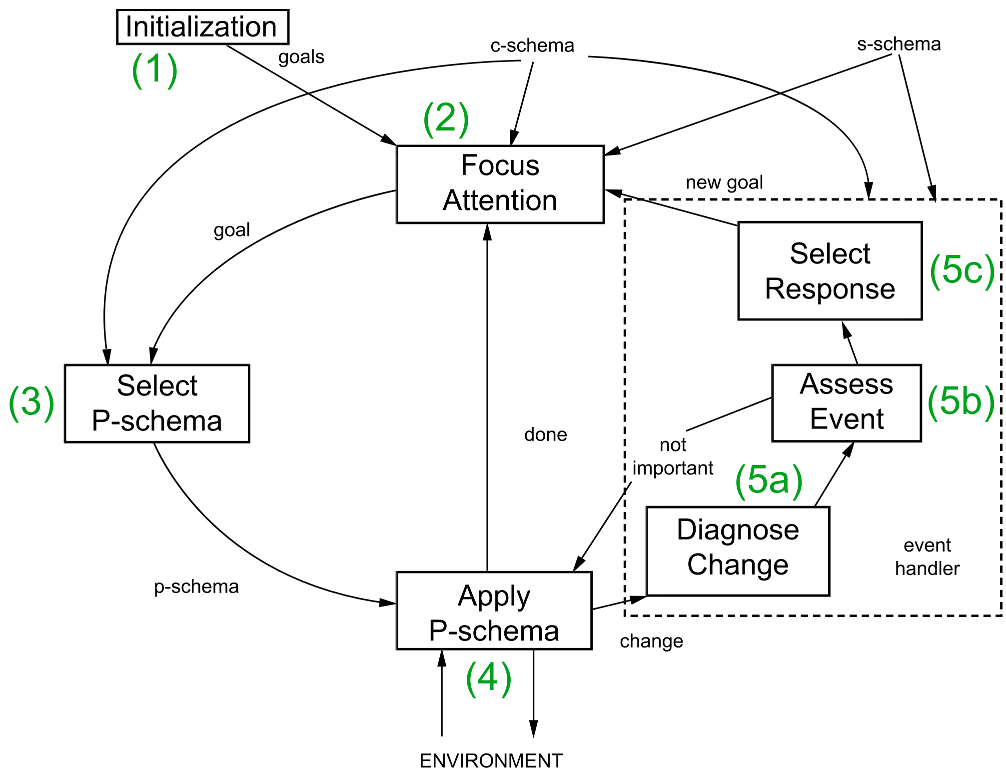

Figure 2.6: The schema-based reasoning process diagrammed by Turner (1994), with annotated steps.

c-schema mapping goals to actions within that context. In step (4), the p-schema is applied while simultaneously updating the environment and monitoring for external changes. If no external change is detected, the process returns to step (2) and repeats; otherwise the reasoner proceeds through the event handler component. The change is first diagnosed/classified in (5a), and assessed as important or unimportant using the c-schema and s-schema's *event importance thresholds* in (5b). If a change is deemed important, it is used to select a new goal according to the c-schema's *event information*, which is added to the list of active goals. The process then repeats at step (2).

2.5.2 Schemas for Task-Oriented Dialogue

More recently, the schema-guided approach to dialogue has been proposed as a method for guiding statistical dialogue models in task-oriented domains by determining alignment between observations and explicit dialogue schemas (Rastogi et al., 2020; Mosig et al., 2020; Mehri and Eskénazi, 2021). These schemas take the form of dialogue flow charts – DAGs encoding possible system responses and expected user responses – for specific practical tasks such as booking a hotel reservation. Where this approach differs from the recipe-based approach is in the fact that these schemas are interpreted as *expectations* about the flow of a dialogue; the actual execution of a dialogue, however, is to be determined using a statistical model that attempts to *align* the conversation with a schema.

To enable this, the authors use a *schema attention model* (SAM) that computes attention weights between nodes in the schema (specifically, their textual representations) and the dialogue history, with the objective of predicting the node in the schema that best corresponds to the current dialogue context. They train this model on a dataset of crowdsourced dialogue flow annotations and are able to achieve improved performance in narrow task-oriented domains, while also demonstrating zero-shot transfer to held-out domains.

2.5.3 Discussion

These schema-guided dialogue models demonstrate several features which will ultimately be important in systems such as our own – for example, an algorithm to instantiate plans from schemas whenever external observations match certain conditions within those schemas. However, the schema representations used by both frameworks are greatly limited.

The schemas of Turner (1994) rely on simple propositional slot values rather than semantically rich logical representations. Furthermore, it's questionable to what extent certain contextually-relevant actions by the user should be *separated* from the concept of a procedural schema (or more generally, an event schema). Ordinary procedures, such as the classic Schankian "restaraunt" script (Schank and Abelson, 1977), often do encompass conditional behavior associated with the procedure (e.g., whether or not to leave a tip), rather than any inherent aspect of the context.

The schema attention model, on the other hand, focuses primarily on task-oriented dialogue systems for tasks such as booking or scheduling events, and represents dialogue schemas as simple conversation trees. In contrast, the schemas used by Eta contain knowledge about not only expected dialogue flows, but also various types of expected conditions associated with dialogue events, and are used to dynamically plan a dialogue rather than to directly select the next action to take. This allows for a greater degree of topical flexibility and generality in the dialogues supported by Eta.

3 Background

The concept of a “schema” as an expected or prototypical event or object draws from a long interdisciplinary history[1]: the concept was first popularized in psychology as a construct for analyzing cultural variation in stereotypical perceptions (Bartlett and Kintsch, 1995), and was later adapted – under various labels – by cognitive science and artificial intelligence researchers as a proposed means of representing general knowledge. Examples include *frames*, introduced by Minsky (1974) to represent prototypical objects or concepts; *event scripts*, introduced by Schank and Abelson (1977) to represent prototypical patterns of events (such as the sequence of steps and conditions typically involved in going to a restaurant); *schematic structures* used by Van Dijk and Kintsch (1983) to analyze strategies used in human discourse comprehension; and *discourse scripts*, proposed by Poesio and Traum (1997) as a representation of stereotypical discourse knowledge (comparable to Schank’s event scripts) to be used in discourse analysis. We consider all of these concepts as subsumed by the concept of a schema, and henceforth use “object schema” to refer to prototypical object knowledge, “event schema” to refer to prototypical patterns of events, and “dialogue schema” to refer to

[1] In fact, the earliest known concept of schematic knowledge traces back to writings of the Chinese philosopher Mozi, who postulated ethics, reasoning, and argumentation to be guided by analogy to *fa*, i.e., prototypical models that aid in pattern recognition and decision making (Garrett, 1993).

prototypical patterns of discourses/dialogues[2].

While the formal representation of schemas varies across theories, they are generally characterized by several properties they share in common: first, the information they represent is prototypical and abstract. An actual instance of a type of event or object need not satisfy all properties in a schema to be "matched" to that schema. Part of the power of this representation is that a partial schema match of an observation can be used to abductively infer other properties of that observation – for instance, observing a person in a bank with a gun might trigger a "bank robbery" schema, causing the observer to infer other properties associated with the event, such as that the robber intends to steal cash. These inferences are fallible, however, allowing for deviations from the stereotyped pattern in specific cases – for instance, if the gun toter demands valuables from the bank vault rather than cash; and partial schema matches may be discarded altogether if other schemas become more appropriate – for instance, if the gun toter is revealed to be a security guard.

Second, schemas may be hierarchically organized at different levels of specificity. For example, an event schema for "going to a restaraunt" may have associated subschemas, such as "going to a buffet", "getting fast food", etc. These lower-level schemas inherit the expectations within the subsuming schema, though they may also specialize or modify particular properties (for instance, a fast food restaurant will typically not have waiters).

This property also implies that the processes of *generalization* and *specialization* are likely play an important role in how humans learn and modify schemas to begin with: As we encounter similar experiences, we often generalize from them; on the other hand, when we encounter situations that contradict our generalizations, we may refine

[2]We note that, insofar as dialogues are just a particular type of event, all dialogue schemas are in fact a special case of event schemas. However, since knowledge about expected dialogues may contain knowledge not relevant to external events – such as expectations about *discourse obligations* (Traum and Allen, 1994) – it is useful to distinguish them.

our expectations to be more specific to particular contexts (Van Dijk and Kintsch, 1983). Although the topic of schema learning lies outside the scope of this dissertation, we note that any generalist agent must ultimately be capable of updating and refining its world model based on experience, and that the hierarchical nature of schemas allows for this (Lawley et al., 2019).

3.1 Episodic Logic

Before discussing our formal schema representation, it is necessary to provide some preliminary background on the semantic representation that we use throughout our dialogue framework: Episodic Logic (EL). EL is a type-coherent, expressive intensional logical form designed to closely resemble the surface form of English. The full syntax and a proposed semantics of EL can be found in (Schubert and Hwang, 2000)[3]; in this section we provide only a high level overview.

EL represents formulas as S-expressions in which lexical symbols are annotated with suffixes resembling the part-of-speech (e.g., `.v`, `.n`, and `.pro` for verbs, nouns, and pronouns, respectively). Each suffix denotes a set of possible semantic types, e.g., `.v` denotes an n-ary predicate whereas `.pro` denotes an entity. A closed set of operators or special predicates are conventionally left without suffixes, such as temporal relations, quantifiers, and type-shifting operators. Atomic formulas follow an infix notation – for example, in the formula `(|Jo| give.v |Bo| |Book1|)`, the predicate *give* takes the arguments *Jo*, *Bo*, and *Book1*, where pipe notation is used to represent proper names of entities. A variety of type-shifting operators allow for phenomena such as predicate-to-modifier conversion, as in `((mod-n white.a) wine.n)`; predicate modification, as in `(k snow.n)` (i.e., the kind of stuff, snow); or sentence reification using the

[3]Subsequent publications reflect some minor syntactic changes and more significant semantic revisions (e.g., Schubert, 2000; Kim and Schubert, 2019).

operator *that*, as in `(|Jo| tell.v |Bo| (that (|Mo| leave.v)))`; for the moment neglecting the roles of events and their relations.

Three types of variables are allowed in EL expressions: a symbol prefixed with `?` indicates a variable of type *entity* – e.g., `(|Jo| give.v |Bo| ?x)` – while a symbol prefixed with `!` indicates a variable of type *sentence* (i.e., it is an alias for a sentence) – e.g., `(|Jo| tell.v |Bo| (that !s))`. A symbol prefixed with `^` indicates an *indexical* variable whose value is some entity in the current dialogue context, e.g., `(^me tell.v ^you (that !s))`.

Most importantly, events, situations, or states, broadly referred to as *episodes*, are *first-class* individuals in EL, and can be represented using explicit episodic variables or constants. These episodic terms are linked to formulas that describe them through the use of the *characterization* operator `**`. For instance, the formula `((|Jo| give.v |Bo| |Book1|) ** E1)` asserts that the episode `E1` is characterized by an event of Jo giving Book1 to Bo. This allows episodes to be explicitly related to other episodes in terms of part-whole, temporal, and causal relations.

As an intermediate level of interpretation between natural English and EL, we also employ an Unscoped Logical Form (ULF) representation (Kim and Schubert, 2019). ULF is an initial variant of EL that, while still being fully type-coherent, leaves certain phenomena such as quantifier scope, word sense, temporal relations, and anaphora unresolved. ULF also employs several *macros* in order to handle surface phenomena such as WH-movement. This provides the advantage of allowing for a simpler mapping between surface English and ULF, while still providing a sufficient representation that allows for certain types of natural language inference (Kim et al., 2019).

In certain dialogue applications – for instance, a shallow chatbot for entertainment purposes – it may not be necessary to use a representation beyond natural language at all. It is therefore useful to think of these multiple levels of representation as forming a hierarchy, with natural language (NL) at the most shallow level and EL at the deepest level. Such a hierarchy is shown in Figure 3.1, along with an example of each

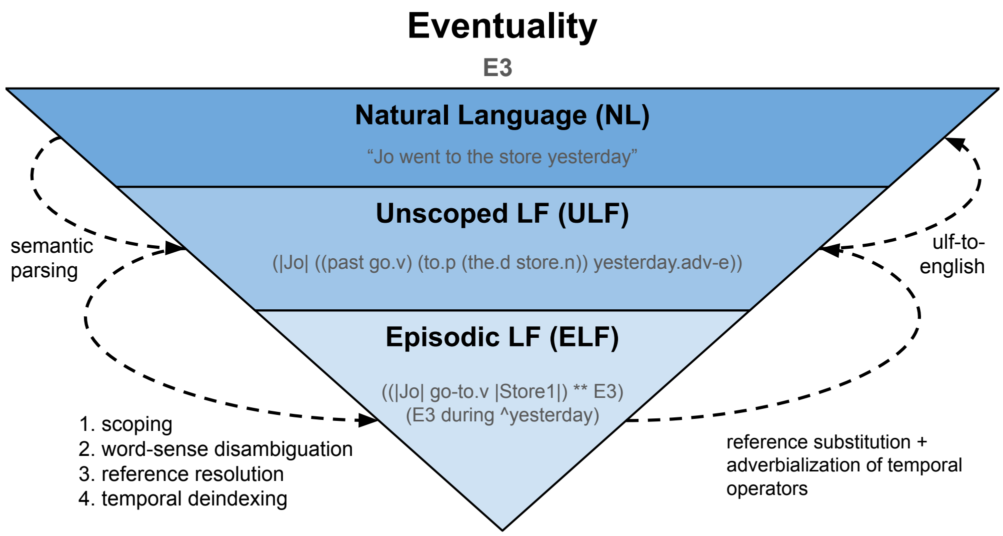

Figure 3.1: The hierarchy of semantic representations that may be used to describe a particular "eventuality", i.e., an event denoted by an episode variable or constant (in this case, `E3`. Examples are shown for each level of representation. The arrows between each level show the processes for converting from one representation to another.

representation. Converting from NL to ULF is a matter of semantic parsing (Kim et al., 2021), while converting from ULF to EL follows a multi-step pipeline that includes scoping, word-sense disambiguation, reference resolution, and temporal deindexing. In the other direction, converting from EL to ULF requires an inverse pipeline (e.g., reference substitution, insertion of tense and temporal adverbs), while converting from ULF to NL is a relatively straightforward process because of the close similarity between the two representations[4].

In our dialogue framework, we encapsulate all three levels of representation within the concept of an *eventuality* – a representation of a particular event characterized by the expressions contained within[5]. Minimally, an eventuality consists of an episodic variable

[4]The library that we use for mapping ULF to English can be found here: https://pypi.org/project/ulf2english/

[5]The term "eventuality" has previously been suggested by Hobbs (1985) as a way to conflate events and reified propositions, as is also implicit in other popular meaning representations such as Abstract Meaning

or constant denoting the event (such as “E3” in Figure 3.1) and a natural language string. If warranted by a particular dialogue application, an eventuality may furthermore contain ULF and EL interpretations of the NL expression.

3.2 Schema Representation

A schema is represented by a *schema header* – a logical formula characterizing the prototypical object or event of the schema – followed by a list of schema sections, each representing a particular kind of stereotypical knowledge associated with that object or event. Each schema section contains a list of eventualities, i.e., episode variables[6] paired with NL/ULF/EL expressions that characterize them[7]. The sections allowed within a dialogue schema are shown in Table 3.1[8].

We illustrate the schema representation with a hypothetical example for a “job interview” dialogue in Figure 3.2. The schema header specifies the arguments of the schema – an interviewer, a candidate, and a particular job. The schema contains sections such as PRECONDS and GOALS – containing ULF/EL eventualities in this case – for expectations associated with this event, such as that the interviewer wants to assess the candidate’s qualifications. The EPISODES section contains intended or expected steps

---

Representation, Minimal Recursion Semantics, etc. While EL maintains an ontological distinction between events and reified propositions, we borrow the term as a means of capturing the flexibility between shallow natural language representations and formal episodic representations.

[6]Technically, the episode variables within a schema are Skolem functions of the main episodic variable characterized by the schema header.

[7]One may make a distinction between schema sections that contain *fluent* and *nonfluent* formulas – for instance, the expected preconditions of an event versus the nominal types of participants in the event. Rather than denoting episodes, the variables associated with nonfluent predications – indicated using !-variables – denote sentence meanings, i.e., (as already noted) they act as aliases for those sentences, allowing us to form terms such as `(that !p)`.

[8]These sections are the same as those for event schemas in general, with the exception of the “obligations” section. Object schemas contain a different set of sections that we do not cover in this paper.

```
(dial-schema :header ((?interviewer conduct-interview.v ?candidate ?job) ** ?e)
;````````````````````````````````````````````````````````````````````````````````
:types (
  !t1 (?job job.n))

:preconds (
  ?p1 (?candidate want.v ?job)
  ?p2 (?job have.v (an.d opening.n)))

[...]

:goals (
  ?g1 (?interviewer want.v (to (assess.v
         ((?candidate 's) (n+preds qualifications.n (for.p ?job))))))
  ?g2 (?candidate want.v (to (display.v
         (k (n+preds qualifications.n (for.p ?job)))))))

:episodes (
  ?e1 (?interviewer say-to.v ?candidate '(Please tell me about yourself .))
  ?e2 (?candidate reply-to.v ?e1)
  ?e3 (:repeat-until (?e3 finished.a)
     ?e4 (?interviewer ask-interview-question.v ?candidate))
  [...]
))
```

Figure 3.2: A simplified example of a dialogue schema corresponding to an expected job interview dialogue, with the eventualities in each schema section represented as ULF/EL formulas. The schema illustrates an example of a repeating episode, where `ask-interview-question.v` may itself correspond to a separate dialogue schema.

by the user in the dialogue, including an example of a questioning episode that repeats until the interview is complete. Steps within schemas – such as `?e4` – may themselves correspond to other dialogue schemas[9].

A schema may be *matched* if there is a sufficiently strong similarity between the eventualities contained within a schema and the eventualities observed within a dialogue. When a schema is successfully matched, its contents are *instantiated*, allowing additional knowledge to be inferred from the eventualities contained within the schema. Our proposed dialogue management framework drives dialogue through incremental instantiation of the episodes within a selected schema.

[9]It is also possible to represent the episodes in the primary schema at a finer level of granularity; for the purpose of illustration, we provide only a coarse representation.

Section	Explanation
EPISODES	Expected sub-events that occur within a dialogue – most commonly speech acts by participants, but may be any anticipated event or more complex procedural behavior such as repeating an episode until a contextual condition is met.
EPISODE-RELATIONS	Temporal relations between episode variables specified in the episodes section. The default ordering between episodes is sequential in the order they occur in the schema, but other constraints can be specified (such as “consec” for two directly consecutive episodes, or “same-time” for simultaneous episodes).
TYPES	Non-fluent type predications for individuals occurring in the schema, e.g., that the speaker participant is type `robot.n`, the hearer is type `person.n`, or that some variable `?x` has type `block.n`.
RIGID-CONDS	Any non-fluent predications about individuals occurring in the schema apart from the types, e.g., that `?x` is yellow.
STATIC-CONDS	Fluent predications which are expected to hold throughout the schema episode, e.g., that the hearer is located near the speaker.
PRECONDS	Fluent predications that are expected/required to be true at the initiation of the episode represented by the schema.
POSTCONDS	Fluent predications that are expected/required to be true at the conclusion of the episode represented by the schema.
GOALS	Formulas corresponding to the goals of participants in the schema.
OBLIGATIONS	Discourse obligations associated with particular episodes in the schema, e.g., a request speech act may oblige the hearer to accept or reject the request.

NECESSITIES	Predications associating schema conditions with values in [0,1] indicating how necessary it is that those conditions hold. For instance, if some precondition is necessary to degree 1, then it is strictly expected to hold in order for the schema to match an observed event. If a precondition is necessary to degree 0.5 and doesn't hold, the schema might be dispreferred but could still be matched if other conditions are sufficiently compelling.
CERTAINTIES	Predications associating schema episodes with values in [0,1] indicating how certain it is that those episodes will be observed. If an episode is certain to degree 1, then that episode must be matched to an observation for the schema to proceed. If an episode is certain to degree 0.5, the agent might proceed with a schema even if that episode is not matched to an observation.

Table 3.1: Dialogue schema sections and explanations.

4 The Eta Dialogue Management Framework

In this chapter, we provide a technical description of our schema-based dialogue management framework, Eta[1]. At a high level, Eta consists of four parallel processes: perception, reasoning, planning, and execution. Each process operates on a shared dialogue state that includes a schema library, episodic memory, dialogue context, and a dynamic dialogue plan. Within each process are several subtasks that involve transduction over structured data – for instance, natural language input to an eventuality, an eventuality to a (sub)plan, etc. These mappings are enabled using modular *transducers* – developer-defined functions with pre-specified input and output types – allowing for a flexible combination of techniques to be used in their concrete implementation. In order to create a conversational agent using Eta, a designer need only create a set of dialogue schemas and transducers; as we show in the subsequent case studies, many aspects of these are portable across domains.

4.1 Dialogue State

We begin by describing the components of the dialogue state used by Eta. The dialogue state contains records storing information for the overall session – such as names of the

[1]"Eta" echoes the 3 most frequent letters in English text (cf. Winograd's SHRDLU), and the phonetics of "Ada".

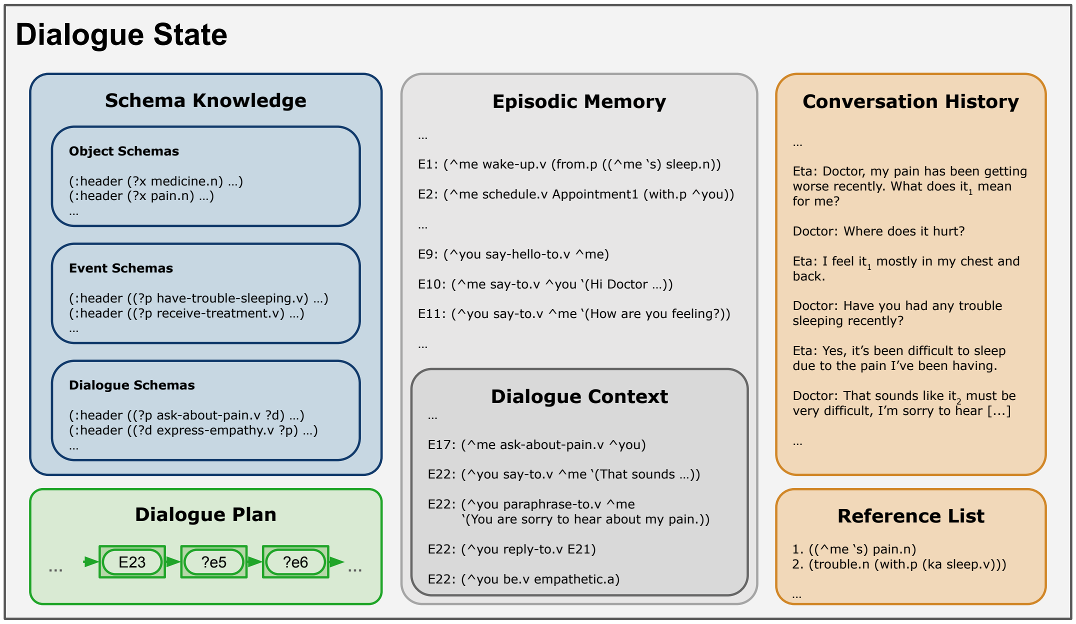

Figure 4.1: The main components of Eta's dialogue state, shown with a simplified example from a medical domain.

agents involved, buffered outputs, etc. – as well as dialogue state. We focus on three primary components of the dialogue state: the schema library and record of instantiated schemas, the agent's episodic memory and dialogue context, and the current dialogue plan. In addition to these, Eta also maintains a conversation log (a record of utterances by each agent) and a reference list containing prior entity mentions for reference resolution.

A snapshot of a dialogue state is shown in Figure 4.1, using a simplified interaction between a doctor and a simulated patient (similar to the actual case study discussed in Chapter 7) as an illustrative example.

4.1.1 Schema Library

An Eta agent is initially endowed with a *schema library* containing abstract object, event, and dialogue schemas. In the example shown in Figure 4.1, the schema library contains knowledge about objects such as medicine, events such as receiving treatment, and prototypical dialogues such as a patient asking a doctor about their pain. The behavior of the dialogue manager is driven principally by the selection and incremental instantiation of dialogue schemas from this schema library, beginning with a top-level schema for the overall interaction.

Instantiating a schema creates a copy of that schema, and if any variables within the schema were unified with constants upon matching the schema to observations, those variables are *bound* throughout the schema. For instance, if a schema with header `((?s say-hello-to.v ?h) ** ?e)` is matched to the observation `(^me say-hello-to.v ^you)`, the variable `?s` is bound to the indexical variable `^me` (i.e., Eta), and likewise `?h` is bound to `^you` (i.e., the user) throughout each schema section. Each instantiated schema is maintained in a separate database; note that the same abstract schema may result in multiple distinct instantiations as a dialogue progresses.

Schema Retrieval

Eta, in its current state, allows two methods for retrieving relevant schemas from its schema library. First, it allows for efficient retrieval of particular schemas (or relevant knowledge from schemas) through an associatively indexed array of schema headers – allowing, for instance, the schema corresponding to an observed or intended event to be retrieved and instantiated.

Second, Eta allows for retrieving schemas (or knowledge from schemas) through computing statistical similarity with some query string[2]. Specifically, Eta implements a multi-level retrieval system that uses an embedding function (e.g., a pre-trained transformer model) to pre-compute embeddings for each schema, each fact within each schema, and the given query string q:

[2] The query string may be, for example, the previous conversation turn, or a textual representation of the full dialogue context.

$$
\begin{aligned}
\boldsymbol{e}_{S_i} &= T(S_i) && \forall S_i \in \mathcal{S} \\
\boldsymbol{e}_{S_i,f_j} &= T(f_j) && \forall f_j \in S_i, \forall S_i \in \mathcal{S} \\
\boldsymbol{e}_q &= T(q)
\end{aligned}
$$

Where T is the embedding function, $\mathcal{S}$ is the full set of schemas, $f_j \in S_i$ is the full set of facts contained within each section of schema S_i.

We then retrieve the schema most relevant to q using a cosine similarity measure. If particular knowledge within the schema is desired, we additionally score the facts within that schema based on computed similarity, returning the top n facts.

$$
\begin{aligned}
\text{score}(f_j) &= \text{sim}(\boldsymbol{e}_{S^R,f_j}, \boldsymbol{e}_q) && \forall f_j \in S^R \\
S^R &= \underset{S_i \in \mathcal{S}}{\text{argmax}}\, \text{sim}(\boldsymbol{e}_{S_i}, \boldsymbol{e}_q) \\
\text{sim}(\boldsymbol{e}_1, \boldsymbol{e}_2) &= \frac{\boldsymbol{e}_1 \cdot \boldsymbol{e}_2}{\|\boldsymbol{e}_1\| \|\boldsymbol{e}_2\|}
\end{aligned}
$$

4.1.2 Episodic Memory and Dialogue Context

While schemas encode *general* knowledge about prototypical concepts (i.e., semantic memory), Eta also retains knowledge about specific events that occur prior to or over the course of a dialogue. Eta stores this knowledge in an *episodic memory* – a set of memories in which each memory (a particular event represented by an eventuality) is associated with a start time, an end time, the time of the most recent "access" of that memory, and potentially an importance score assigned to the memory upon creation. In the case where a memory encapsulates a current event, the end time of that memory is not defined.

A subset of Eta's episodic memory – those memories that are currently true, and assumed to be part of the *common ground* (Clark, 2006) for both agents – are of particular importance for the schema instantiation process. We maintain these memories in a separate *dialogue context*, where it is assumed that all memories in the dialogue context correspond to current events or facts that are currently true. Memories are removed from context when the events are observed or inferred to no longer be true. Additionally, some telic verbs are monitored and flushed from context periodically – e.g., "say-to.v" events within a dialogue are essentially instantaneous and therefore are removed from context as soon as Eta is able to plan its next dialogue turn (but are retained in memory).

In the example shown in Figure 4.1, Eta's episodic memory includes events experienced by the agent prior to the start of the dialogue (e.g., waking up from sleep) and events that occur during the dialogue (e.g., past speech acts). The dialogue context includes current speech acts as well as currently true facts inferred from those speech acts (e.g., that the doctor is being empathetic).

Memory Retrieval

To enable efficient lookup of particular memories, memories are stored in associative arrays indexed on both episode constants and predicate-argument combinations of the encapsulated eventualities.

Furthermore, we allow for statistical retrieval of memories using a scoring function $\mathcal{S}\colon \mathcal{M} \to \mathbb{R}$. Inspired by the retrieval mechanism proposed by Park et al. (2023), we choose this scoring function to be $\mathcal{S}(m) = \alpha_1 R(m) + \alpha_2 I(m) + \alpha_3 S(m, q)$, where for each memory $m \in \mathcal{M}$: $R(m)$ is the *recency* of that memory, i.e., the most recent access time; $I(m)$ is the *importance* score assigned to that memory; and $S(m, q)$ is the *salience* of that memory to a query string q, if provided, using embedding cosine similarity[3].

[3] If a query string is not provided, or an embedding function is not defined for an Eta agent, this term is 0.

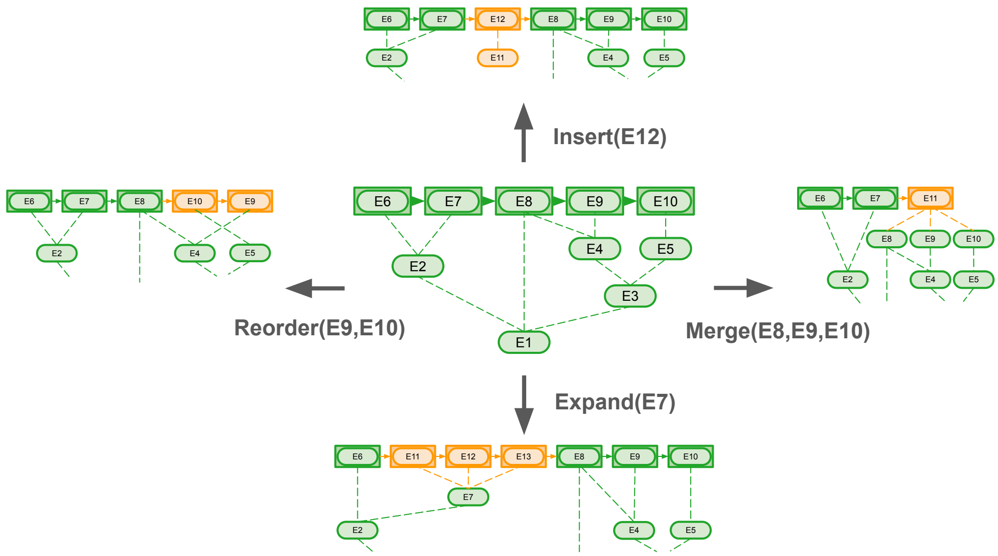

Figure 4.2: An example of an Eta dialogue plan and underlying graph of plan steps (center), and examples of each of the four types of possible modifications to the dialogue plan. Boxed nodes indicate the plan nodes in Eta's current dialogue plan, which form the frontier of the graph of plan steps. The nodes affected by each modification are highlighted (note that parts of the original graph are omitted for space). Expansion is shown as "upward" to the more specific level, i.e., the frontier of the graph.

Each subscore is first normalized across all candidate memories to a value in $[0, 1]$. α_1, α_2, and α_3 are tuneable hyperparameters; by default $\alpha_1 = \alpha_2 = \alpha_3 = 1$. After ranking all candidate memories using the scoring function, the top n may be retrieved.

4.1.3 Dialogue Plan

The overall course of an Eta dialogue is determined by dynamic and incremental operations over Eta's *dialogue plan*. A dialogue plan is a sequential list of plan nodes, each of which is either an intended action by the Eta agent or an expectation. This is a plan in the sense that the agent attempts to schedule actions that further the interaction according to its expectations about a particular dialogue. Unlike classical approaches to dialogue planning, however, these actions are not necessarily chosen using systematic search

Modification	Explanation	Example
INSERT	Add a new plan node to the dialogue plan between two existing plan nodes, additionally inserting a subgraph with a new source vertex into the DAG of plan steps.	In a collaborative dialogue with a robot in a "blocks world" domain, the user accidentally knocks a block off the table. Matching a schema for an "knocking over block" event, the robot reacts by inserting an action to pick up the block.
MERGE	Create a new plan node that combines several existing plan nodes, such that the new plan node becomes a substep of each merged step. The new plan node replaces the merged steps in the dialogue plan.	An agent may have an intention to greet a user in a schema for casual conversation. However, if the user greets the agent first, the agent may react by inserting a reaction into the plan. Since reacting to the user's greeting satisfies the original intention to greet the user, these nodes may be merged in the plan.
EXPAND	Decompose a plan node into several substeps; the substeps replace the original node in the dialogue plan, and the original node becomes the superstep of the new plan nodes in the DAG.	A train-booking agent may have an intention to help a user book a train. By matching a schema for booking a train, this plan step may be decomposed into several intentions and expectations, such as the expectation that the user requests a ticket, the intention to show the user ticket options, etc., which replace the original intention in the dialogue plan.

REORDER	Swap the position of two nodes in the dialogue plan according to an agent's temporal constraints or priorities.	A healthcare education agent may discuss topics with a user. The agent may initially intend to discuss the topics in a specific order; however, if the user informs the agent that they wish to discuss a particular topic first, the agent may reorder the intended actions in its dialogue plan to satisfy the new temporal constraint.

Table 4.1: Descriptions of each type of plan modification, and examples of where they might occur in hypothetical agent interactions.

towards a particular goal, but instead are driven primarily by incrementally matching expectations and intentions in schemas in a manner that is subject to modification based on new observations.

In the example shown in Figure 4.1, Eta had just executed the step `E23`, instantiating that event throughout the plan and corresponding schema instances (i.e., binding the episode variable to a new episode constant). The step `?e5` is now pending.

Since steps in Eta's dialogue plan are often substeps of a more abstract schema (e.g., a primitive `say-to.v` action may be a step within a more abstract `say-hello.v` schema), steps must also be related hierarchically in order to allow for behavior such as modifying the plan to account for failed actions. Formally, the plan nodes in Eta's dialogue plan form the "frontier" of a directed acyclic graph (DAG) of *plan steps*, where an edge from plan step $s1$ to plan step $s2$ indicates that $s2$ is a substep of $s1$ (we also maintain backpointers from substeps to supersteps). Therefore, all sink vertices in the graph are in the current dialogue plan, and source vertices correspond to top-level schemas. Each plan step in the DAG contains a particular eventuality, any discourse obligations associated with that step, and a list of pointers to dialogue schema instances containing that step as an expectation, allowing schema instances to be updated as a plan is modified (for instance, variables within the schema and plan may be bound to values in the course of executing an action or matching an observation). An example plan is shown in the center of Figure 4.2; boxed nodes indicate the plan nodes in Eta's dialogue plan, which are the frontier of the overall DAG of plan steps.

We allow for four types of modifications to Eta's dialogue plan, illustrated in Figure 4.2. We provide descriptions for each type of modification in Table 4.1, along with examples of these modifications in hypothetical interactions.

4.2 Dialogue Manager Architecture

One challenge particular to generalist agent architectures is that goals pertaining to multiple different tasks, or observations from multiple sources, may potentially be interleaved in the same interaction. Consequently, pipelined architectures in which natural language understanding, planning, and generation occur sequentially will struggle in more complex domains where interleaving is appropriate (Yu et al., 2017). We instead opt to use a multiprocess architecture consisting of four parallel processes – perception, reasoning, planning, and execution – allowing for these processes to be interleaved in any order. Each process consists of several subtasks that use transducers to modify aspects of the dialogue state. These processes are depicted in Figure 4.3, and the subtasks within each are elaborated further in the following sections.

Two additional difficulties emerge from the use of multiprocessing. First, since each process operates on a shared dialogue state, there is a risk of two processes attempting to modify the dialogue state simultaneously and creating a race condition. To solve this, we use a mutual exclusion lock to ensure that only one process can modify the dialogue state at any time.

Second, since a process may consume external resources (for instance, computing

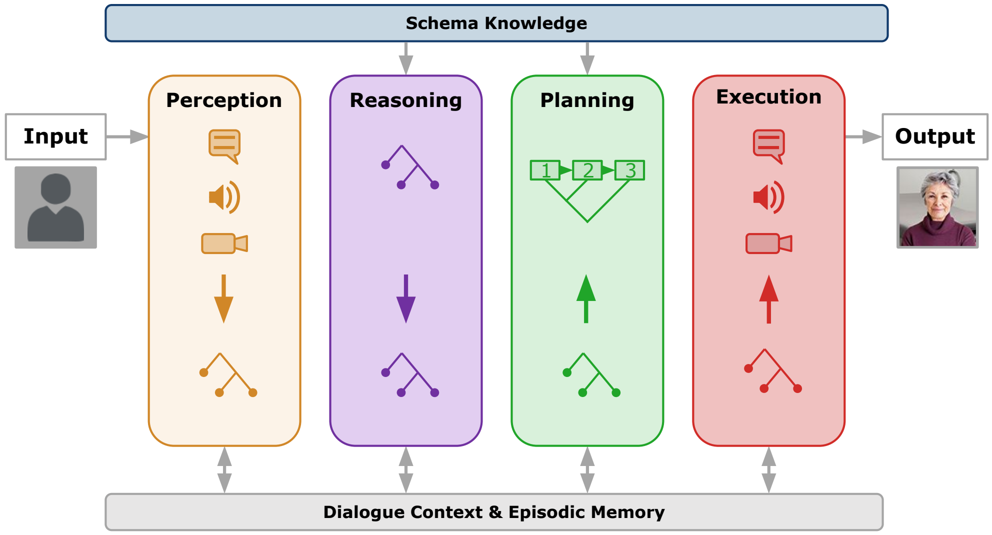

Figure 4.3: The Eta architecture consists of four parallel processes that operate over a shared dialogue state (schema library, dialogue context, episodic memory). The perception process collects observations from multimodal perception servers interprets them in context, resulting in new eventualities (indicated by a tree). The reasoning process attempts to infer new eventualities from existing eventualities. The planning process uses the dialogue context and schema library to modify the dialogue plan. The execution process attempts to advance the dialogue plan by executing actions or matching expectations, possibly resulting in multimodal outputs.

power for model inference, or monetary cost for external APIs), it is important that a process only executes if some aspect of the dialogue state has changed to warrant the execution of that process (with the exception of the perception process, since the external world may change at any point). Therefore, we use a set of *data buffers* – i.e., priority queues – to store recently updated observations, inferred knowledge, plans, and potential actions. Each process will pop items from the relevant data buffers, and will be skipped in the case where those data buffers are empty.

4.2.1 Perception

The perception process allows Eta to collect observations from the external world, including both speech acts from the user and other types of observations depending on the particular agent configuration (such as physical actions by the user, visual inputs, etc.). Upon each execution of this process, Eta collects observations from *perceptual servers*, i.e., external servers that collect input across modalities such as speech, text, and video and communicate with Eta through either natural language strings or ULF/EL formulas.

If an observation is not a speech act, it will be directly converted to an eventuality. If an observation is a speech act, Eta will attempt to interpret the observed utterance in the context of the current conversation using a multi-stage NLU pipeline.

First, a *gist transducer* will attempt to map the utterance, given the context of the previous turns in the conversation history, to one or more explicit, minimal, context-independent "gist clauses" that capture the contextual meaning of the original utterance – a process known as decontextualization (Choi et al., 2021). For example, if an agent asks the user *"What do you do for fun?"*, and the user replies *"well, maybe playing volleyball with friends, and then there's Dungeons and Dragons"*, then using the question it asked (itself possibly in some regularized form) as context, the agent might obtain gist clauses like *"I enjoy playing volleyball with friends"* and *"I enjoy playing Dungeons and Dragons."*. Since these are no longer dependent for their meaning on the question that elicited them, they are portable across multiple contexts and may be directly useful (e.g., for generating responses) even without further levels of interpretation.

In subsequent interpretation steps, a *semantic transducer* will attempt to map the extracted gist clauses to their underlying semantic representations (e.g., ULF/EL interpretations of the gist clauses in the prior examples). Likewise, a *pragmatic transducer* will attempt to extract the pragmatic meaning of an extracted gist clause (i.e., any meaning that the gist clause may carry apart from its literal semantic meaning). A reference list containing previously mentioned entities may also be used for coreference resolution, as illustrated by subscripts in the conversation history in Figure 4.1. All eventualities created in this process are added to the dialogue context, as well as to the buffer of new observations.

4.2.2 Reasoning

The reasoning process allows Eta to infer new facts and possible courses of action from observations and existing knowledge; it executes whenever new facts are added to the inferred knowledge data buffer. Since facts inferred during one iteration may be grounds for further inferences, Eta tracks the *inference depth* of each eventuality in the inferred knowledge buffer. New observations are given depth 0, and if a fact c is inferred from facts $p_1, p_2, ..., p_k$ with respective depths $d_{p_1}, d_{p_2}, ..., d_{p_k}$, then c is assigned depth $d_c = \min(d_{p_1}, d_{p_2}, ..., d_{p_k}) + 1$. Any facts with a depth greater than a maximum depth threshold are removed from the buffer in order to prevent indefinite recursion[4].

Two separate methods of reasoning are employed to infer new facts: a *top-down* method that attempts to infer new facts using the context of the current intended or expected plan step, and a *bottom-up* method that attempts to infer new facts from relevant facts in Eta's episodic memory. Each method of inference is accomplished using a specific transducer.

In addition to inferring new facts, Eta may also use reasoning to suggest possible courses of action from observed events. A separate *reaction transducer* is used to infer possible actions from each buffered observation; these actions are then added to the data buffer of potential actions.

[4]By default, we set this threshold to 3.

4.2.3 Planning

The planning process will attempt to modify the current dialogue plan using the modification strategies shown in Figure 4.2. First, Eta will pop all actions from the buffer of potential actions, and attempt to insert a subset of them into the current dialogue plan[5].

Second, if a modified plan exists in the plan data buffer, Eta will sequentially attempt to expand the current plan node, merge plan nodes, and reorder plan nodes, updating the plan data buffer if any change is made. Each plan modification type allows the use of a transducer that operates over a serialized form of the dialogue plan; however, the expansion step also supports multiple special cases. Special branching eventualities – namely *conditional* and *repetition* plan steps, such as the example in Figure 3.2 – will cause Eta to query the current dialogue context for the truth of the corresponding conditions, potentially expanding the current step with the matching conditional branch or a new copy of the repeating subplan, respectively. If a plan step matches the header of a dialogue schema in Eta's schema library, a copy of that schema will be instantiated and used to expand the current plan step. Finally, the plan step may be a member of a fixed set of special actions that use specific transducers to determine an appropriate expansion. For example, a `paraphrase-to.v` step by the agent will be expanded to form a primitive `say-to.v` action using a *paraphrase transducer* that maps a gist clause to a surface utterance, given the conversation history and relevant facts from memory as context; a `respond-to.v` step is handled using a similar transducer in cases where a gist clause is not specified.

[5] For simplicity, we currently only insert the highest priority action and abandon the rest. However, more complex methods for selecting actions for insertion may potentially be devised, such as determining the overall compatibility of each action with the current plan.

4.2.4 Execution

The execution process will attempt to either execute the current intended plan step, or attempt to match an expected plan step. In the case of an intended plan step, Eta will execute the step if it belongs to a set of *primitive actions* available to a particular agent, and the conditions for executing that action are satisfied by the current dialogue context. Each primitive action is associated with a procedural implementation that is invoked upon execution of that action; for instance, a `say-to.v` action will cause Eta to output an utterance (which may be associated with a particular affect (i.e., emotion) class using an *affect transducer*), as well as reflexively deriving a gist clause from the resulting utterance.

In the case of an expected plan step – either by the user, or pertaining to the environment – Eta will attempt to match the step to an eventuality in dialogue context, until some time period has elapsed. The time period that Eta may wait for an expectation is derived from the certainty score c associated with that step (see 3.1) using the formula $T = -\delta \log(1-c)$, where we set the coefficient $\delta = 30$ so that a certainty of about 0.632 corresponds to a 30 second delay. If an expected step has a certainty of 1, Eta will wait indefinitely to match that step. If the time period is exceeded without a successful match, Eta will characterize that step as a failure and advance the plan.

In any the case where an execution or match is successful (or in which the time period for a match was exceeded), the plan is advanced, and a list of variable bindings obtained from the execution or match is applied throughout the dialogue state. Additionally, if the plan was advanced, the plan data buffer is updated with the modified plan, allowing the planning process to make further modifications.

Analogously to the perceptual servers used by the perception process, Eta also supports integration with external *specialist servers* that implement narrow domain-specific reasoning – including temporal models, spatial models, and type ontologies. The specialist servers communicate with the core dialogue manager through natural language, ULF, or EL queries; these “server communication acts” are also enacted through primitive actions that may appear in the dialogue plan.

4.3 Transducers

All mechanized dialogue agents in some sense employ pattern matching and generation systems. In the case of autoregressive transformers, the matching and generation processes are hidden in the operation of the multiple layers of the encoder and decoder and their associated attention mechanisms; whereas in symbolic approaches the matching process seeks to match particular patterns, which may be at the word or part-of-speech level or at the level of more abstract phrasal constituents; and output may be template-based or generated in some more flexible way. We propose the use of *generalized pattern transduction* as a way to retain the robustness of early systems like ELIZA (Weizenbaum, 1966), while enabling the use of utterance context and responsiveness to the content and intent of user inputs at a much more specific level, setting the stage for construction of more relevant, more meaningful outputs.

A transducer in the Eta framework is a standalone module responsible for mapping from one representation (e.g., utterance, logical form, plan sequence, etc.) to another representation; Eta invokes several transducers throughout the processes described in Section 4.2 to carry out subtasks. We distinguish between the *type* of a transducer – the unique function signature of a transducer relevant to a particular subtask – and the *implementation* of a transducer. For example, a *gist* transducer takes an utterance and a conversation history as arguments and returns a list of gist clauses, but this mapping may be implemented using a variety of methods (or potentially an ensemble of methods whose results are combined). In Table 4.2, we summarize the types of transducers currently supported by the Eta framework.

Each process in Section 4.2 invokes particular types of transducers, but leaves the exact implementation of these transducer to be supplied in a specific agent configuration.

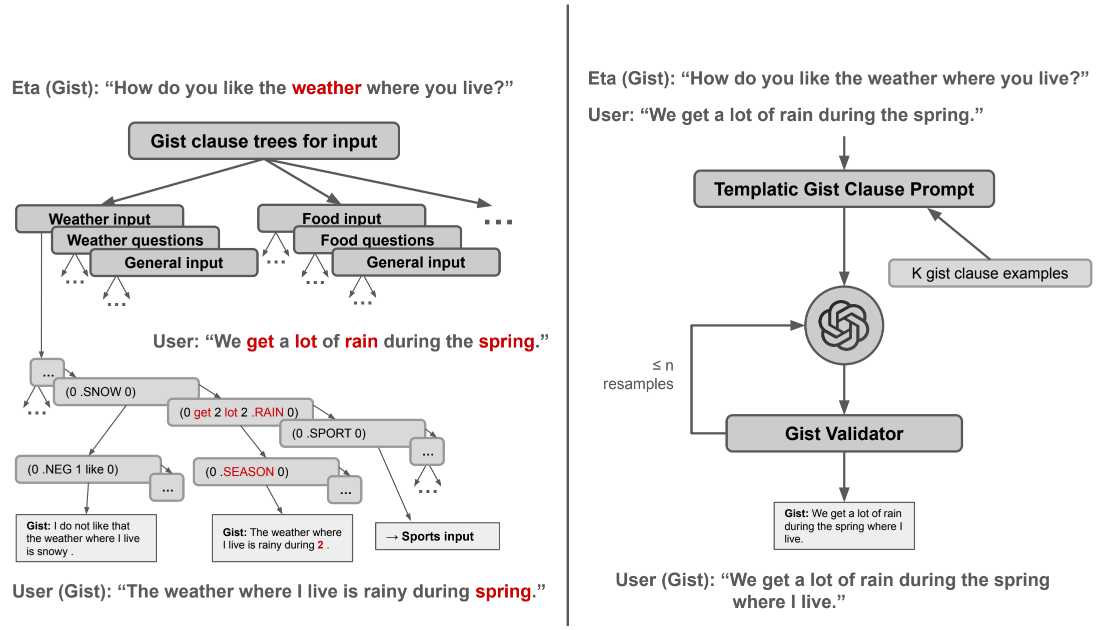

Figure 4.4: Examples of transducers for gist clause extraction implemented using hierarchical pattern transduction (left) and a large autoregressive language model (right). Although the implementations of each transducer differ, both map an utterance and a conversation history (only the gist of the prior turn is shown for brevity) to a gist clause capturing the decontextualized meaning of the original utterance.

While any method that satisfies the given function signature may be used to implement a transducer, the case studies that we present primarily rely on two methods that we describe in greater detail: *hierarchical pattern transduction* and *large language model (LLM) transduction*.

4.3.1 Hierarchical Pattern Transduction

The hierarchical pattern transduction method uses *transduction trees* based on inexact feature-based pattern matching to map an input expression to an output expression[6]. Transduction trees specify patterns at their nodes, with branches from a node provid-

[6]The library that we use for pattern matching can be found here: https://pypi.org/project/transduction/

Transducer	Argument Types	Return Type	Process
GIST	utterance + conversation history	gist clauses	Perception, Execution
SEMANTIC	gist clause	eventualities (semantic meaning)	Perception
PRAGMATIC	gist clause	eventualities (pragmatic meaning)	Perception
REASONING (TD)	plan step + eventualities (premises)	eventualities (inferred facts)	Reasoning
REASONING (BD)	memory + eventualities (premises)	eventualities (inferred facts)	Reasoning
REACTION	eventuality (observation)	eventualities (possible actions)	Reasoning
SUBPLAN	plan step	plan steps (subplan)	Planning
MERGE PLAN	plan	plan step (merged step)	Planning
REORDER PLAN	plan	reordered plan	Planning
PARAPHRASE	gist clause + conversation history + eventualities (relevant facts)	utterance	Planning
RESPONSE	conversation history + eventualities (relevant facts)	utterance	Planning
AFFECT	utterance + conversation history	affect class	Execution

Table 4.2: Types of transducers used within the Eta framework, their argument types, their return types, and the top-level Eta processes that invoke them. "TD" and "BU" abbreviate top-down and bottom-up, respectively.

ing alternative continuations as a match proceeds, following a recursive backtracking algorithm. Terminal nodes have associated *directives* indicating whether they provide a gist clause template, send input to some subordinate tree, or some other outcome. The pattern nodes use template-like S-expressions, allowing for arbitrary tree transduction. Atomic symbols in each pattern expression specify particular words, lexical features assigned to words, length-bounded "match-anything" sub-sequence spans, or special evaluable predicates matching one or more sub-sequences[7].

A simplified example of hierarchical pattern transduction for gist clause interpretation in a casual conversation domain is shown in Figure 4.4 (left). A top-level tree first matches keywords in Eta's previous gist clause to select a set of relevant subtrees to match to user input – in this case, the system selects trees for matching inputs and reciprocal questions related to weather that the user might give, as well as a tree containing general fallback rules for off-topic replies. Next, the system attempts to match the patterns within a selected subtree to the user's input in a depth-first order. In matching `(0 get 2 lot 2 .RAIN 0)`, `0` may match a word span of any length, while `2` may match at most two words. `.RAIN` may match any word assigned the lexical feature "rain", and likewise for `.SEASON` in the child pattern. Thus, this single pattern may match a number of different wordings of the user's response. In this example, the matching process results in the selection of a gist clause template directive that uses positional indices to borrow words that were matched by the previous pattern, producing the gist clause shown beneath. Directives at terminal nodes may also redirect input

[7]Following regular expression notation, for any given predicate we generate variants prefixed with `?`, `!`, `+`, and `*` that respectively match 0 or 1 sequences, exactly 1 sequence, 1 or more sequences, and 0 or more sequences.

to some other subtree, for instance if the user answered the question by talking about winter sports, matching the `(0 .SPORTS 0)` pattern instead.

4.3.2 Large Language Model Transduction

While hierarchical pattern transduction provides a flexible method of pattern transduction that can be quickly modified or extended by domain experts, it also requires hand engineering and may be brittle in the case of unanticipated phrasings of inputs. We also explore the use of autoregressive LLMs to implement various types of transducers, providing wider coverage at the expense of reliability and interpretability.

Each LLM-based transducer is equipped with a templatic prompt containing variables to be filled using the arguments of the transducer, as well as possibly k agent-specific examples for in-context learning. The LLM is conditioned on the filled-in prompt, and the generated response is passed through an explicit validator function. The LLM will be repeatedly prompted until it generates a valid response for that transducer type or a limit of n retries has been reached, in which case an empty result will be returned. An example of a LLM-based transducer for gist clause interpretation is shown in Figure 4.4 (right).

5 LISSA Virtual Agent for Conversational Practice

5.1 Domain

Social interaction is critical for maintaining meaningful relationships and overall well-being. Demographics that experience impairments in communication – such as individuals with autism spectrum disorder (ASD) or elderly adults with impairments in cognitive functioning – are at increased risk for social isolation and lower quality of life (Segrin, 2019, among others).

Spoken dialogue systems, together with simulated virtual humans, have been used to create digital companions and assistants (Vardoulakis et al., 2012; Yaghoubzadeh et al., 2013; Bernardini et al., 2021, among others) and virtual conversational coaches (Hopkins et al., 2011; Torres et al., 2019, among others) to allow at-risk groups of people to practice communication skills and reduce social isolation. However, these systems typically simulate narrow interaction scenarios that may not reflect everyday interactions.

We discuss a particular digital companion and social skills coach – LISSA – created with the goal of simulating natural, casual conversations, allowing users to improve their social skills in the comfort of their own home (Razavi et al., 2016, 2019). A primary challenge in this domain is that a digital companion system must be able to

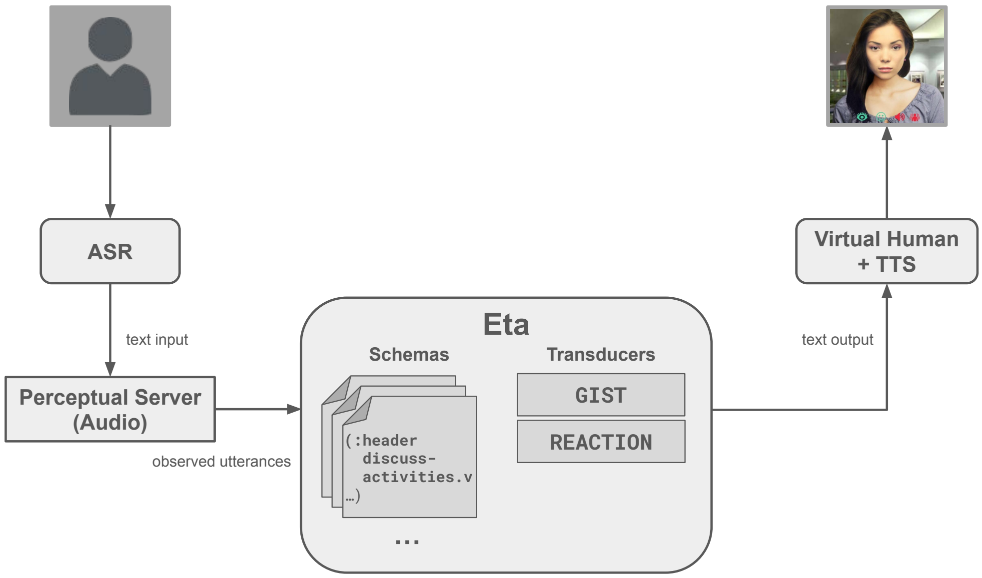

Figure 5.1: The full agent configuration used for the LISSA domain.

converse naturally and meaningfully across a broad variety of topics, keeping the user engaged in the conversation. Addressing this challenge served as the impetus behind the Eta dialogue framework, and consequently LISSA became the first application of the framework as it was used to design conversations targeting teenagers with Autism Spectrum Disorder and elderly adults at risk of isolation. The full configuration of Eta used to develop the LISSA agent is shown in Figure 5.1.

5.2 Schema Design

We developed two sets of schemas for two variants of the LISSA agent – one representing a younger adult (shown in Figure 5.1) and equipped with dialogue schemas that are relevant to ASD teenagers, and the other representing an elderly adult and equipped with dialogue schemas that are relevant to other elderly adults[1]. While some of these schemas

[1]The LISSA system was initially prototyped as a speed-dating conversational coach (Ali et al., 2015). Evaluations showed showed that feedback on facial and verbal behavior by the practice system indeed

overlap, many are unique (e.g., the former agent may have schemas for discussing school, whereas the latter may have schemas for discussing independent living).

Two dialogue sessions were developed for the LISSA variant used in the experiment with ASD teenagers, each containing 3 dialogue schemas that LISSA progresses through in sequence; 6 schemas in total. These dialogue schemas were greatly expanded in the case of the elderly LISSA variant; for that experiment, we designed 10 dialogue sessions, each with 3 dialogue schemas – 30 schemas in total. Each dialogue schema focused on a particular topical dialogue (e.g., "activities", "getting to know you", "pets", etc.), and contained 3-5 expected dialogue turn pairs wherein LISSA asks the user a particular question and the user responds to LISSA's question (implicitly, with LISSA adding a reaction to the user's response). Two examples of dialogue schemas for the elderly variant of LISSA are shown in Figures 5.2a and 5.2b.

5.3 Transduction Methods

Our symbolic approach in the initial LISSA variants employed the use of hierarchical pattern transduction – described in Section 4.3.1 – to allow for robust responses while enabling the understanding of utterance contexts and user intents. For maximum generality and transportability, we do not use a single search tree to transduce inputs (along with prior context) directly to outputs. Rather, we first apply our hierarchical pattern transduction method to derive a gist clause, which will typically decontextualize the input to a great extent and "clean up" and simplify the input. Then a further processing stage applies pattern transduction trees to the gist clause(s) to produce an appropriate reaction to their contents, just as if the user had directly produced the gist clause(s). Since these are no longer dependent for their meaning on the question that elicited them,

improved subjects' skill, as judged by clinical science graduate students; however, this prototype was based on a human-operated Wizard of Oz (WOZ) agent, so we omit it from the present discussion.

```
(dial-schema :header (((set-of ^me ^you)
  discuss-activities.v) ** ?e)
;``````````````````````````````````````
:episodes (
  ?e1 (^me say-to.v ^you '(Do you have any
    hobbies or anything in particular you
    like to do for fun ?))
  ?e2 (^you reply-to.v ?e1)

  ?e3 (^me say-to.v ^you '(I haven't been
    able to play sports recently, but I
    love to read . Do you like to read?))
  ?e4 (^you reply-to.v ?e3)

  [...]
))
```

(a)

```
(dial-schema :header (((set-of ^me ^you)
  discuss-technology.v) ** ?e)
;``````````````````````````````````````
:episodes (
  ?e1 (^me say-to.v ^you '(Lets talk about
    the new technology these days . Do you
    have a smartphone? If so, what do you
    use it for ?))
  ?e2 (^you reply-to.v ?e1)

  ?e3 (^me say-to.v ^you '(What do you
    think is the best part of new
    advances in technology ?))
  ?e4 (^you reply-to.v ?e3)

  [...]
))
```

(b)

Figure 5.2: Two examples of topical dialogue schemas created for the elderly LISSA agent; each consisting of 3-4 dialogue pairs beginning with a question by LISSA.

the transduction trees producing appropriate reactions are often portable across different kinds of conversation and conversational contexts.

5.3.1 Gist

The gist clause transduction method used for LISSA resembles the example shown in Figure 4.4 (left). A top-level tree first selects a set of topical subtrees based on LISSA's prior utterance using keyword-based patterns. For example, if LISSA's prior utterance were `?e1` in the schema shown in Figure 5.2a, the method would select a subtree for interpreting user responses related to activities, another for interpreting user questions related to activities (including reciprocal questions such as "How about you?"), and a fallback tree for interpreting miscellaneous inputs.

Each subtree is then matched to the user input in order to extract a gist clause, potentially borrowing terms from the input through templatic placeholders, such as in Figure 4.4. Any resulting gist clauses from each subtree are combined.

We created subtrees for each topic contained within the LISSA schemas; about 120 subtrees in total in the case of the elderly variant of LISSA. 3 dialogue developers were responsible for creating these subtrees, and each individual topic took about half a day to complete.

5.3.2 Reaction

We employed hierarchical pattern transduction to enable the selection of appropriate reactions to the user's input. In the case where the user's input contains multiple gist clauses – for example, a response followed by a question – we select and combine multiple reactions, e.g., a short acknowledgement followed by an answer to the question. Otherwise, LISSA selects a speech act reacting to the user's input in a manner that is consistent with LISSA's persona. For example, if the user were to tell LISSA about activities that they enjoy, LISSA might react by stating that she enjoys a particular activity as well, stating that she's never tried the activity but finds it interesting, etc. A fallback rule in the case where LISSA fails to extract a gist clause allows LISSA to react with a low-content acknowledgement of the user.

A reaction subtree was developed for each individual topic, along with the subtrees for gist clause extraction. A separate subtree is used for selecting answers to questions from the user based on LISSA's persona.

5.4 Evaluation

The two variants of the LISSA agent were evaluated through user experiments with the two respective demographic groups, in which users had multiple dialogue sessions

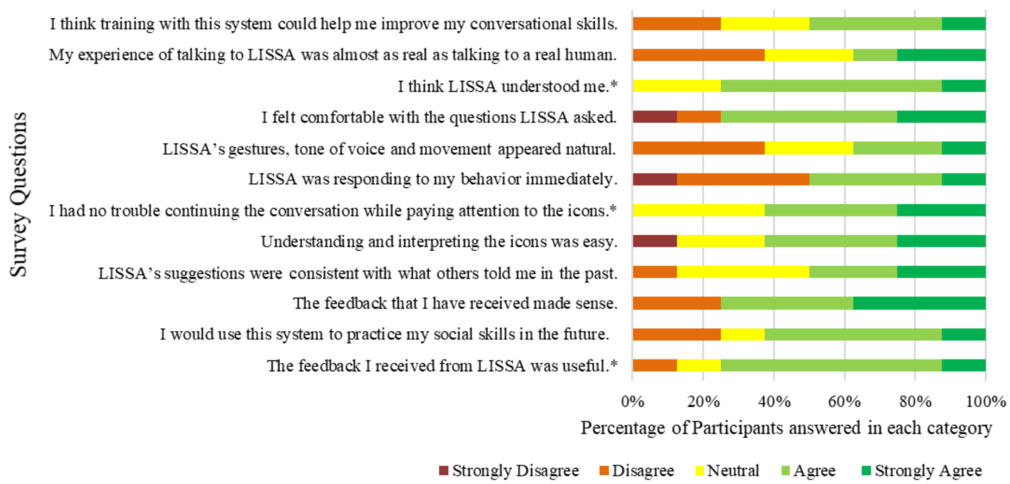

Figure 5.3: Evaluation results for LISSA in the ASD experiment; the distributions of survey responses for each question are shown. Responses whose means are statistically significant from "neutral" are indicated with a star (*).

with LISSA while being provided dynamic feedback between each. Ali et al. (2020) conducted an experiment with ASD teenagers, while Razavi et al. (2019) conducted an experiment with elderly adults at risk of isolation. The conversational performance of LISSA was primarily evaluated through post-session user surveys; we summarize these results below.

5.4.1 ASD Teenagers

In the first experiment, 9 teenage participants with high functioning ASD diagnoses were recruited to interact with the LISSA agent through the University of Rochester Medical Center (URMC) Department of Pediatrics. Each participant interacted with LISSA for a single session; within a session, a participant first conversed with LISSA for five minutes, then took a break for two minutes, and then held another conversation with LISSA for four minutes. Participants received real-time feedback on particular social skills (e.g., eye contact) through a graphical interface as they interacted with LISSA, as well as post-session summary feedback.

In the debrief interviews following their session, each participant filled out a survey that included questions about both the usability of the overall system and the conversational abilities of the LISSA agent. Users were asked to judge LISSA on each criterion using a five-point Likert scale ranging from ”strongly disagree” (=1) to ”strongly agree” (=5). These results are shown in Figure 5.3. Questions marked with a star (*) had responses that were significantly higher ($p < 0.05$) than “neutral” using a non-parametric bootstrap test with Bonferroni corrections.

5.4.2 Elderly Adults

In the second experiment, 9 elderly adults (≥ 60 years old) were recruited to interact with LISSA through the URMC facility, while an additional 10 participants were assigned to a control group. In this experiment, participants in the treatment group interacted with LISSA over 10 separate sessions, in order to evaluate the longer-term impact of LISSA on communication skills. These interactions were held in the users’ own homes using a personal computer or laptop, apart from the first and last sessions. Each interaction consisted of three short conversations separated by breaks where the user received feedback on their communication.

The transcripts from this experiment were compared to transcripts from a Wizard of Oz (WOZ) baseline in which the LISSA dialogues were simulated by human operators who were responsible for selecting appropriate responses (the topics were identical to those in the LISSA experiment); 25 elderly adults participated in this experiment over a single session. 8 transcripts were randomly sampled from both sets and were then randomly assigned to 6 research assistants who were blind to the study condition. Each RA was tasked to independently rate each assigned transcript on the following criterion using a five-point Likert scale ranging from ”strongly disagree” (=1) to ”strongly agree” (=5).

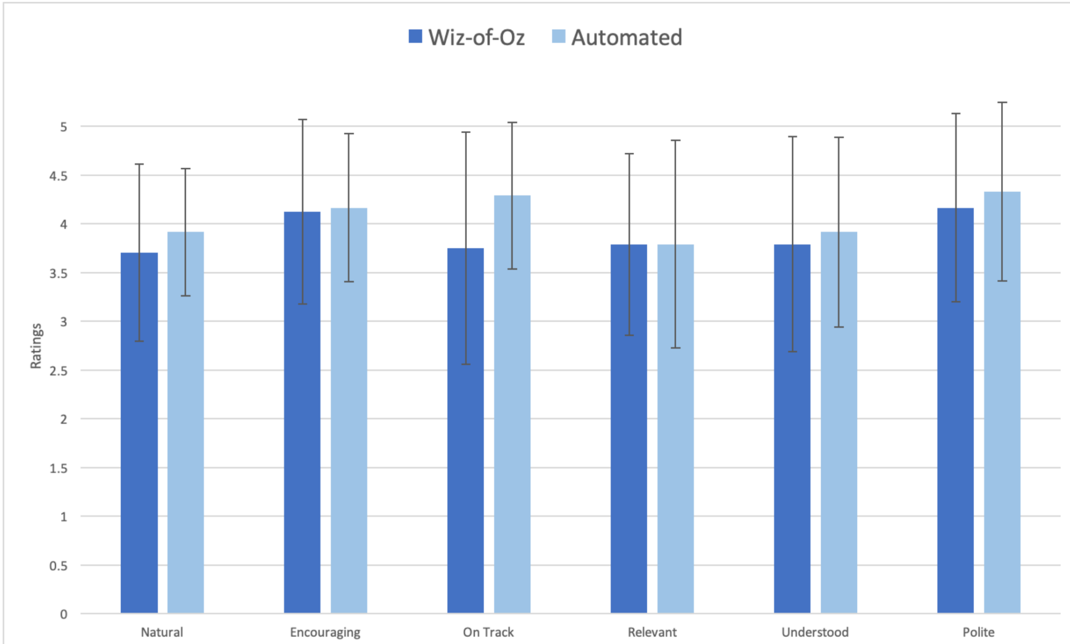

Figure 5.4: Evaluation results for LISSA in the elderly adult experiment; means and standard deviations of survey results are shown for LISSA and a Wizard of Oz (WOZ) baseline.

Q1. How *natural* were LISSA's contributions to the conversation?

Q2. LISSA's questions/comments *encourage* the user's participation.

Q3. The conversation stayed *on track*.

Q4. LISSA's responses were *relevant* to the conversation.

Q5. LISSA *understood* what USER said.

Q6. LISSA's responses were *polite* and respectful.

Figure 5.4 plots the mean responses and standard deviations for both the WOZ baseline and the full experiment. The responses had high internal agreement (Cronbach's alpha = 0.89). The automated LISSA system performed comparably well to the WOZ baseline; most of the mean responses were slightly higher for the LISSA system, but none of the differences are statistically significant. We show a qualitative example of a dialogue from the LISSA experiment in Figure 5.5.

```
LISSA: Tell me about your day. For example, what did you have for breakfast?
User:  Well I had a granola bar, a banana and a cheese stick. I didn't take the time
       for cereal and milk which is my usual. Then I came over here to this to this
       session and that's been my day so far. I didn't have any trouble with traffic.
LISSA: I try to have fruit for every breakfast. It's very healthy for me. Speaking of
       food, what's your favorite flavor of ice-cream?
User:  Oh, gosh... I like all sorts of ice-cream. Maybe... maybe coffee ice-cream
       would be my favorite although chocolate is right up there too.
LISSA: I think coffee ice-cream is pretty tasty. Though I am a big fan of mint
       chocolate chip. Do you have a favorite food? What is it and why do you like it?
User:  Oh fruit. I eat all sorts of fruit. Um... I suppose the favorite is grapefruit.
       Uh but I try to eat a banana, banana every day or almost every day.
LISSA: I like grapefruit. It's nice when food is healthy and tastes good. How did
       you get here today? Did someone drive you or did you take the bus?
User:  No, I drove my car here today. It's a good day to be out driving.
LISSA: That's good that you drive. It must be nice to have the freedom to go
       somewhere when you want to.
User:  It is.
...
```

Figure 5.5: Example dialogue from the LISSA aging and engaging experiment.

5.5 Discussion

In both experiments, we observe that LISSA was able to converse with users in a manner that was judged to be fluent and that enabled useful social skills feedback to be provided to the user. In particular, the participants in the ASD experiment felt that LISSA understood them, while raters in the experiment with elderly adults felt that LISSA was natural, on track, and demonstrated understanding of the user. Moreover, the results from the latter suggest that LISSA was able to produce conversations that are approximately as good as the human-operated WOZ agent, attesting to the ability of the Eta framework to enable flexible, topically broad dialogue.

One advantage of the schema-based dialogue framework – as opposed to black box statistical dialogue models – is the ability for dialogue designers to control the types of topics, questions, and responses employed by an agent. The LISSA agent was developed using guidance provided by psychiatric experts, with particular care taken to build rapport through incremental self-disclosure and to gradually increase emotional intensity over subsequent sessions. We observe, likely as a consequence of this deliberate design, that LISSA achieved particularly high scores for the ”Encouraging”, ”Polite”, and ”On Track” metrics.

These initial experiments, however, were relatively small-scale; since participants would only interact with LISSA for several sessions, we were able to use a restricted set of schemas and topical knowledge, with LISSA’s reactions generated through hard-coded pattern transduction trees. A major limitation of pattern transduction trees is the effort required to scale them to a wider set of topics. In Section 5.6, we discuss a technique that allows for an expansion of the topical knowledge available to LISSA – particularly the *habitual* knowledge that constitutes LISSA’s persona, allowing LISSA to use this knowledge to generate interesting and relevant reactions using LLMs instead of transduction trees.

5.6 Improving Response Generation using Habitual Schemas

A critical challenge in the design of conversational agents such as LISSA is endowing them with a specific *persona*, and generating responses that are both natural and consistent with this persona across a variety of topics. Systems that are able to do this are both found to be more engaging by users (Zhang et al., 2018), and increase the level of confidence and trust that users place in the system (Shum et al., 2018). Furthermore, in many practical applications beyond chit-chat, there is a complementary need to control the flow of dialogue; for example, ensuring consistency of generated responses with hand-engineered templates may help to improve a dialogue system’s topical coherence (Grassi et al., 2021). Recent work has focused on combining explicit representations of personas and knowledge with LLMs using retrieval-in-the-loop methods for generation (Shuster et al., 2021). Typically, these approaches represent personas using unstructured sets of natural language "facts" about an agent, possibly augmented with additional knowledge from a knowledge base.

In casual human-human dialogue, however, personas are often revealed through story-like narratives about experiences rather than one-off facts (Dunbar et al., 1997). For example, if a speaker mentions something involving sports, the interlocutor might respond by relating their typical experiences playing a sport in the past. These types of narratives, typically taking the form of "generic passages" (Carlson and Spejewski, 1997), often capture *habitual knowledge* – knowledge about the kinds of events that an agent participates in, or used to participate in. This knowledge includes the typical steps of a habitual event, as well as the typical goals, preconditions, and postconditions of the event. Event schemas, in addition to representing prototypical dialogue events, can be used to capture the rich types of prototypical knowledge associated with generic and habitual events, such as causal and enabling relations, temporal relations, etc. (Chambers, 2013; Lawley et al., 2021; Li et al., 2021).

In this section, we present novel approach to dialogue generation that uses a collection of explicit event schemas to augment an agent's persona, and that conditions an LLM to generate narrative-like responses consistent with these schemas through in-context prompting[2]. Furthermore, since it is often desirable for dialogue designers to be able to specify a persona using a small number of simple natural language facts – such as in the case of LISSA – we propose a method for *bootstrapping* the creation of schemas from a set of simple facts. This method involves leveraging LLMs to first generate "generic passages" from the given facts, and then to induce structured habitual schemas from the passages – capturing both explicit steps from the passage and implicit knowledge associated with the event described by the passage. A high-level diagram of our approach is shown in Figure 5.6.

[2]Code can be found at https://github.com/bkane2/habitual-response-generation

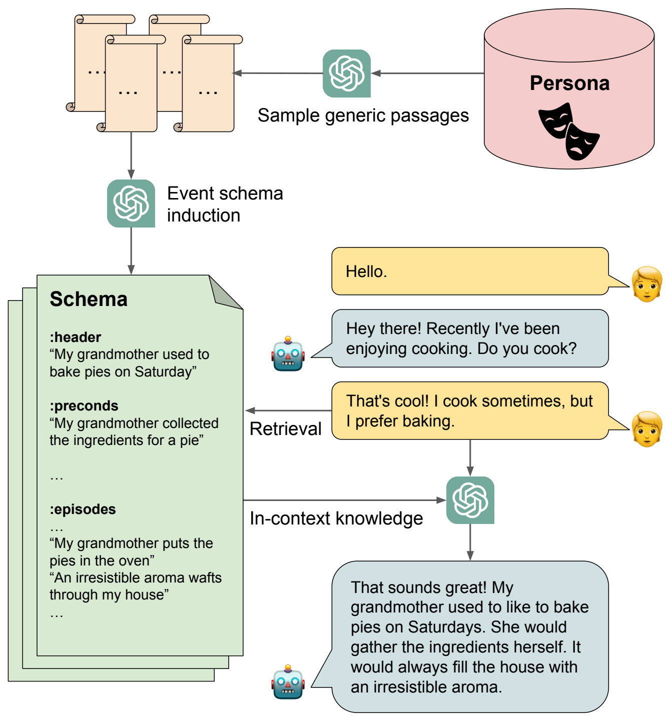

Figure 5.6: A diagram of our approach. (1) Given an unstructured persona dataset, we first sample "generic passages" from the facts in the persona, and then induce structured event schemas from the sampled stories. (2) We condition an LLM to generate dialogue responses that are fluent with previous conversation – yet that make use of the rich knowledge contained in the resulting schemas – by first using a retrieval model to select a relevant schema, and then providing the schema to the LLM as in-context knowledge.

We present evaluation results showing that the generated schemas are generally high quality, and can be used to condition LLMs to generate responses that are more diverse and engaging, yet also controllable. In order to perform a large-scale evaluation, we turn to the PersonaChat dataset (Zhang et al., 2018) to generate open-domain responses across a variety of topics. However, this approach can also be used to create habitual schemas given a particular LISSA persona, and in Chapter 7, we employ a similar mechanism for response generation.

5.6.1 Method

Given a dialogue context $\mathcal{U} = \{u_1, u_2, ..., u_{n-1}\}$ containing system and user utterances, our goal is to generate a response u_n that utilizes knowledge from a relevant event schema $S^R \in \mathcal{S} = \{S_1, S_2, ..., S_m\}$ – this schema represents knowledge about a habitual activity that is part of the speaker's persona and that is relevant to the previous turn u_{n-1}. We ensure that the selected schema is relevant using a multi-level information retrieval system to embed both the event schemas (treated as individual documents) and the knowledge contained within each event schema (treated as collections of documents), and to rank the schemas in $\mathcal{S}$ based on similarity to the embedding for u_{n-1}.

Following Zheng and Huang (2022), we employ a prompting-based approach in which a pre-trained LLM is used to produce a response utterance, provided a prompt that is dynamically constructed from the dialogue history and the selected schemas.

Schema Induction

Since structured event schemas for habitual activities are typically expensive for dialogue designers to create, requiring reasoning about causal relations and other implicit knowledge, we focus on the problem of automatically inducing event schemas from an unstructured persona $\mathcal{P} = \{p_1, p_2, ..., p_n\}$, where p_i are natural language "facts" such as "I like to play tennis."[3]. Formally, we represent an event schema as a tuple $\langle \mathtt{H}, \mathtt{Pr}, \mathtt{S}, \mathtt{Po}, \mathtt{G}, \mathtt{E} \rangle$. H is a schema *header*; a sentence characterizing the overall schema event. Pr, S, and Po are sets containing schema *preconditions*, *static conditions* (conditions expected to hold throughout the overall event), and *postconditions*, respectively. G is a set containing typical goals of participants of the event, and E is a set containing typical episodes (i.e., substeps) of the event. We show an example of an event schema in Figure 5.7.

[3]These facts may be hand-designed by a dialogue designer, crowdsourced (as in (Zhang et al., 2018)), or generated by an LLM.

```
:header "I work in a bookstore."

:preconds (
  "My shift has started.")

:static-conds (
  "The bookstore is stocked with books."
  "Customers visit the bookstore."
  "I am knowledgable about books and customer service.")

:postconds (
  "My shift at the bookstore is over."
  "Some customers have purchased books.")

:goals (
  "My goal is to assist customers in finding the books they are looking for."
  "The customers' goal is to find the books they want to purchase.")

:episodes (
  "Customers come looking for new titles to add to their collection, or to browse."
  "I welcome the customers and ask if they need any assistance."
  "I help the customers find books by using my knowledge of the store's inventory."
  [...]
  "I organize the bookshelves when the customers are not in the store.")
```

Figure 5.7: An example of an event schema for a habitual "work at bookstore" activity. Note that some episodes are omitted for brevity.

In order to generate sufficiently interesting and accurate schemas, we employ the method of *latent schema sampling (LSS)* introduced by Lawley and Schubert (2022) – this method regards an LLM, when conditioned on a schema header, as implicitly characterizing a distribution over stories sampled from that distribution. A full schema can then be induced from the sampled stories.

Thus, for each $p_i \in \mathcal{P}$, we sample N_p stories (specifically, *generic passages* (Carlson and Spejewski, 1997) describing the typical process of a habitual event) using the

GPT-3.5-TURBO LLM[4]. We use a few-shot prompt in which the LLM is supplied with a short definition of a generic passage, followed by K_p examples. In contrast to the neuro-symbolic pipeline in (Lawley and Schubert, 2022), we leverage the in-context learning capabilities of GPT-3.5-TURBO to directly induce an event schema from a set of N_p passages, given an abstract schema template and K_s in-context examples[5].

Dialogue Generation

We use the GPT-3.5-TURBO LLM to generate fluent responses, conditioned on a prompt containing a subset of the knowledge contained within a retrieved schema. Additionally, in order to allow controllable dialogue flow management – which is necessary for usability in many applied domains (Grassi et al., 2021) – we allow for two modes of generation: *unconstrained generation*, in which the LLM is prompted with the entire dialogue history and generates the next utterance without any constraints (apart from the retrieved knowledge); and *few-shot paraphrase generation*, where the LLM is prompted with a given sentence to paraphrase along with several in-context examples. In practice, the mode of generation may be mediated by a dialogue manager that manages the conversation flow and provides "raw" utterances (which may, for instance, be programmed by dialogue designers) to be selected for paraphrasing. For the purposes of this evaluation, we assume that, in the case of paraphrase generation, we have raw utterances available.

Schema Retrieval As a first step in constructing a prompt, we use the multi-level schema retrieval algorithm described in Section 4.1.1 to retrieve relevant schema knowledge. In our experiments, we use a pre-trained Sentence Transformer model[6] (Reimers and Gurevych, 2019) as an embedding function. As a query string, we embed the

[4] https://platform.openai.com/docs/models/overview

[5] In practice, we found $N_p = 1$, $K_p = 2$ and $K_s = 1$ to be sufficient to produce accurate generations.

[6] https://huggingface.co/sentence-transformers/all-distilroberta-v1

previous utterance u_{n-1} for each dialogue turn. The top N_f facts are retrieved to be used in the prompt.

Unconstrained Generation If the current episode of a dialogue schema corresponds to a basic speech act (e.g., “say-to”, “tell”, etc.), we employ an unconstrained generation mode where the response is sampled from the LLM by prompting with the full dialogue history, after conditioning on facts from the relevant habitual schema and the current dialogue schema:

$$u_n \sim \text{LLM}(F_R \mathbin{+\!\!+} F_D \mathbin{+\!\!+} \mathcal{U}),$$

where $F_R = \{f_1, ..., f_{N_f}\} \subset S^R$ are the relevant facts retrieved in the previous step, $F_D = S^D \setminus \texttt{E}(S^D)$ are all non-episodic facts from the current dialogue schema (i.e., preconditions, goals, etc.), and $\mathcal{U} = \{u_1, ..., u_{n-1}\}$ is the dialogue history.

Few-shot Paraphrase Generation If the current episode of a dialogue schema specifies a “paraphrase-to” act with a specific sentence as an argument, we employ a few-shot prompting strategy to condition the LLM to paraphrase the given sentence in a manner that is interesting, appropriate, and that makes use of the relevant facts. Specifically, in addition to the inputs used in the unconstrained setting, we format several in-context paraphrase examples along with a “raw” utterance to paraphrase, given the actual dialogue context:

$$u_n \sim \text{LLM}(F_R \mathbin{+\!\!+} F_D \mathbin{+\!\!+} \mathcal{E} \mathbin{+\!\!+} \mathcal{U} \mathbin{+\!\!+} \hat{u}_n),$$

where $\hat{u}_n$ is the sentence to paraphrase, and $\mathcal{E}$ is a set of K_e in-context examples: $\mathcal{E} = \{(\mathcal{U}^1, \hat{u}_n^1, u_n^1), ..., (\mathcal{U}^{K_e}, \hat{u}_n^{K_e}, u_n^{K_e})\}$.

5.6.2 Experiments

We first evaluate our response generation method according to the following desiderata: (1) the generated responses improve diversity of output; (2) the generated responses are engaging, interesting, and relevant given the previous conversation, and (3) the generated responses are controllable; i.e., a dialogue designer can ensure that the responses still correctly express an intended response.

Since an important advantage of our approach is the reusability of the generated schemas for downstream tasks (e.g., for inferring additional facts from a dialogue agent's experiences), we also conduct an evaluation of the quality of the generated schemas – specifically, whether the facts within the schema correctly represent typical knowledge associated with the event that the schema describes.

Dataset We conduct our experiment using the PersonaChat dialog dataset[7] (Zhang et al., 2018). We generate schemas and evaluate the performance of our response generation method using the test split, containing of 131,438 unique utterances. When evaluating our paraphrase generation method, we use the gold response annotations from the PersonaChat dataset for the raw utterances that are input to the model.

Baselines We consider two baselines for evaluating the performance of our approach: First, we use the GPT-3.5-TURBO LLM without schema retrieval, provided only with the base persona and dialogue history in the prompt (**BASE**). Second, we consider the human-generated gold utterances from the PersonaChat dataset themselves (**GOLD**) as a baseline for our diversity, engagement, and relevancy metrics. Against these, we compare our two generation methods: unconstrained generation **UNCS** and paraphrase generation **PARA**. The differences between the three generation methods are summarized in Table 5.1 for reference.

[7] https://huggingface.co/datasets/bavard/personachat_truecased

	BASE	UNCS	PARA
Base Persona	✓	✓	✓
Dialogue History	✓	✓	✓
Event Schema	✗	✓	✓
Raw Response	✗	✗	✓

Table 5.1: A summary of the differences in the resources available to each method that we compare in our evaluations. Note that each method in the order presented has access to all resources available to the previous method.

Response Generation Evaluation

Automatic Evaluation Following prior work (Majumder et al., 2021; Li et al., 2016), we use several methods to measure the diversity of the generated outputs, per desideratum (1). First, we compute the mean percentage of unigrams and bigrams in the generated outputs that are distinct relative to the total number of generated words, reported as **D-1** and **D-2** respectively. We also report the mean lengths of the outputs as **Length**. Since the distinct n-gram measures do not represent the actual frequency distributions of words (and will tend to be penalized with longer responses), we also report the mean **ENTR** score across outputs – calculated as the geometric mean of entropy values of n-gram frequency distributions, for $n \in \{1, 2, 3\}$.

In order to test the controllability of our paraphrase generation method against other baselines, per desideratum (3), we also report several text similarity methods computed between a generated output and the gold PersonaChat response. We report widely-used n-gram-based similarity metrics such as **BLEU**, **ROUGE-L**, and **METEOR**, as well as the cosine similarity between contextualized embeddings produced by the ALL-DISTILROBERTA-V1 Sentence Transformer model (Reimers and Gurevych, 2019) (**ST**). However, since not all sentences in a generated response may be directly related to the gold response (e.g., an acceptable paraphrase may consist of a story followed by the

Method	GOLD	BASE	UNCS	PARA
Diversity				
LENGTH	50.1	122	303	372
D-1	**97.0**	93.8	81.7	78.9
D-2	88.9	94.2	96.0	**96.7**
ENTR	2.20	2.91	3.61	**3.84**
Controllability				
BLEU	-	1.25	.843	**8.60**
ROUGE-L	-	19.3	19.8	**34.6**
METEOR	-	14.6	16.5	**33.2**
ST	-	35.6	35.0	**55.6**

Table 5.2: Diversity and controllability metrics on the PersonaChat test set. D-1/2 are the % of distinct uni- and bi-grams; ENTR is the geometric mean of n-gram entropy. BLEU, ROUGE-L, and METEOR are standard n-gram-based similarity metrics, and ST is the Sentence Transformer similarity measure. All similarities are calculated as average maximum pairwise values across sentences in each response. Best scores are bolded.

intended response), it is difficult to interpret these metrics on the level of the full response. Hence, we compute the maximum *pairwise* similarity for each full sentence[8] between the generated and gold responses, and report the average value across all responses.

These results are shown in Table 5.2. We observe that the methods that use event schemas for generation generate responses with higher diversity than the baseline methods that do not have access to the schemas, as measured by D-2 and ENTR (although D-1 tends to favor the methods that generate responses that are shorter and therefore have a higher relative fraction of distinct uni-grams). Furthermore, we observe

[8]Split based on ".", "?", and "!" punctuation, filtering out sentences less than 5 words in length.

PARA vs.	UNCS		BASE		UNCS v BASE		BASE v GOLD	
Metric	win	loss	win	loss	win	loss	win	loss
Engaging	34.7	27.4	46.8*	21.1	39.5*	24.2	43.0*	23.0
Relevant	33.2	23.7	44.7*	24.2	37.4	22.6	40.5*	25.0

Table 5.3: Pairwise comparisons between responses generated from each method (% win/loss, leaving ties out). Entries with * are statistically significant with $p < 0.05$ using a non-parametric bootstrap test on 2000 subsets of size 50.

that the paraphrase generation method achieves considerably higher similarity to the gold responses than both the baseline and unconstrained methods (which perform comparably well on this metric).

Human Evaluation To assess desideratum (2), we conduct a human evaluation of 100 randomly sampled examples on two metrics associated with response quality, following prior work (Majumder et al., 2021) – namely, whether the generated responses are **engaging** and **relevant** given the dialogue context. Annotators are tasked to make a pairwise comparison between responses from a pair of generation methods. We first collect annotations comparing the two baseline methods; under the assumption of transitive preferences, we then use the "winning" baseline as a comparison for each proposed method. We hired two Anglophone annotators for every sample.

Our results are shown in Table 5.3, with starred values indicating differences that are significant with $p < 0.05$, using non-parametric bootstrap tests on 2000 subsets of size 50. The collected annotations are fairly noisy, with inter-annotator agreement (Krippendorff's alpha) being 0.21 and 0.23 for "engaging" and "relevant", respectively.

Despite this, we were able to observe moderate and statistically significant preferences for both the paraphrase and unconstrained methods over the LLM baseline in terms of engagement, and for the paraphrase method over the baseline in terms of relevancy.

The LLM baseline itself was, in turn, significantly preferred over the gold responses for both questions. We believe that this can be attributed to the relatively short length and low diversity of language of the gold responses (as indicated in Table 5.2), as well as the ability of LLMs to interpolate smoothly with conversation history, even when constrained by our proposed methods.

We note, however, that many annotators were indifferent between the different generation methods. This is plausibly due to the fact that, generally, multiple response strategies are considered acceptable for the open-ended conversations in the PersonaChat dataset, and attests to the capability of LLMs to generate suitably engaging and relevant responses across prompting strategies.

Schema Evaluation

We evaluate the quality of the schemas, in themselves, through another human evaluation. We randomly select a subset of 200 individual schema facts from all generated schemas, each paired with the header of the schema it was taken from. An equal number of facts are selected for each type of schema relation. As a baseline, we select another 200 facts from the generated schemas, but randomly swap schema headers so that facts are paired with headers from unrelated schemas. For each type of schema relation, given a fact of that type and a schema header, we hire two Anglophone annotators to rate, on a 5-point Likert scale, how typical the fact is of an event described by the schema header. For instance, for a "static-condition" fact, an annotator might be asked "How typical is it that Sentence 2 is true throughout the duration of the event in Sentence 1?".

The mean Likert ratings for the baseline and the generated schemas are shown in Table 5.4. All differences are significant with $p < 0.05$ using a Mann Whitney U test. We observe that the generated schemas are generally found to contain facts that are typical of the described event, relative to the randomized baseline. The smallest typicality differences were observed for the "postcondition" relation, suggesting that inferences of this type may be more complex than other schema relations.

Relation	BASE	SCHEMA
PRECONDITIONS	2.51	3.65
STATIC-CONDITIONS	2.89	3.74
POSTCONDITIONS	2.93	3.23
GOALS	2.99	3.55
EPISODES	2.80	3.36
ALL	2.82	3.50

Table 5.4: Mean Likert ratings for the baseline and generated schemas; both the aggregate value and disaggregated values across schema relation types are shown. All differences are statistically significant with $p < 0.05$.

Qualitative Analysis

Table 5.5 shows generated responses from different methods for a particular persona and context. Qualitatively, we observe that the two models that are conditioned on the habitual schema from Figure 5.7 are able to generate longer and more detailed responses, making use of generic knowledge such as that people who work at bookstores can generally help customers find books of interest. On the other hand, the baseline model tends to generate responses that are fairly short and open-ended[9]. Furthermore, we observe that the paraphrase method is more frequently able to preserve the meaning of the intended raw utterance, as indicated.

Discussion

In this section, we demonstrated that habitual knowledge in the form of explicit event schema representations could be used to condition LLMs to generate more diverse and

[9]One important caveat is that this behavior is not necessarily undesirable; short open-ended questions can often be used in a conversation to demonstrate interest or empathy towards the interlocutor, although in this section we are focused on the challenge of generating more engaging responses.

Persona: I do not care about fashion. I work in a bookstore. ...
User: I worked at Petsmart for two years and then I got canned.
GOLD: Ah too bad! I work at a barnes and noble. Chemistry grad.
BASE: Oh really? Why did you get fired?
UNCS: Oh no, sorry to hear that. But hey, maybe your knowledge of pets could come in handy if you're interested in finding some books on the subject. As someone who works at a bookstore, I'm pretty knowledgeable about different books [...]
PARA: That's tough. I actually work at a barnes and noble bookstore. Although, I didn't major in English or literature. I graduated from college with a chemistry degree. When I'm not shelving books, I help customers find what they're looking for [...]

Table 5.5: Sample generations by different methods, using the schema from Figure 5.7. Content in the gold utterance preserved by the paraphrase method are highlighted in red.

engaging dialogue responses. We experimented with two generation settings, one of which furthermore allows for a greater degree of controllability by a dialogue designer who may wish to provide intended utterances for the LLM to paraphrase. Moreover, to ease the burden of schema design, we proposed a novel method of inducing schemas from a base persona using an LLM through sampling "generic passages" about habitual activities.

Although the inclusion of habitual knowledge can be used to produce more engaging responses, it is not sufficient – often, conversations focus around more specific experiences and memories, and the knowledge captured by schemas generated with our approach can be somewhat banal. In future work, we aim to extend our approach to generate schemas that capture *atypical* aspects of an agent's experience with a particular kind of event, as well as more ordinary memories or knowledge.

Moreover, although our method succeeds at generating more diverse, and engaging responses, this can often be inappropriate in certain conversational contexts, such as in a scenario that calls for a short affective response from the agent rather than a lengthy narrative-like response. Such responses may become repetitive over the course of a full conversation. When integrated into a broader dialogue management framework such as Eta, however, this response generation method can be balanced with other strategies in order to enable engaging conversation for agents such as LISSA.

6 DAVID Spatially-Situated Blocks World Agent

6.1 Domain

The "blocks world" domain has a rich history in AI research as a testing ground for spatially-situated conversational agents and robots, beginning with the simulated SHRDLU collaborative planning system (Winograd, 1972). Although the domain is highly idealized, achieving high performance in complex blocks world tasks requires capabilities such as deep semantic understanding and spatial reasoning – competencies that are also critical for more general domains, such as a "room world" with everyday objects.

We used the Eta framework to create a virtual conversational agent – DAVID – that can hold collaborative dialogues with a user within a physical blocks world setting, depicted in the top of Figure 6.1. Our setup consists of 9 blocks that can be referred to by associated company names (e.g., "the Twitter block") as well as by color (indicated by the tape along the edges of the blocks); the scene is captured by two Kinect sensors and reconstructed in the Blender 3D modeling program. In the full agent configuration – shown in 6.1 – both the ASR module and "vision"/state-tracking component of the Blocks World system are connected to Eta via *perceptual servers* that send relevant observations to Eta, while the "spatial component" is connected to Eta as a *specialist*

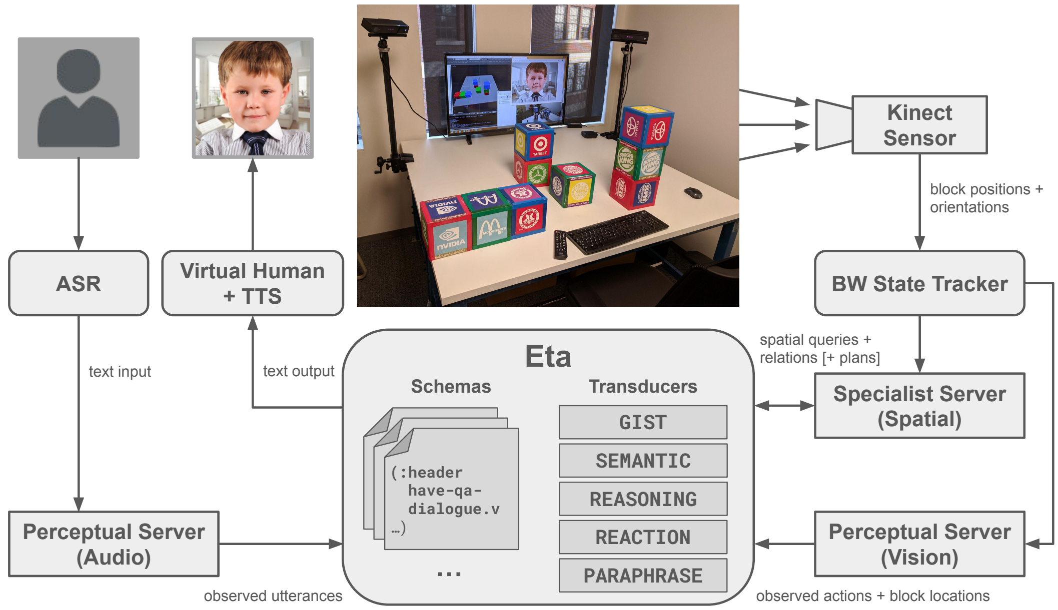

Figure 6.1: The full DAVID agent configuration used for the blocks world domain, with the physical blocks world apparatus pictured in the top center.

server capable of task-specific reasoning (e.g., determination of spatial relations between blocks, spatial planning, etc.) given queries from Eta. This specialist subsystem supplies Eta with the ability to use fully general mathematical models of spatial relations (such as "near" or "behind") that can ultimately be re-used in more realistic domains (Platonov and Schubert, 2018; Platonov et al., 2021b).

The DAVID virtual agent was initially configured for an interactive question-answering (QA) task, with the aim of developing the reasoning capabilities necessary for a full collaborative interaction. In this task, users are able to move blocks around on the table while asking the agent free-form questions about the world. These questions may include *spatial questions* about relations or properties of the blocks on the table and *historical questions* about past world states or actions, such as the following examples:

- *"What color is the leftmost block?"*
- *"Where is the Texaco block?"*

- *"How many red blocks are below a blue block?"*
- *"Which blocks were near a red block in the beginning?"*
- *"What block have I moved three times?"*
- *"Where was the Twitter block before I moved it?"*

6.2 Schema Design

Given the relatively structured dialogue flow of the QA task, development of the DAVID agent required only a small set of dialogue schemas. A top-level schema, shown in Figure 6.2a, specifies a repeating dialogue episode wherein DAVID prompts the user for a question, followed by the expected event that the user will reply. This continues until the user is observed to say goodbye to the agent. Throughout an interaction, DAVID may react with several subschemas for handling particular types of user responses – including answering spatial and historical questions, saying goodbye to the user, temporarily pausing the conversation, and reacting to "smalltalk".

An example of a subschema for reacting to a spatial question is shown in Figure 6.2b. The schema first matches an expectation that the user *articulates* a spatial query to DAVID[1]. This is followed by DAVID querying a spatial specialist server and receiving a set of relations satisfying the query. The spatial specialist server uses the constraint satisfaction algorithm described by Platonov et al. (2020) to obtain the set of relations satisfying a query logical form, along with associated confidence scores (allowing DAVID to insert hedge words into its paraphrased responses in the case of low confidence). For example, given a spatial query such as `((|NVidia Block| (be.v where.pro)) ?)` and the block configuration in Figure 6.1, the specialist server may return the relations `(|NVidia Block| next_to.p |McDonald's`

[1]We use `articulates-to.v` to represent a verbalization of an underlying semantic meaning, cf. the use of `paraphrase-to.v` to represent paraphrasing of a gist clause.

```
(dial-schema :header (((set-of ^me ^you)
  have-QA-dialogue.v) ** ?e)
;````````````````````````````````````````
[...]

:episodes (
  ?e1 (^me say-to.v ^you
    '(Hi, I'm David. I'm ready to answer
      your spatial questions.))

  ?e2 (:repeat-until (^you say-bye.v ^me)

    ?e3 (^me say-to.v ^you
       '(Do you have a question for me?))

    ?e4 (^you reply-to.v ?e3))))
```

(a)

```
(dial-schema :header
 ((^me react-to-spatial-question.v) ** ?e
;````````````````````````````````````````
:episodes (
  ?e1 (^you articulate-to.v ^me ?ulf)

  ?e2 (^me query.v |Spatial-Server| ?ulf)

  ?e3 (^me receive-from.v
       |Spatial-Server| ?relations)

  ?e4 (^me paraphrase-to.v ^you
       (answer-to.f ?ulf ?relations))))
```

(b)

Figure 6.2: The main dialogue schema used for the blocks world QA task (a) and a schema for reacting to a spatial question (b).

`Block|)` and `(|NVidia Block| left_of.p |Texaco Block|)` with certainties 0.9 and 0.8, respectively. Finally, DAVID uses the relations and query ULF to generate an answer to paraphrase to the user; `answer-to.f` is an evaluable function that substitutes satisfying relations for a WH-term in the query ULF and applies syntactic transformations to generate an answer clause. For example, given the above relations, DAVID might output "The NVidia block is next to the McDonald's block and to the left of the Texaco block".

DAVID is able to react to historical questions using a similar subschema that instead queries a temporal specialist server based on the temporal constraint solver described in Section 6.4.

6.3 Transduction Methods

The DAVID agent is configured with several transducers depicted in Figure 6.1 and elaborated in the following section.

6.3.1 Gist

Given a user input, a rule-based gist clause transducer is first used to both classify intent and preprocess the input into a simplified form for semantic parsing. Inputs are initially classified as spatial/historical questions, special requests (e.g., to pause the conversation), or "smalltalk". In the first case, the preprocessing steps include fixing common errors introduced by the ASR system, combining multi-word predicates (e.g., `to_the_left_of`), and trimming fillers or tag questions.

6.3.2 Semantic

The semantic transducer plays an important role in DAVID's interpretive pipeline since it allows for deep semantic understanding of user inputs, rendering them into a form that can be processed by the specialist servers. We employ the hierarchical pattern transduction algorithm described in 4.3.1, augmented with phrase-based recursion; this enables a form of compositional semantic parsing that is quite efficient and accurate for the linguistically constrained blocks world domain, yet extensible.

An example transduction tree used for parsing a historical question into ULF is shown in Figure 6.3. A top-level tree identifies different types of input sentences and accordingly sends them to more specialized trees. These trees again use hierarchical pattern matching based on words and their features to identify meaningful (generally phrasal) segments of the input, such as an NP segment or a VP segment. They then dispatch the corresponding word sequences to transduction hierarchies appropriate for their phrasal types; these recursively derive and return ULF formula constituents, which

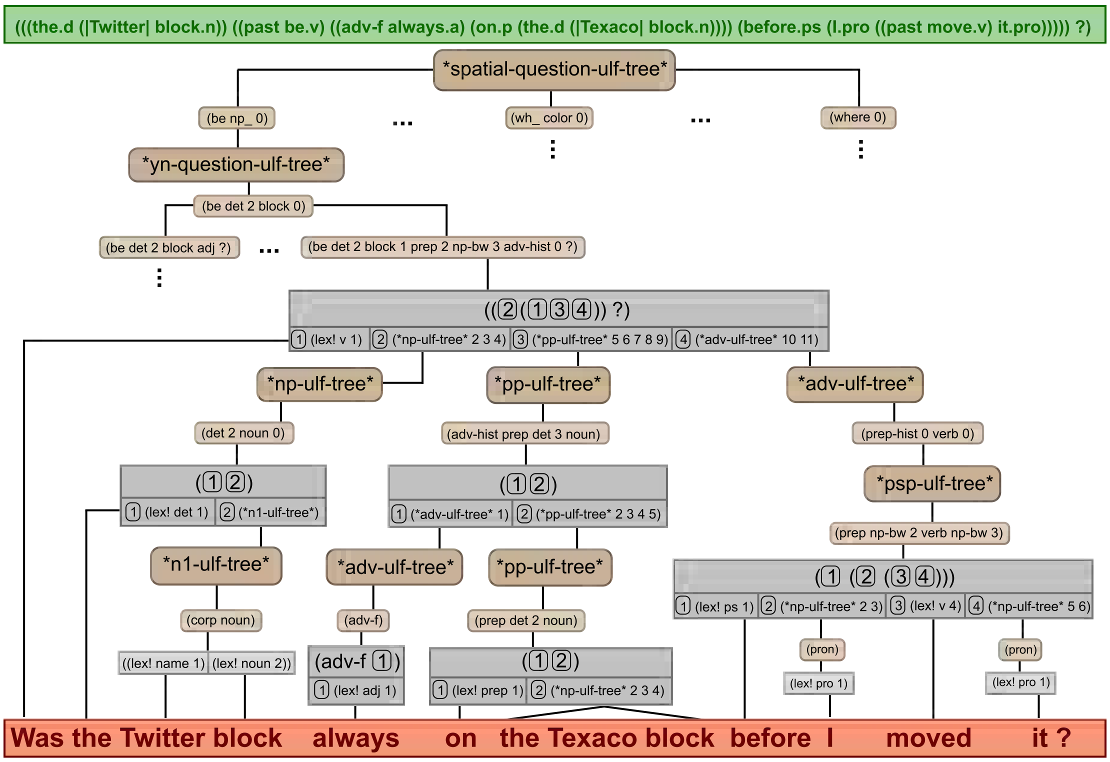

Figure 6.3: An example ULF parse, with the input shown in red, and the resulting ULF (at each composition step) shown in green. The nodes with rectangles represent ULF composition nodes, where the numbers in the upper box correspond to the indices of the lower boxes (if there is no upper box, the constituent ULFs are simply concatenated). All other nodes are patterns to be matched to the corresponding span of input text.

are then composed into larger expressions by the “calling” tree, and returned. At the level of individual words (or certain phrases), a lexicon and lexical routines supply word ULFs.

6.3.3 Reasoning

While much of the reasoning involved in the blocks world domain relates to quantitative spatial relations or temporal relations, and are therefore outsourced to the respective specialist servers, we also employ a bottom-up reasoning transducer to generate additional inferences from user inputs. One particular case is that of *implicature-generating* questions: for example, the historical question "What blocks did I move?" implies the question "What blocks did I *just* move?" (unless defeated, e.g., by the adverbial "...since the beginning"). These implicatures are inferred through a simple rule-based transducer.

6.3.4 Reaction

Given the classified intent of the user input, a rule-based reaction transducer is used to select an appropriate subschema for handling that input. For example, the schema in Figure 6.2b will be selected for reacting to a spatial question, whereas a schema for pausing the conversation will be selected for reacting to a special request to pause the conversation. In the case of smalltalk, such as the user inquiring about the name of the agent, DAVID may simply react with a direct `say-to.v` action, whose content is computed in the usual way by context-aware pattern transductions.

6.3.5 Paraphrase

We use hierarchical pattern transduction to generate an output utterance given the generated answer to a user's query. In most cases, the paraphrased utterance will simply be identical to the generated answer. However, in cases where the answer is overly lengthy (e.g., a conjunction of a large number of relations) or technical (e.g., involving numerical distance between blocks), we convert the utterance into a more natural form. Furthermore, pronouns may be substituted for block names in some cases. For example, in the case where the user asks DAVID a question such as "*Why* is the Twitter block near the Texaco block?" (testing understanding of the latent factors underlying spatial relations (Platonov et al., 2021a)), the spatial system may generate a technical answer such as "The scaled raw distance between the Twitter block and Texaco block is 1.5"; this would be paraphrased into a more natural form such as "The distance between them, scaled by their relative sizes, is 1.5.”[2].

6.4 Registering Historical Context for Question-Answering

For DAVID to be able to answer questions about historical states of the world – such as *“Which blocks were near a red block in the beginning?”* – the agent must have some sort of episodic memory of past actions and world states, and must be able to use this episodic memory in order to “reconstruct” the spatial relations between blocks in prior states. In this section, we describe the creation of a specialist server that allows for efficient backtracking through an episodic memory of prior actions and the estimation of spatial relations at particular points in time.

A natural question that emerges is: which memories should be preserved, and how should these memories be represented? Since our modelling of spatial relations is based on 3-D Blender graphics representations of the objects in the blocks world, a straightforward approach would be to store successive states in this “imagistic” form. However, this would be intractable in terms of computation and storage in more general scenarios, where there may be object-rich scenes or indefinitely long histories.

Some prior work, such as Rothfuss et al. (2018) and Franklin et al. (2019), takes the approach of representing episodic memory in a sub-symbolic form compatible with deep learning models. The former uses an unsupervised encoder-decoder model to represent episodic memory as latent embeddings, and shows that this model can allow a robot to recall previous visual episodes in a physical scene. The latter introduces a neuro-symbolic Structured Event Memory (SEM) model that is capable of segmenting events in video data and reconstructing past memory items.

[2] Scaled distance, rather than absolute distance, is used here due to the fact that, in general, *nearness* depends on the relative sizes of objects as well as the distance between them (Platonov and Schubert, 2018).

We, however, opt to represent memories at a higher level of abstraction: Studies of human visual memory indicate that detailed visual memories of scenes are quite short-lived, and a few higher-level properties suffice for humans to swiftly reconstruct more detailed representations of a scene (Rensink, 2001). Therefore, we operate over an episodic memory containing symbolic knowledge about *changes* in the world – allowing us to seamlessly integrate spatial reasoning with the memory component of the Eta architecture described in Section 4.1.2. This symbolic history enables approximate reconstruction of past states of the world given a set of current spatial perceptions.

6.4.1 Tracking Temporal Relations

As a spatial question answering session progresses, the perceptual server records the centroid coordinates of blocks and block moves in real time. In the current system, these perceptions consist of block location propositions of the form `(|Twitter| at-loc.p ($ loc ?x ?y ?z))` (where "`$ loc`" indicates a location record structure), and block move propositions of the form `(|Twitter| ((past move.v) (from.p-arg ($ loc ?x1 ?y1 ?z1)) (to.p-arg ($ loc ?x2 ?y2 ?z2))))`[3].

We rely on a simple linear, discrete time representation (possible future modifications are discussed in Section 6.7). Eta stores a symbol denoting the current time, with `|Now0|` representing the time at which the dialogue is initialized. Each sequential action in the world causes Eta to "update" its time twice corresponding to the time during which the move is in-progress and the time at which the move has finished. That is, if the initial time is denoted by `|Now0|`, a block move would cause Eta to update its time to `|Now1|` (the in-progress time), and then to `|Now2|` once the move has finished. These temporal symbols are related to each other via propositions of the form `(|Now1| before.p |Now2|)` and `(|Now2| after.p |Now1|)` stored

[3]In principle our representation also allows named locations, e.g., `(|Twitter| at-loc.p |Loc1|)`, though this is not yet implemented.

in the context.[4] The fact `((|Twitter| ((past move.v) (from.p-arg ($ loc ?x1 ?y1 ?z1)) (to.p-arg ($ loc ?x2 ?y2 ?z2)))) ** |Now1|)` is stored in the dialogue context, where '**' is the episodic characterization operator described in Section 3.1. User utterance actions are similarly stored in the context.

Based on this context, the DM can efficiently reconstruct a scene at any past time by backtracking from currently observed block locations, as well as use these reconstructed scenes to evaluate spatial relationships between blocks in a "rough-and-ready" way, i.e., using approximate calculations of spatial relations based on block centroid coordinates, as opposed to the detailed spatial computations supported by the visual blocks world system.

6.4.2 Interpreting Historical Questions

Following a successful ULF parse of a historical question by the semantic parser, such as the example shown in Figure 6.3, historical modifiers in a ULF will be indicated by constituents of type "adv-e" (event adverbial, e.g., `(adv-e (during.p (the.d move.n))))`, "adv-f" (frequency adverbial, e.g., `(adv-f (three.a (plur time.n))))`, or "adv-s" (sentence adverbial, e.g., `(adv-s (after.ps (|Twitter| (past move.v)))))`.

The algorithm the temporal specialist server uses to answer historical questions is as follows: starting from the present time, the algorithm iterates over past times, reconstructing the scene at each one using stored knowledge about moves. At each time, the algorithm computes and stores a list of salient facts (i.e., propositions about spatial relations or actions which held at that time) depending on the subject, object, predicate, question category, and polarity of the query sentence. Furthermore, temporal constraints

[4]Record structures specifying current world time are also attributed to these symbols; these are used in forming answers to "when" questions.

are applied to filter these times (in the manner described below) to obtain a final list of times with relevant attached facts.

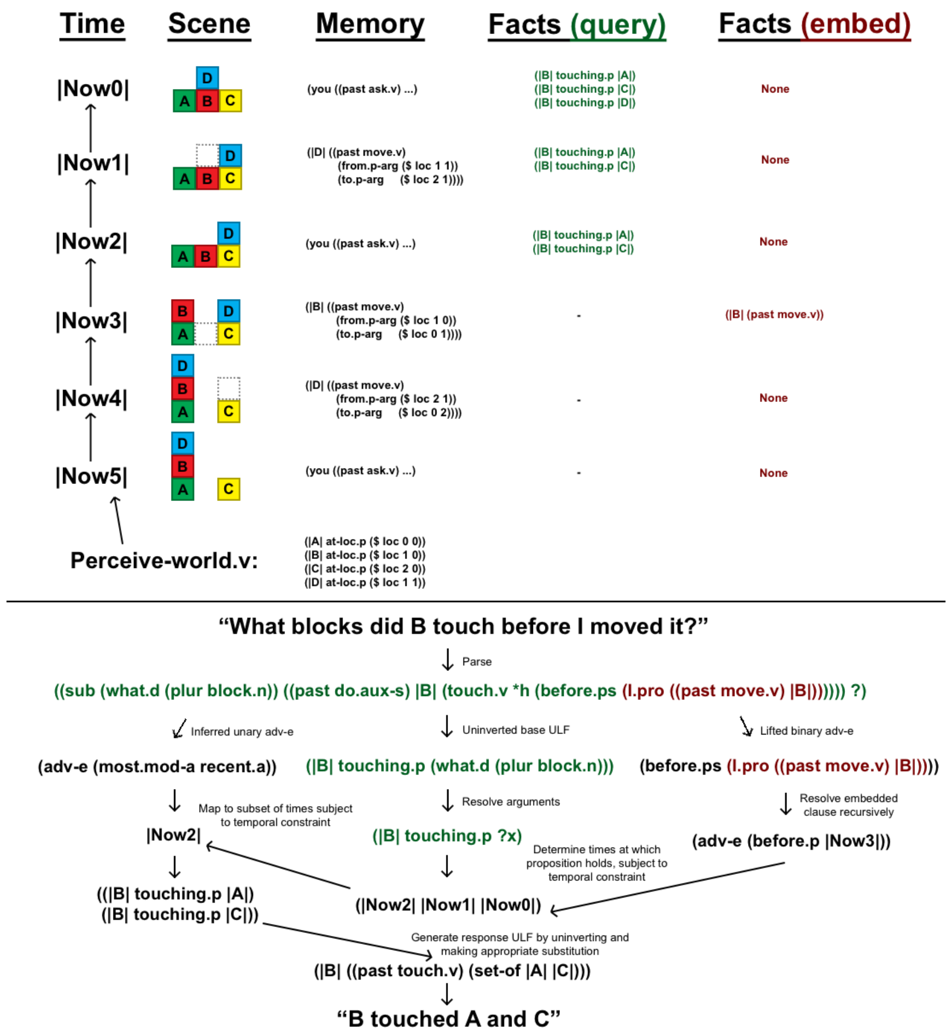

Figure 6.4: A simplified example showing how the temporal specialist uses episodic memory to compute relations given temporal constraints (top) and determine an answer from a specific historical query ULF (bottom).

The semantic types of these temporal and frequency modifiers allow them to be lifted to the sentence level (Kim and Schubert, 2017). Temporal constraints expressed by modifiers may be binary, e.g., `(adv-e (before.p |Now4|))`, or unary, e.g. `(adv-e recent.a)`. A binary constraint takes a temporal entity as an argument and maps it to a truth value, depending on whether the given relation holds with the object of the constraint. This is used by the algorithm described above to filter out each time at which the binary constraint does not evaluate to true. However, first the binary constraint needs to resolve its object constituent (which could be a simple temporal noun phrase or an embedded clause). This is done using a recursive call of the algorithm described above, which maps the ULF of the constituent to a list of times, treating any modifiers in the noun phrase or embedded clause as temporal constraints.

A unary constraint takes a set of times and maps it to a subset (possibly null) of these times. For example, the "recent" constraint above picks out the subset of times that are within some fixed threshold to the present time. Frequency constraints such as `twice.adv-f` or `(adv-f (three.a (plur time.n)))` are similar to unary constraints in that they take a set of times and return a subset of these, though their behavior is slightly more complicated – they pick out all times for which the salient facts attached to that time are also attached to at least N unique times, inclusive. For `(adv-f always.a)`, N is taken to be the size of the set of times being filtered (so that only facts that are attached to every time in the set are selected).

Each constraint may also be modified by a "mod-a" modifier, e.g., `(adv-e (just.mod-a recent.a))`, which modifies how that constraint is applied. In the case of "just recently", the singular most recent time is picked out.

Historical questions don't necessarily involve sentence-level adverbial modifiers, as the temporal content could be embedded within a noun phrase, as in *"What was the first block that I moved?"*. In this case, the DM will resolve this reference to a particular block by calling the above algorithm recursively, treating the noun pre- and post-modifiers as temporal constraints, and using the facts attached to the resulting times.

Once a list of final times and the corresponding facts/relations have been obtained, an answer is generated by making the appropriate substitutions in the query ULF (e.g. a wh-pronoun for the subject of a relation), applying syntactic transformations (e.g., uninverting questions and removing auxiliary verbs such as "do"), and converting this to surface form using the response transducer.

A full example of answering a historical question (using a simplified scene) is shown in Figure 6.4. The extraction of answer relations, given the query ULF and the generation of the answer ULF, is shown in the bottom half of the figure, while the scene reconstruction and computation of relevant facts/relations is depicted in the top half of the figure.

6.5 Evaluation

Two separate user evaluations of the DAVID agent in the QA task were conducted in Platonov et al. (2020) and Kane et al. (2020), respectively, with the former focusing on spatial questions, and the latter focusing on historical questions. These evaluations assessed the ability of DAVID to (a) deeply understand the user – quantified by the accuracy of DAVID's semantic parser – and (b) reason correctly within the domain – quantified by its QA accuracy.

Table 6.1: Spatial question results.

Total well-formed questions	329
Parser accuracy	74%
% Incorrect parses due to ASR errors	53%
Correct answers	66.6%
Partially correct	13.7%
Incorrect	18.8%

Table 6.2: Historical question results.

Total well-formed questions	387
Parser accuracy	94%
% Incorrect parses due to ASR errors	N/A
Correct answers	77%
Partially correct	3%
Incorrect	20%

For the spatial question evaluation, we recruited 5 volunteers from our department –

including both graduate and undergraduate students, as well as native and non-native English speakers – to test the capabilities of the system. Participants were instructed to ask spatial questions of the general type supported by the system, but without restriction on wording; before their first session they were shown a short demonstration of the expected kind of interaction with the system, including a few question-answer exchanges. During each evaluation session they were requested to ask between 40 and 50 questions and mark the system's answer as correct, partially correct or incorrect. Additionally, if no answer could be given because of speech recognition errors, they were asked to indicate that as well. Each session started with the blocks positioned in a row at the front of the table. The participants were instructed to move the blocks arbitrarily to test the robustness and consistency of the spatial models. The results from this session are shown in Table 6.1.

For the historical question evaluation, we recruited 4 volunteers from our department. Since the COVID-19 pandemic made testing the physical blocks world system on-site impossible, we instead conducted this evaluation using a virtual environment that mirrors our setup, sans the physical block tracker and the audio I/O. However, as the crucial components being evaluated (parser, dialogue manager, and spatial context) were unchanged, the significance of the results is not impacted. The participants were instructed to move the blocks around and ask general questions about changes in the world, with no restrictions on wording. After the system displayed its answer, the participants were asked to provide feedback on its quality by marking it as correct, partially correct or incorrect. Each participant contributed about 100 questions or above (primarily historical questions, but also including some non-historical spatial questions). These results are shown in Table 6.2. Non-historical questions, as well as a few malformed questions, were excluded when computing accuracy.

We additionally collect expert annotations (i.e., annotated by the authors) judging the correctness of the semantic parser outputs, and compute the parser accuracy based on the number of well-formed questions that result in semantically correct ULF expressions. In

```
DAVID: Would you like to ask me a spatial question?
User:  (moves a block)
DAVID: David, which block did I just move?
DAVID: You moved the Toyota block.
User:  Where was the Toyota block before?
DAVID: The Toyota block was between the Mercedes block
       and the Burger King block.
User:  Where is it now?
DAVID: The Toyota block is on top of the Texaco block.
User:  (moves another block)
User:  How many blocks have I moved since the beginning?
DAVID: You moved two blocks.
User:  Was the Twitter block always behind the Mercedes block?
DAVID: No.
User:  Did I move the Twitter block after I moved the Toyota block?
DAVID: Yes.
User:  When did I move the Toyota block?
DAVID: You moved the Toyota block three minutes ago.
...
```

Figure 6.5: Example of the sort of dialogue that DAVID can engage in (in the QA setting).

the case of the spatial question evaluation where the physical setup was used, the rate of incorrect parses that are attributed to automatic speech recognition errors is also shown.

Overall, the semantic parser itself is highly reliable, with 94% of grammatical sentences being parsed correctly in the text-based historical question evaluation, and 74% in the speech-based spatial question evaluation, with over half of the incorrect parses attributable to ASR errors. Moreover, DAVID provides the correct answer to a majority of user queries in both question categories, with a larger portion being judged as "partially correct" in the case of spatial questions. We provide a qualitative example of a transcript from the DAVID avatar in Figure 6.5 demonstrating both spatial and historical questions.

6.6 Extending to Concept-Tutoring Dialogues

One limitation of these initial experiments is that, however open-ended they were in terms of the questions that could be posed, they were limited to a relatively rigid and predictable dialogue flow, without much room for back-and-forth collaboration between the agent and user. While the QA task demonstrated strong spatial understanding and reasoning capabilities, a more collaborative task would test another important capability of a spatially-situated agent – namely, *spatial planning*. In this section, we consider an extension of the DAVID agent to support *concept tutoring* dialogues.

The concept tutoring task extends the QA task by enabling interactive collaboration between the DAVID agent and a user. In this task, DAVID has in mind some abstract concept (e.g., an arch – potentially given a made-up name to avoid the user relying on common knowledge), and attempts to interactively teach the user the concept through having them create examples of the concept with blocks. Typically, the user will first be guided by DAVID to construct a simple example block-by-block, and then tested on the concept by being asked to create a larger example (for example, a 5-block arch rather than a 3-block arch) in a more hands-off "supervised" manner, with DAVID only issuing corrections or answering questions.

To support concept tutoring, it is necessary to extend the spatial specialist server shown in Figure 6.1 with the ability to send actions/plan steps to DAVID, when queried – enabled through the integration of a symbolic spatial planner that maintains a hash table of on-relations and tracks possible steps towards a goal structure. We additionally create several schemas for guiding or supervising the construction of concepts within the blocks world. Figure 6.7a shows part of a schema for guiding the construction of a concept `?goal-rep` (e.g., an arch), which has a corresponding object schema containing knowledge about the part structure of the concept, shown in Figure 6.6. DAVID will repeatedly attempt to find an action that is a valid step towards the goal concept – instantiated with a particular action by querying the specialist server – and

```
(obj-schema :header (?x BW-arch.n)
;````````````````````````````````````````````````````````````````
[...]

:types (
  !t1 (?stack1 BW-stack.n)
  !t2 (?stack2 BW-stack.n)
  !t3 (?top BW-block.n))

:rigid-conds (
  !r1 (?top on.p ?stack1)
  !r2 (?top on.p ?stack2)
  !r3 (?stack1 next-to.p ?stack2)
  !r4 (not (?stack1 touching.p ?stack2))
  !r5 (?top clear.a)
  !r6 ((height-of.f ?stack1) = (height-of.f ?stack2))))
```

Figure 6.6: An example of an object schema for an "arch" in the blocks world domain.

then guide the user to perform that action; this continues until the planner does not suggest any further action, indicating that the goal state has been reached.

Part of the schema for guiding the user to perform an action is shown in Figure 6.7b. DAVID will first *propose* an action to the user, followed by an expectation that the user might reply with either an acknowledgement or a clarifying question – however, since this is only a weak expectation and the user may in fact remain silent, this step in the schema is associated with a relatively low certainty. The user is then expected to *try* to perform the proposed action (note that, if the user performs the action incorrectly, the next iteration of finding a step towards the goal state will generate an appropriate correction). Analogous schemas to those in Figures 6.7a and 6.7b were created for supervising construction in a hands-off manner.

The transduction methods in the concept tutoring task remain the same, but with augmentation of the pattern transduction trees used by the paraphrase transducer to support verbalization of proposed actions, and augmentation of the gist clause and semantic transducers to support additional types of user inputs.

```
(dial-schema :header
  ((^me guide-construction.v ^you
    ?goal-rep) ** ?e)
;````````````````````````````````````````
:types (
  !t1 (^you person.n)
  !t3 (?bb (plur block.n))
  !t4 (?goal-rep obj-schema.n))

[...]

:episodes (
  ?s1 (:repeat-until (?s1 finished.a)

    ?s2 (^me find.v (some.d
      ?ka1 (step-toward.p ?goal-rep)))

    ?s3 (:if (?ka1 = (ka do-nothing.v))
      ?s4 (^me say-to.v ^you '(Looks like
                   the structure's done))
      ?s5 (^me commit-to-memory.v
        (that (?e2 finished.a)))

    :else
      ?s6 (^me guide-action.v
           ^you ?ka1)))))
```

(a)

```
(dial-schema :header (((set-of ^me ^you)
    guide-action.v) ** ?e)
;````````````````````````````````````````
:episodes (
  ?e1 (^me propose-to.v ^you ?ka1)
  ?e2 (^you reply-to.v ?e1)
  ?e3 (^you try.v ?ka1))

:certainties (
  !c1 (!e2 0.4)
))
```

(b)

Figure 6.7: The dialogue schema used for guiding the user through the construction of a concept in the concept tutoring task (a), and for guiding a particular action (b).

While a quantitative user evaluation of the concept tutoring task was not conducted, a qualitative example of a dialogue from this task is shown in Figure 6.8.

```
DAVID: I would like to teach you the concept of an arch.
User: Okay, let's go ahead.
DAVID: Let's start with a simple example. Put a block on the table.
User: Sure. (places block A centrally)
DAVID: Now put a block to the right of block A.
User: (places block B touching the right side of block A)
DAVID: Well, block A should not touch block B.
User: Okay. (shifts block B half a block to the right)
DAVID: Good. Now place a block on top of block A and block B.
User: (places block C on top of both blocks)
DAVID: Excellent. You have now built the simplest possible arch.
DAVID: Do you think you understand the concept?
User: Can we try making a bigger one?
DAVID: We certainly can. Why don't you try building it?
User: (creates two 2-tall columns with blocks A-D)
User: How's that?
DAVID: Good, that's correct so far.
User: Should block E be on top of block C and block D?
DAVID: Yes.
User: Okay. (places block E on top of block C and block D)
DAVID: Looks like the structure is completed.
DAVID: I think you've got the idea of an arch. You caught on fast!
```

Figure 6.8: Example dialogue that DAVID can hold from the concept tutoring task. The company block identifiers are replaced with letters for brevity.

6.7 Discussion

The results obtained in the QA evaluation demonstrate that the Eta framework – augmented with expressive ULF representations, specialist reasoning servers, and domain-specific transducers – provides a powerful medium for allowing a situated agent to understand complex spatial and temporal relations and to communicate about them with a user in natural language. Furthermore, hierarchical pattern transduction provides an accurate and efficient means of implementing domain-specific ULF transducers. In particular, we attribute the high accuracy of the ULF parser to the fact that hierarchical pattern transduction allows use of miscellaneous low-level cues – patterns of words (not necessarily adjacent), and their syntactic and semantic features – to robustly segment sequences into phrase types top-down, recursively parsing the constituent phrases and combining their ULFs compositionally. Furthermore, the gist clause preprocessing strategy can trim or add words as may be needed for unimpeded interpretation.

While we did not conduct a quantitative evaluation in the concept tutoring task, it is still notable that the Eta framework allowed for direct transfer from a constrained QA task to a far more complex, interactive tutoring task through the creation of a small set of dialogue schemas and the integration of an external spatial planner. Due to the modular and portable nature of schemas, the tutoring agent also retained its abilities to answer a wide variety of spatial questions.

Although the blocks world domain is highly constrained, the spatial specialist methods used by Eta are fully general and have been demonstrated in a simulated "room world" containing everyday objects (Platonov et al., 2021b). It would be natural to extend the DAVID agent to this more realistic domain. However, the domain-specific techniques used by the DAVID agent – particularly the low-level patterns of words and features that are cues to overall phrase structure in semantic interpretation – are a limiting factor. Hierarchical pattern-matching-based methods would require substantial additional engineering for broader domains, making them difficult to scale.

Open-domain structured semantic parsing presents a challenging problem, but previous work has demonstrated some success by using neural cache transition parsers (Gildea et al., 2018; Kim et al., 2021) or Transformer-based encoder-decoder networks (Kitaev and Klein, 2018; Andreas et al., 2020; Platanios et al., 2021; Gibson and Lawley, 2022). By replacing or augmenting DAVID's semantic transducer with a neural semantic parser trained on diverse linguistic corpora, it may be possible to extend the QA dialogue task to fully general domains. This remains an interesting direction of future work.

7 SOPHIE Virtual Standardized Patient

7.1 Domain

Conversational virtual standardized patients (VSPs) – virtual humans that simulate patient interactions for use in training or evaluating medical practitioners – present an impactful but challenging application of dialogue technology. Communication skills on the part of the physician are a well-recognized determinant of patient satisfaction (Korsch and Negrete, 1972), and prior research has shown that poor communication by doctors leads to lower quality health outcomes at a higher cost (Ha and Longnecker, 2010; Riedl and Schüßler, 2017; Stewart, 1995; Begum, 2014; Butow and Hoque, 2020). Unfortunately, low cost communication training videos or reading materials have been shown to have little effect (Arnold et al., 1994; Ijaz et al., 2017). Training courses using standardized patients (SPs) are a viable remedy widely used in medical schools (Fiscella et al., 2007; Teherani et al., 2008). For example, our institution offers interdisciplinary workshops for practicing patient care professionals (e.g., physicians, nurses, advanced practice providers, social workers, and chaplains) through the Advanced Communication Training (ACT) program (Carroll et al., 2021), which teaches the MVP (Medical situation, Values, Plan) paradigm and emphasizes the 3 E skills: Empower, be Explicit, Empathize skills (Horowitz et al., 2020). Receiving feedback has been found to improve the communication skills of clinicians. For example, feedback from communication coaching experts based on recorded interactions with real patients has been shown to improve a clinician's ability to empathize with their patient and empower them by eliciting questions (Pollak et al., 2023). However, due to the cost and limited availability of human SPs and coaches who can provide relevant feedback, these traditional approaches are hard to scale; this is compounded by the diminishing effects of communication training over the course of a physician's career (DiMatteo, 1998). A system that allows a physician to practice with a VSP at the convenience of their personal computer, and receive direct automated feedback, could have a sizable impact on medical training.

The utility of automated feedback metrics for behavior such as empathy, however, depend critically on the quality of dialogue supported by the system – a VSP must provide the physician with *opportunities* to be empathetic (or to empower the patient, etc.), by generating natural reactions that are consistent with a real patient in their setting, express appropriate emotions, and demonstrate understanding of the physician's responses. Effective dialogue management in this domain is challenging because the types of conversations that oncologists have with real cancer patients are emotionally and topically diverse, and often involve complex mixed-initiative dialogue – i.e., a mixture of the patient taking initiative and the physician taking initiative at various times. These types of dialogues have radically different structure than task-oriented dialogues, requiring a greater degree of collaboration between the two agents (Walker and Whittaker, 1990). Furthermore, these dialogues can span a relatively wide range of topics, and can jump suddenly between topics.

Owing to present limitations in dialogue management technology, current VSPs have been largely limited to narrow single-initiative domains – for instance, allowing a doctor to practice eliciting a medical history from a patient and creating a differential diagnosis, where the interaction is enabled using a combination of statistical retrieval and pattern-matching methods (Maicher et al., 2017; Carnell et al., 2015; Talbot et al., 2012; Rossen et al., 2009). Other more general techniques, such as fine-tuning large language models

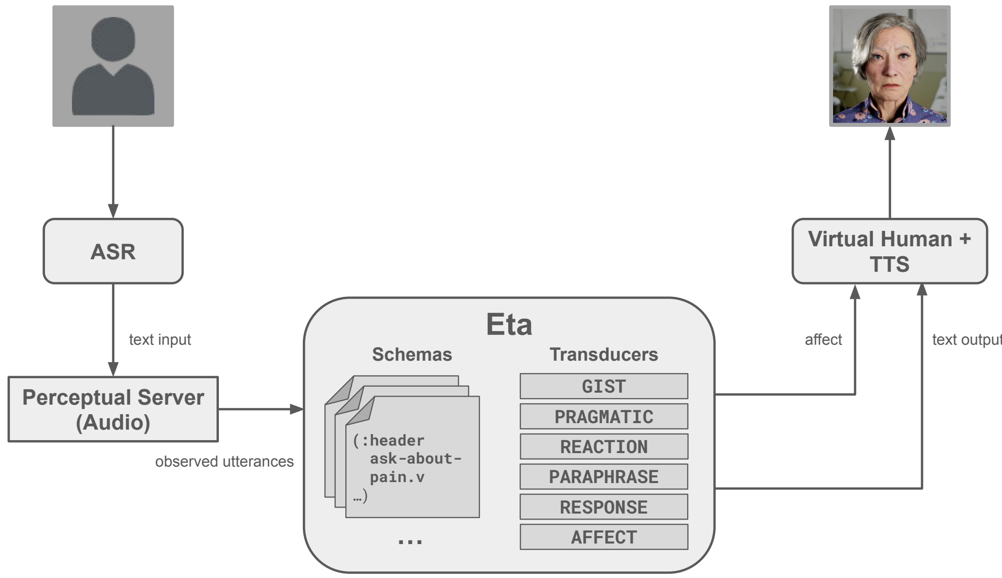

Figure 7.1: The full agent configuration used for the SOPHIE domain.

on domain corpora, have been successfully employed for creating robust conversational chatbots for entertainment purposes (Adiwardana et al., 2020; Roller et al., 2020; Zhang et al., 2020a). Yet, these methods present limitations for developing realistic VSPs such as a lack of goal-directed planning abilities and a tendency to "hallucinate" false or contradictory information (Roller et al., 2020).

In this chapter, we present the creation of a VSP – SOPHIE – that plays the role of a virtual patient who has been recently diagnosed with lung cancer and is seeking medical advice about her condition, prognosis, and treatment options from a user. The full SOPHIE configuration is shown in Figure 7.1.

7.2 Schema Design

In designing schemas for the SOPHIE agent, it was important to balance three primary desiderata: First, the the agent must be able to handle mixed-initiative dialogue often present in doctor-patient interactions – i.e., a mixture of the patient taking initiative and the doctor taking initiative. Second, in order for a VSP to provide opportunities for the user to practice particular skills – e.g., to respond empathetically, to empower the patient, etc. – the agent must be capable of generating sufficiently natural, affective responses. Third, the VSP must be able to steer the conversation to ensure that its own conversational goals (e.g., learning about possible medical options) are met over the course of a conversation. These desiderata are often intertwined in complex ways; for instance, a failure of the user to display appropriate empathy may naturally lead to the VSP becoming less cooperative in a dialogue, providing a form of feedback to the user. The design of the Eta framework allows us to capture such interactions by inferring the use of particular skills by a user and designing schemas that condition SOPHIE's reactions on the presence or absence of these skills.

We developed four dialogue sessions with the SOPHIE agent in collaboration with palliative care experts. The first three sessions involve short medical scenarios that each target a particular skill in the "Medical Situation, Values, and Plan" (MVP) model of patient communication (Horowitz et al., 2020). A session in which SOPHIE talks about her recent pain allows the user to practice responding empathetically; a session in which SOPHIE asks about her prognosis allows the user to practice being explicit; a session in which SOPHIE asks about her treatment options allows the user to practice empowering SOPHIE to explore her values and goals. The top-level schema design for each of these sessions consists of an initial question by SOPHIE followed by multiple levels of escalation if the user fails to employ the appropriate skill, with the conversation terminating for feedback after 3 failed attempts. If the user successfully employs the skill, SOPHIE responds with a positive acknowledgement and pauses the conversation for feedback. An example of a schema for the empathy-focused session is shown in Figure 7.2. Note that the use of conditional episodes allows for the possibility of multiple different dialogue flows depending on the use of skills by the user, and the presence of particular social obligations (e.g., that talking about pain obligates empathy) allows for later checking the successful use of skills by the user (e.g., for providing feedback).

```
(dial-schema :header ((^me ask-about-pain.v ^you) ** ?e)
;````````````````````````````````````````````````````````````````````````````````````
:episodes (
  ?e1 (^me paraphrase-to.v ^you '(Why has my pain been getting worse recently ?))
  ?e2 (^you reply-to.v ?e1)

  ?e3 (:if (not (^you be.v empathetic.a))
    ?e4 (^me react-mildly-to-non-empathy.v ^you)

    ?e5 (:if (not (^you be.v empathetic.a))
      ?e6 (^me react-moderately-to-non-empathy.v ^you)

      ?e7 (:if (not (^you be.v empathetic.a))
        ?e8 (^me paraphrase-to.v ^you '(I don't think I can handle this right now .
                                        I need a break .))
        ?e9 (^me say-to.v ^you '(Let's pause here for feedback on this conversation .))
        ?e10 (^me stop-conversation.v))))

  ?e11 (^me acknowledge-empathy.v ^you)
  ?e12 (^me say-to.v ^you '(Let's pause here for feedback on this conversation .)))

:obligations (
  !o1 (?e1 obligates (that (^you be.v empathetic.a)))
  [...]
))
```

Figure 7.2: A simplified portion of the dialogue schema used for the empathy-focused SOPHIE session, in which SOPHIE asks the user about the pain that she's recently been experiencing. Steps such as `react-mildly-to-non-empathy.v` correspond to subschemas to be instantiated.

A final session is aimed at mimicking a more natural doctor-patient conversation in which all three skills may be employed. The schema for this session progresses through each of the topics of the earlier sessions in sequence (however, because of the dynamic planning enabled by the Eta framework, this order may be modified – for instance, in case the user indirectly answers a later question by SOPHIE at an earlier point in the conversation).

In order to handle potential discursive conversations that might naturally happen in a doctor-patient conversation, we additionally create a set of subschemas for reacting to questions or statements by the user. For example, if the user mentions the possibility of chemotherapy, SOPHIE may select a subschema for inquiring about the side-effects of chemotherapy. If the user asks where SOPHIE's pain is located, SOPHIE may select a subschema for telling the user that her pain is located in her chest. Each subschema typically consists of a speech act by SOPHIE followed by an expected user response. We created 103 subschemas across a variety of medical topics (66 of which involve a statement by SOPHIE; 37 involve a question from SOPHIE).

7.3 Transduction Methods

The SOPHIE agent relies in part on hierarchical pattern transduction methods similar to those used in the LISSA agent, as described in Chapter 5. However, because of the greater complexity of doctor-patient conversations, we augmented the gist clause and reaction transducers with additional transduction methods, described in the following. These include methods for generating pragmatic inferences (particularly concerning the use of skills by the user), generating surface utterances from the expectations in a given schema, and producing an appropriate affect/emotion classification for a particular SOPHIE utterance.

7.3.1 Gist clauses

The user's input is split into individual clauses prior to gist clause extraction; the system attempts to obtain a gist clause for each input clause, which are then combined. The transduction tree for gist clause interpretation begins by selecting appropriate subtrees for interpretation based on keyword patterns in the prior SOPHIE utterance, as in the case of LISSA. For a given prior utterance, this tree will select both a specific topical subtree (e.g., if SOPHIE were to ask how much time she has left, a "prognosis input" subtree would be selected, possibly matching user responses like "You likely have three months.") as well as a general interpretive tree matching responses outside the detected topic (e.g., if the user were to reply "What's your biggest concern right now?"). This allows SOPHIE to accurately match expected answers to her questions while also matching unexpected or topic-shifting responses.

We designed 37 specific topical subtrees, each containing about an average of 28 top-level patterns. Each topical subtree took approximately 15-30 minutes to develop, depending on the level of detail required. The general interpretive tree in turn attempts to match an input using a subtree for statements or a subtree for questions, depending on the syntactic structure and punctuation of the input. These subtrees contain 265 and 442 top-level patterns, respectively.

7.3.2 Pragmatic meanings

We use a simple rule-based pragmatic transducer to infer the "pragmatic" meaning of a given gist clause. This is primarily used to classify the skills present in a user's input – i.e., whether they were empathetic, explicit, or empowering – but also allows SOPHIE to infer whether the user directly or indirectly provided her with relevant information about a topic. For example, if SOPHIE extracts the gist clause "your cancer has spread .", the pragmatic transducer will infer that (a) the user was explicit, and (b) the user told SOPHIE some information about her condition.

The pragmatic transducer consists of about 200 pattern-matching rules that were iteratively developed using a subset of the VOICE corpus of doctor-patient conversations (Hoerger et al., 2013) annotated for skills, as well as consultation with palliative care experts.

7.3.3 Reactions

Hierarchical transduction trees are used to select appropriate reactions to particular gist clauses, akin to the method employed for the LISSA agent. However, unlike in the case of LISSA, where the agent reacts with shallow speech acts, the reaction rules used by the SOPHIE agent primarily select *subschemas* that are instantiated and inserted into the dialogue plan. For example, if the user inquires about SOPHIE's mental health, SOPHIE may react with a subschema for discussing mental health. The advantage of this is twofold: First, since each subschema consists of a pair of expected dialogue turns, it allows for discursive, mixed-initiative dialogue flows, unlike the relatively constrained LISSA domain. Second, since the subschemas for reaction may involve SOPHIE paraphrasing some clause, it allows for conditional response generation based on the user's utterance.

7.3.4 Paraphrasing

In the case where a schema or subschema specifies a particular gist clause to paraphrase – such as `?e1` in Figure 7.2 – we use a paraphrase transducer to conditionally map the gist clause to a surface utterance based on the user's prior utterance. For example, if the user previously mentioned something about SOPHIE's cancer becoming worse, it may be more natural for SOPHIE to paraphrase the gist clause in `?e1` as "Do you think the cancer is why my pain has been worsening?" rather than the literal gist clause.

A top-level tree first selects an appropriate topical subtree on the basis of the *gist clause to paraphrase* – for instance, the gist clause in `?e1` may be used to select a subtree for generating responses related to pain. Each subtree contains multiple patterns matching the user's prior utterance, each containing several alternatives for SOPHIE outputs.

7.3.5 Responses

The response transducer is used in the case where a schema step simply specifies an arbitrary speech act – e.g., `(^me say-to.v ^you ?resp)` – rather than a particular gist clause to paraphrase. This method is used rarely in the SOPHIE agent for handling "small talk" or simple emotive responses (e.g., "Oh no..."); we use a single pattern transduction tree to select responses based on the user's prior utterance.

7.3.6 Affect

Finally, we classify each SOPHIE utterance as an appropriate affect or emotion label; these emotion labels are used to control the speech and facial expressions of the SOPHIE virtual human. The agent currently supports the following affect labels: neutral, sad, happy, worried, and angry. We use a rule-based affect transducer to associate a subset of possible SOPHIE outputs with predetermined emotion labels. By default, an utterance is associated with a neutral affect.

7.4 Pilot Experiment

In this section, we present a pilot experiment with the SOPHIE system in which we validate the system as a whole by evaluating whether participants who interact with the SOPHIE system are able to improve their communication skills. In addition to assessing the performance of the overall system, we also collect the conversation transcripts themselves, which we analyze further in Section 7.5.

7.4.1 Methods

Participants who are assigned to the intervention group with the SOPHIE system undergo three steps. First, they watch an instructional video about the MVP/3E's communication paradigm. Second, they view a tutorial video on how to use the SOPHIE system. Finally, they interact with the SOPHIE VSP for two sessions, including feedback after each conversation. The feedback page is split into a transcript, and a section for each of the three E's – including metrics such as hedge words and speaking rate (explicit), personal pronouns and empathetic words (empathy), and number of questions asked and turn-taking (empowering). Participants in the control group, on the other hand, received no training.

Both groups of participants interacted with one of four randomly assigned trained human standardized patient (SP) actors before and after, and were also rated by the SPs following each interaction. The SP rating scale was developed in close collaboration with URMC Oncologists and Palliative Care Specialists, and is based on communication behaviors that the human SP observes during their interaction with the participant. The full set of questions can be seen in Table 7.1.

7.4.2 Results

We summarize the results from the SP ratings in Figure 7.3, and present the full results in Table 7.1. We found that the intervention group performed significantly better on the "overall communicator" (intervention: 6.000, control: 5.067, $p < 0.05$) and "aggregate score" (intervention: 36.067, control: 29.600 $p < 0.05$) metrics. For every other question, there was a trend towards the intervention group, but the difference was not always statistically significant. A Bonferroni corrected pairwise t-test showed no significant differences between mean ratings given by the different SPs.

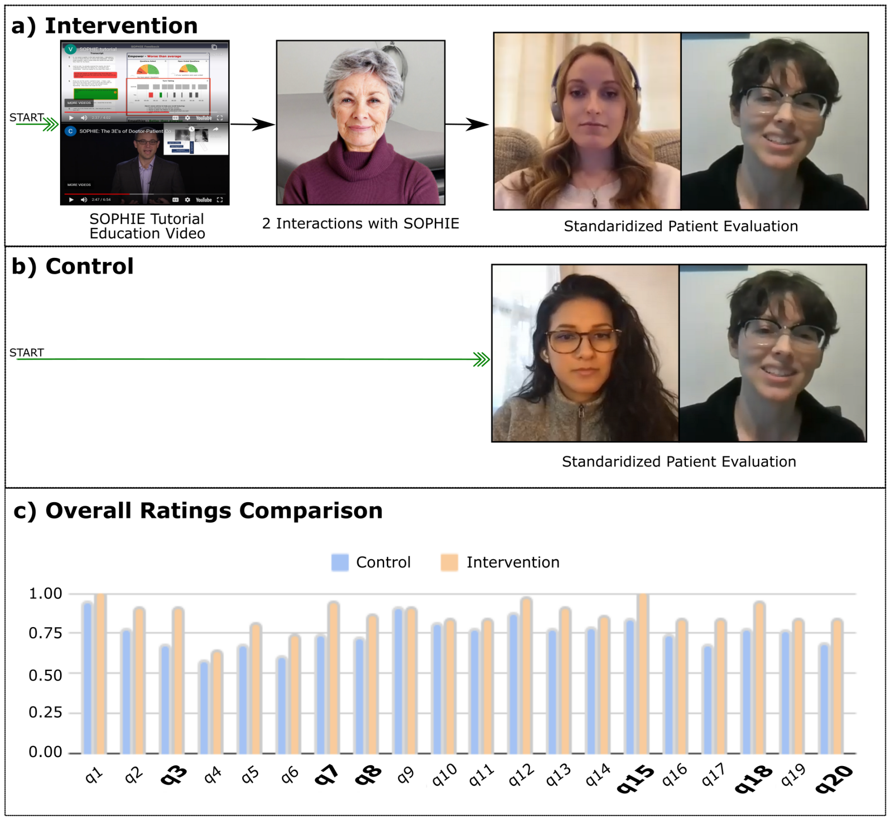

Figure 7.3: **Experiment with 30 participants** - a) Intervention group, underwent educational intervention with SOPHIE before speaking to SP. b) Control group, received no training before speaking to SP. c) Overall ratings comparison between control (blue) and intervention (tan) **bold** denotes significant differences. The numbers have been normalized to a 0-1 scale with 1 being "good." The raw numbers and full question text can be found in Table 7.1 by looking up the question ID. (Images of participants used with permission).

ID	Question	Control Mean	Intervention Mean	p-value
q1	The participant elicited the patient's major concerns within the first 5 minutes of the conversation.	0.867	1.0	0.175
q2	The participant asked for permission to share information about prognosis.	0.533	0.8	0.114
q3	*The participant asked how much information the patient would like concerning prognosis.*	*0.333*	*0.8*	*0.03**
q4	The participant checked the patient's prognostic understanding by asking them to state what they understood, using a teach-back approach.	0.133	0.267	0.282
q5	The participant actively encouraged the patient to ask questions using facilitating questions/statements.	0.333	0.6	0.078
q6	The participant helped the SP make a plan regarding with whom, and when, to convey prognostic information to family members.	0.2	0.467	0.123
q7	*The participant gave the SP many opportunities to talk.*	*0.467*	*0.867*	*0.024**

q8	*Empower Rating*	*5.267*	*6.133*	*0.003**
q9	The participant described the medical situation (the cancer has spread) clearly and without euphemism or jargon.	0.8	0.8	0.488
q10	The participant shared the prognosis accurately (a few months - less than one year).	0.6	0.667	0.476
q11	The participant used clear language without euphemism or jargon when sharing the prognosis.	0.533	0.667	0.252
q12	The participant used difficult to understand medical jargon.	-0.733	-0.933	0.079
q13	The participant lectured the patient (uninterrupted information for what seemed like a long time).	-0.533	-0.8	0.067
q14	be Explicit rating	5.667	6.067	0.084
q15	*The participant was generally empathetic.*	*0.667*	*1.0*	*0.04**
q16	The participant used states of empathy.	0.467	0.667	0.205
q17	The participant used silence appropriately in response to patient emotion.	0.333	0.667	0.051
q18	*The participant validated the SP emotional responses.*	*0.533*	*0.867*	*0.027**

q19	Empathize Rating	5.533	6.0	0.102
q20	*Overall Communicator*	*5.067*	*6.0*	*0.003**
–	**Total**	**29.6**	**36.067**	**0.005***

Table 7.1: Average SP ratings for each item in the rating scale. Italics and an asterisk ("*") denote $p \leq 0.05$

7.5 Conversation Transcript Evaluation

In this section, we turn to an evaluation of the quality of the response generation by the SOPHIE agent, using the transcripts collected in the pilot experiment. From the treatment group in the pilot experiment, we collected a dataset consisting of 397 conversation turns in total.

7.5.1 Expert Annotations

As a preliminary form of evaluation, two researchers involved in the design of the pattern transduction methods for SOPHIE independently annotated each conversation turn for (a) Whether the system extracted a correct gist clause, an incorrect gist clause, or failed to extract a gist clause; (b) Whether the system gave an appropriate response, inappropriate response, or a non-contentful clarification request; and (c) Whether there was any notable ASR errors that were observed, such as transcription errors and turn-taking errors where the user was cut off. These results are shown in Table 7.2, corrected for ASR errors by assuming that a clarification request by the system is correct behavior when the input contains a significant ASR error. Inter-annotator agreement (Cohen's kappa) was quite high for both annotations, at 0.85 for gist clause annotations and 0.71 for response annotations.

7.5.2 Neural Baseline Model

To allow comparison of the schema-based approach to a statistical approach to dialogue, we also establish a "neural baseline" for our conversation domain – that is, the performance of a large language model fine-tuned on human SP transcripts from our domain – and compare the responses generated by this model, when prompted with turns from the pilot dataset, to the responses from our system. Specifically, we used the DialoGPT-medium model Zhang et al. (2020a), as it was the largest language model we were able to train within our financial constraints at the time the study was conducted, and was a standard benchmark for a variety of dialogue tasks. We fine-tuned DialoGPT on the VOICE human SP dataset Hoerger et al. (2013), which contains 109,134 dialogue turns (44,917 of which are patient turns) across 389 dialogues between human actor cancer patients and doctors.

After filtering for only patient turns and creating a 20%/80% training set/validation set split, we trained the model for 5 epochs, using a context window size of 5 previous utterances, and a batch size of 1. This resulted in a validation set perplexity of 6.53; gains after 5 epochs were negligible. To generate model responses for the pilot data, for each turn in the dataset, we concatenated the user utterance with the context of the immediately preceding utterance (separated by end-of-turn tokens) and let the model generate the next response. We used a length penalty of 0.5 and a repetition penalty of 1.4 as generation parameters; these were found to generate the best responses through manual inspection.

7.5.3 Crowdsourced Evaluation

Given the SOPHIE responses and corresponding responses generated by the neural baseline model, we crowdsourced annotations on response quality using Amazon Mechanical Turk. We first removed 89 turns that were judged by either of the expert annotators to include significant ASR errors in the doctor's input. The remaining turns were used

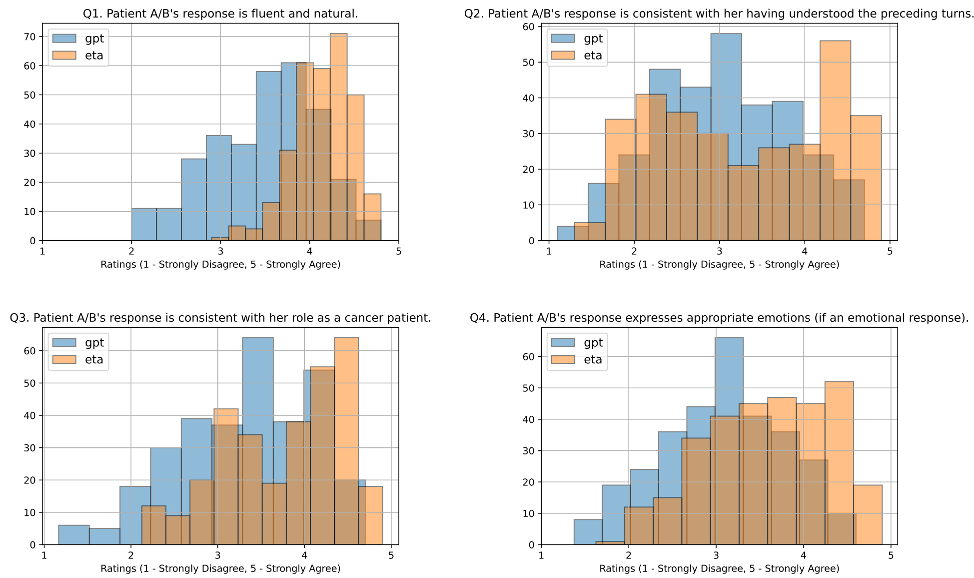

Figure 7.4: Crowdsourced evaluation results for SOPHIE from the pilot transcripts; distributions of average crowdsourced response ratings are shown for each question, compared with the LLM baseline.

to form 308 items, each consisting of the context of the previous patient utterance, the doctor's utterance, the response generated by Eta, and the response generated by the neural baseline model.

Items were randomly distributed into 20 Human Intelligence Tasks (HITs), each containing 16 items. To avoid introducing possible annotation bias, items were exactly balanced on string length of each text field (i.e., items were assigned to four high/low bins for each text field computed using a median split), and were approximately balanced on the expert annotations of response quality (ensuring that each HIT had about as many good responses as bad responses from each system). For each item, workers were shown a response from Patient A and the previous two dialogue turns, and then asked to rate the following four questions about the response on a Likert scale. Then workers were shown a response from Patient B to the same previous turns, and asked to rate the same four

ASR Errors	20%
Correct Gist Extracted	39%
No Gist Extracted	41%
Incorrect Gist Extracted	20%
Appropriate Response	49%
Clarification Request Response	28%
Inappropriate Response	24%
Appropriate Response, When Gist Clause Extracted	72 %

Table 7.2: Results from the expert annotations of the SOPHIE pilot transcripts.

Agent		Q1	Q2	Q3	Q4
GPT	M	3.49	3.03	3.30	3.06
	Mdn	3.55	3.00	3.40	3.11
Eta	M	4.15	3.29	3.78	3.60
	Mdn	4.20	3.20	4.00	3.60
Diff	M	+0.66	+0.26	+0.48	+0.54
	Mdn	+0.65	+0.20	+0.60	+0.49
	p	0.001	0.002	0.001	0.001

Table 7.3: Mean (M) and median (Mdn) of average crowdsourced responses across items for each set of SOPHIE responses, with p-values shown for mean differences.

questions. The Eta responses and neural baseline responses were randomly assigned to Patient A and Patient B.

- Q1. Patient A/B's response is fluent and natural.
- Q2. Patient A/B's response is consistent with her having understood the preceding turns.
- Q3. Patient A/B's response is consistent with her role as a cancer patient.
- Q4. Patient A/B's response expresses appropriate emotions (if emotional response).

Workers were instructed to focus on the quality of the response irrespective of the patient background and previous turns for answering Q1, and to overlook fluency in the response for answering Q2-Q4. Additionally, Q3-4 had "Not Applicable" options, in the event that these questions could not be evaluated (for instance, Q3 is impossible to evaluate if the patient asks *"Could you repeat that?"*, and Q4 is impossible to evaluate if

```
SOPHIE: I've already scanned the results, but don't understand what they are
        telling me. Could you explain what they mean?
User:   So unfortunately Sophie I have some bad news. It looks like the cancer has
        grown and spread.
SOPHIE: Oh no... those are not the words I wanted to hear. I mean, I was bracing for
        the worst, since I could tell by the pain that it's bad. But to hear that the
        cancer has spread is quite depressing. What does it all mean for me?
User:   How much information would you like to know about the prognosis?
SOPHIE: I feel very anxious about my condition. I feel like it's gotten worse. I want
        to know what this means for me.
User:   What concerns do you have about the future?
SOPHIE: What worries me the most is just how uncertain my future is. Some days I
        believe my condition will get better and others I brace myself for the worst.
        Is there any way you can let me know what I should expect?
...
```

Figure 7.5: Example of a dialogue between a medical student and SOPHIE from the pilot experiment.

the patient says *"I'm taking Lortab for the pain."*); any Not Applicable responses were discarded.

We obtained 10 ratings for questions Q1-4 for each item, from 167 native English speakers. The resulting distributions of mean ratings per item are shown as histograms in Figure 7.4. The mean and median scores for each question across all items are shown in Table 7.3; the mean differences between SOPHIE and DialoGPT were found to be statistically significant for each question ($P < 0.05$) using a Mann–Whitney U test. We provide a qualitative example of a dialogue excerpt from the SOPHIE pilot study in Figure 7.5.

7.5.4 Discussion

From the crowdsourced ratings summarized in Table 7.3, we observe that the SOPHIE avatar obtained high ratings for fluency/naturalness, consistency with the role of a patient, and appropriateness of emotions. Moreover, the avatar outperformed the fine-tuned DialoGPT model for each of these metrics, indicating that the customizability of Eta provides a distinct advantage over an end-to-end deep learning approach, since our framework allows the dialogue schemas and response transduction methods to be designed with a particular role and character in mind. The lowest mean rating for SOPHIE's responses pertained to whether the agent demonstrated understanding of the previous turns. Looking at the underlying rating distributions in Figure 7.4, the ratings for this question were strongly bimodal, indicating that SOPHIE would sometimes demonstrate high understanding, and at other times fail to understand the user – in comparison, DialoGPT tended to obtain consistently middling ratings for this question.

Turning to the expert annotations in Table 7.2, we observe that the rates of correct gist clause extraction and appropriate response generation were 39% and 49% respectively. Since SOPHIE employed strategies to ask non-contentful clarification requests in cases where it fails to extract a gist clause, the relatively low accuracy of gist clause extraction directly impaired SOPHIE's ability to generate an appropriate response – although in some cases it was able to select an appropriate default response. In the cases where the system was able to extract a gist clause, however, the fraction of appropriate responses was substantially higher – providing a direct explanation of the bimodality observed in Figure 7.4.

7.6 Extending SOPHIE with LLM-Aided Interpretation and Generation

These results from the pilot experiment and subsequent transcript analysis highlighted two significant limitations of the pattern transduction methods used by the SOPHIE agent in these experiments: First, even though the transduction trees covered a wide range of topics, the recall of this method was relatively low due to the large number of possible phrasings that were unaccounted for. Second, although extracting a gist clause allowed

SOPHIE to give highly appropriate and natural responses, the gist clause interpretation step was also acting as a bottleneck, preventing SOPHIE from generating an appropriate response in cases where a gist clause could not be obtained. The results, however, show that LLMs – while being less natural than Eta on average – are also more robust to cases where Eta fails to extract a gist clause, suggesting that the way forward may be a hybrid of the two approaches.

In this section, we discuss recent extensions of the SOPHIE agent to address these limitations through integration of LLMs. First, we enable robust interpretation of user inputs – particularly in the gist clause interpretation and pragmatic interpretation steps – by augmenting the existing pattern transduction trees with LLM-based interpretation methods. Second, we extend the paraphrase and response transduction methods used by SOPHIE with LLM-aided generation methods, allowing SOPHIE to give responses that demonstrate understanding even when it fails to extract a gist clause, while still retaining the advantages of the schema-based approach.

7.6.1 Robust Gist Clause and Pragmatic Interpretation

We augment the gist clause transduction method used by the SOPHIE agent with an LLM-based gist clause transducer, using the setup shown in Figure 4.4 (right). We use OpenAI's gpt-3.5-turbo model with the following initial prompt: "*I want you to rewrite the utterance sentences I give you in a maximally context-independent and explicit way, given a context sentence. Only generate a single sentence, and try to keep it as short as possible, without redundant information.*"[1]. This prompt is followed by 9 in-context examples, each consisting of a context, utterance, and rewritten gist clause. Finally, the LLM is provided with the utterance to be interpreted and the previous turn in the context. We use a simple gist validator that checks whether the generated gist clause contains any

[1]This specific prompt was designed through manual inspection of the resulting gist clauses from a small validation set.

clear contextual terms, e.g., pronouns. Any gist clauses obtained using this method are combined with those obtained through hierarchical pattern transduction.

In addition, we extend the pragmatic interpretation method with a data-driven model of medical conversation skills (i.e., being empathetic, being explicit, or empowering the user), allowing these to be inferred from the user input (hence, used to drive dialogue flow or to provide user feedback) even when the gist clause is incorrect. For this, we fine-tune a BERT (Devlin et al., 2018) multi-label classifier to predict skills given an input utterance. Three expert annotators independently annotated 3598 turns from the VOICE dataset (Hoerger et al., 2013) for skills, which were split to create a training set of 2878 turns and a validation set of 720 turns. The model was fine-tuned on the training set for 3 epochs, obtaining a final validation accuracy of 72% and an F1 score of 51%. Since false negatives were assessed to be more costly than false positives in this domain, we take the union of all skills predicted by this model and all skills inferred using the existing rule-based method.

7.6.2 LLM-Aided Response Generation

We next employ LLM-based transduction methods for paraphrasing and response generation in place of hierarchical transduction trees. We follow the same generation methods proposed by Kane and Schubert (2023) and discussed in Section 5.6; however, instead of retrieving knowledge from habitual schemas, we directly use schema knowledge associated with the currently selected dialogue schema. For example, as a reaction to the user providing a prognosis, SOPHIE may initially react by expressing mistrust in the user's prognosis – an example of a dialogue schema for this reaction is shown in Figure 7.6[2]. In addition to the expected episodes for each dialogue schema, we

[2]To ease development, and since strict goal/condition-based planning isn't required in this domain, we use only natural language eventualities (with indexical variables) for all schema sections but `episodes`. However, since Eta incorporates a ULF-to-English conversion algorithm, our method is also compatible with a more precise representation.

```
(dial-schema :header ((^me express-doubt-about-prognosis.v ^you) ** ?e)
;````````````````````````````````````````````````````````````````````````````````
:static-conds (
  ?s1 (^me has lung cancer .)
  ?s2 (^me has an uncle Fred who outlived his prognosis .))

:preconds (
  ?p1 (^me doesn't fully accept ^me 's prognosis .)
  ?p2 (^me doesn't trust ^you 's prognosis .))

:goals (
  ?g1 (^me wants to know more about ^me 's prognosis .)
  ?g2 (^me wants to know whether ^me can trust ^you 's prognosis .))

:episodes (
  ?e1 (^me paraphrase-to.v ^you '(Can I trust your prognosis ?))
  ;; ?e1 (^me say-to.v ^you ?response)

  ?e2 (^you reply-to.v ?e1)))
```

Figure 7.6: A portion of a dialogue schema in which SOPHIE expresses a lack of trust in the user's prognosis; for instance, because of conditions such as SOPHIE's relative having outlived their prognosis. Note that, as an alternative to the paraphrase episode `?e1`, the schema may directly specify a `say-to.v` step where the variable `?response` is bound to the response produced by an LLM.

add knowledge pertaining to the expected static conditions, preconditions, goals, etc. of each dialogue event. When paraphrasing the clause in step `?e1` (or generating an unconstrained response, as in the commented alternative), the LLM is conditioned on this schema knowledge. An example generation for this schema, for instance, may be "I'm having trouble trusting your prognosis. After all, I have an uncle Fred who outlived the prognosis he was given.".

One challenge that we occasionally encountered is the tendency of the LLM to hallucinate in its responses; in the most common failure mode, the LLM would role-switch and generate a response from the doctor's perspective rather than from the patient. We found this error to be less common when using OpenAI's gpt-4 model instead of gpt-3.5-turbo. Ultimately, we addressed this challenge by using gpt-4 to validate whether the gpt-3.5-turbo model contains a hallucination, and if so, retrying the generation using gpt-4 instead.

We additionally use gpt-3.5-turbo to classify SOPHIE's output with an appropriate affect/emotion tag. The model is provided the following prompt, along with a list of supported emotions, the utterance, and the last 3 turns of dialogue history: "From the following list, which emotional state most closely describes the feelings of ^me?".

Augmented with LLMs, the SOPHIE agent is able to generate natural responses even in cases where the agent is unable to correctly extract a gist clause from the user utterance; but importantly, the schema-based dialogue framework allows imposition of constraints on the LLM responses, ensuring that the responses are still on-track, consistent with SOPHIE's role, and emotionally appropriate.

7.7 Discussion

In the pilot experiment described in Section 7.4, we observed that, even despite the relatively limited capabilities of this version of SOPHIE, users were able to improve certain communication behaviors after several sessions of interacting with the SOPHIE agent and receiving feedback. We attribute this success to the fact that SOPHIE was able to display responses that were judged as generally natural, consistent with her role as a cancer patient, and emotionally appropriate, providing the user with an opportunity to rehearse particular skills even if the agent was not always viewed as truly understanding the user.

We used these insights in order to further construction of the SOPHIE agent, resulting in a far more intricate, comprehensive, and capable version of the system. Augmented with LLMs, the SOPHIE agent is able to generate natural responses even in cases where the agent is unable to correctly extract a gist clause from the user utterance; but importantly, the schema-based dialogue framework allows imposition of constraints on the LLM responses, ensuring that the responses are still on-track, consistent with SOPHIE's role, and emotionally appropriate.

Although it lies outside the scope of these experiments, another benefit of the schema-based design of SOPHIE is that it allows for explicit *personalization* of the virtual human through modification of the dialogue schemas. This is particularly important for VSPs since disparities in human SPs often lead to implicit biases in how medical practitioners interact with different demographics; allowing VSPs to be personalized allows for medical practitioners to practice with patients that have a variety of demographics or dispositions.

In the near future, we aim to conduct a larger scale and more rigorous experiment with the SOPHIE system in order to assess the performance of the augmented system. The preliminary results discussed in this chapter are promising, and further positive results could demonstrate SOPHIE to be a beneficial technology for medical communication training. To our knowledge, there is no other dialogue manager that has yet been developed capable of enabling realistic VSPs in open-ended and mixed-initiative patient-physician conversations, to the extent that we support in the SOPHIE system.

8 MegaIntensionality: Linking Prototypical Knowledge and Language

In the previous chapters we have focused on demonstrating the utility of a dialogue management framework based on prototypical knowledge, where we have assumed that prototypical knowledge could be explicitly represented in the form of schemas and matched to natural language utterances. In other words, in these systems, we began by assuming a fully specified model of prototypical dialogue events within a domain and *interpreting* natural language within the context of this model. However, it is possible to *invert* this relationship and instead ask the question: What prototypical knowledge is actually attested by the natural language used to communicate about events?

In this chapter, we turn our attention towards the MegaIntensionality project: an effort to answer this question by collecting lexical-scale data on prototypical judgments associated with English predicates and deriving a taxonomy of predicates according to this prototypical knowledge. We find not only that clusters of predicates appear to be associated with distinct patterns of prototypical knowledge, but also that the syntax of natural language itself is to some extent structured around these patterns.

8.1 Introduction

Some linguists have endeavored to study prototypical event knowledge at the level of individual lexical items, i.e., expectations associated with the events characterized by verbal predicates. Dowty (1991), for instance, postulates the existence of prototypical thematic roles – "proto-roles" – that play a part in determining argument selection (see also Reisinger et al., 2015; White et al., 2017). For example, an act of *running* typically involves an agent who performs the action with volition, whereas an act of *falling* typically has a patient who undergoes the change of state without volition. Similarly, it is hypothesized that clause-taking verbs[1] are associated with prototypical attitudes towards the embedded clause (Hooper, 1975; Heim, 1992; Anand and Hacquard, 2013, 2014). For example, for someone to *think* that some event happened typically involves a belief that that event happened, while for someone to *love* that some event happened typically involves both a desire and a belief that that event happened (as well as a *presupposition* that the event did, in fact, happen).

One way of capturing this prototypical knowledge is through evaluating which *inferences* a person may plausibly make in the presence of a particular lexical item, across a variety of syntactic contexts. Specifically, we say that an inference is *triggered* by a predicate if the use of that predicate consistently gives rise to that inference across contexts; for example, (1-a) $\rightsquigarrow$ (1-b). Furthermore, we may distinguish between types of inferences based upon whether the inference projects from negation (see e.g. Karttunen, 1973). If an inference is targeted or cancelled by negation, as in (1-b) vs. (2-b), we say that the inference is an *entailment*. If an inference is not targeted by negation, as in (1-c) vs. (2-c), we say that the inference is a *presupposition*.

(1) a. Jo remembered that he went to the grocery store.

b. $\rightsquigarrow$ Jo believed that he went to the grocery store.

[1] I.e., verbal predicates that are used in constructions involving some embedded phrase, such as finite clauses (e.g., *that something happened*) or nonfinite clauses (e.g., *to do something*).

c. ⇝ Jo went to the grocery store.

(2) a. Jo didn't remember that he went to the grocery store.

b. ⇝̸ Jo believed that he went to the grocery store.

c. ⇝ Jo went to the grocery store.

With these definitions in hand, we focus on the following subset of lexically triggered inferences: *veridicality inferences* (3-a), *neg(ation)-raising inferences* (3-b), *doxastic inferences* (3-c), *bouletic inferences* (3-d), and *intention inferences* (3-e).

(3) A predicate v triggers...

a. ...{*veridicality*, *antiveridicality*} *inferences* in sentence np (*not*) v ((to) np) s iff the use of np (*not*) v ((to) np) s consistently triggers the inference that {s, *not* s} across contexts.

e.g., Jo {*liked*, *lied*} that Bo left. ⇝ Bo {*left*, *didn't leave*}.

b. ...*neg(ation)-raising inferences* in a sentence np *not* v ((to) np) s iff the use of np *not* v ((to) np) s can trigger the inference that np v ((to) np) *not* s in some contexts.

e.g., Jo *didn't think* that Bo left. ⇝ Jo *thought* that Bo *didn't* leave.

c. ...{*doxastic*, *antidoxastic*} *inferences* about the role associated with position i in a sentence np$_1$ (*not*) v ((to) np$_2$) s iff the use of np$_1$ (*not*) v ((to) np$_2$) s consistently triggers the inference that {np$_i$ *believes* s, np$_i$ *believes not* s} across contexts.

e.g., Jo {*thought*, *doubted*} that Bo left. ⇝ Jo {*believed*, *didn't believe*} that Bo left.

d. ...{*bouletic*, *antibouletic*} *inferences* about the role associated with position i in a sentence np$_1$ (*not*) v ((to) np$_2$) s iff the use of np$_1$ (*not*) v ((to) np$_2$) s consistently triggers the inference that {np$_i$ *wants* s, np$_i$ *wants not* s} across contexts.

e.g., Jo {*loved*, *hated*} that Bo left. ⇝ Jo {*wanted*, *didn't want*} Bo to leave.

e. ...{*intention*, *anti-intention*} *inferences* about the role associated with position i in a sentence $\textsf{np}_1$ (*not*) v ((to) $\textsf{np}_2$) s iff the use of $\textsf{np}_1$ (*not*) v ((to) $\textsf{np}_2$) s consistently triggers the inference that {$\textsf{np}_i$ *intends* s, $\textsf{np}_i$ *intends not* s} across contexts.

e.g., Bo {*promised*, *refused*} to leave. ⇝ Jo {*intended*, *didn't intend*} to leave.

These inferences are of interest for at least two reasons. First, they display apparent correlations with each other across lexical items – potentially suggesting some core set of lexicosemantic components (i.e., semantic components that the meanings of each lexical item are built from) that interact to give rise to them. For instance, Anand and Hacquard (2014) suggest that, while there are predicates for which doxastic inferences are "foregrounded" entailments (as diagnosed by sensitivity to negation) – e.g. (4-a) ⇝ (6-a), (5-a) ⇝ (7-a) – and predicates for which bouletic inferences are entailments and doxastic inferences are "backgrounded" presuppositions (as diagnosed by *in*sensitivity to semantic operators like negation) – e.g. (4-b) ⇝ (6-a), (5-b) ⇝ (6-a), (4-b) ⇝ (6-b), (5-b) ⇝ (7-b) – there are no predicates for which doxastic inferences are "foregrounded" and bouletic inferences are "backgrounded" (see also Hooper, 1975; Heim, 1992; Anand and Hacquard, 2013). Such gaps in logically possible patterns of lexically triggered inferences have long played an important role in semantic theory because they suggest potentially deep constraints on lexicalization (Horn, 1972; Barwise and Cooper, 1981; Levin and Rappaport Hovav, 1991, a.o.).

(4) a. Jo knew that Bo left.
    b. Jo liked that Bo left.

(5) a. Jo didn't know that Bo left.
    b. Jo didn't like that Bo left.

(6) a. Jo believed that Bo left.
    b. Jo wanted Bo to have left.

(7) a. Jo didn't believe that Bo left.
    b. Jo didn't want Bo to have left.

Second, these inferences appear to correlate with the syntactic distribution of predicates – potentially suggesting that said lexical properties may be formally represented, rather than solely a byproduct of how conceptual representations interact with pragmatic reasoning. For example, veridicality and neg-raising inferences have been claimed to correlate with interrogative selection (Hintikka, 1975; Karttunen, 1977; Zuber, 1982; Berman, 1991; Ginzburg, 1995; Lahiri, 2002; Egré, 2008; George, 2011; Uegaki, 2015; Theiler et al., 2017, 2019; Elliott et al., 2017; Uegaki and Sudo, 2019; Roberts, 2019) (cf. (White, 2021)); and mood/finiteness selection (Giannakidou and Mari, 2021); and doxastic and bouletic inferences have been claimed to correlate with mood/finiteness selection (Bolinger, 1968; Hooper, 1975; Farkas, 1985; Portner, 1992; Giorgi and Pianesi, 1997; Giannakidou, 1997; Quer, 1998; Villalta, 2000, 2008). A long-standing question is thus whether there exists a *systematic* relationship between syntactic structures and prototypical inference patterns, or whether this relationship is always mediated by the lexical item.

A major remaining challenge in this domain is that, not only is the space of logically possible inference patterns vast, even the cleanest measurements of at least veridicality and neg-raising using inference judgment tasks display substantial gradience (White and Rawlins, 2018; An and White, 2020). This gradience makes it difficult to ascertain which inference patterns are attested because it makes it difficult to determine (i) whether a particular sentence should be considered to trigger a particular inference; and (ii) whether there exist patterns consisting partly or wholly of non-necessary inferences. Such difficulties may be unavoidable–e.g. because gradience indicates that no formally represented lexical property controls whether a particular inference is triggered (see Degen and Tonhauser, 2022). But there are at least two other (non-exclusive) possibilities: (i) apparent gradience is partly or wholly a product of the methods often used to collect inference judgments and that there are discrete, formally represented lexical properties that are active in triggering the necessary inferences; and/or (ii) that there are discrete, formally represented lexical properties active in triggering non-necessary inferences.

We argue that assessing these possibilities requires a lexicon-scale approach through which an inference pattern taxonomy can be derived.

Our primary aim in this chapter is to derive such a lexicon-scale taxonomy of English predicates based on (i) the doxastic, bouletic, intention, neg-raising, and veridicality inference patterns that they give rise to across a variety of syntactic contexts, and (ii) their associated syntactic distributions. To carry out this derivation while addressing the challenges posed by gradience, we apply a soft clustering model to each lexicon-scale dataset. Three of these datasets already existed prior to this work: one collecting syntactic acceptability judgments for various syntactic contexts (the MegaAcceptability dataset; White and Rawlins, 2020) one focused on veridicality inferences (the MegaVeridicality dataset; White and Rawlins, 2018) and one focused on neg-raising inferences (the MegaNegRaising dataset; An and White, 2020). We review these datasets in Section 8.2.

No similar, lexicon-scale dataset capturing doxastic, bouletic, and intention inferences – i.e., inferences pertaining to *intensional states* – existed prior to this work. To address this gap, we extended the methodology used to collect MegaVeridicality and MegaNegRaising to collect doxastic, bouletic, and intention inferences for 725 finite and nonfinite clause-taking predicates, covering a wide variety of semantic classes and resulting in the MegaIntensionality dataset. These include cognitive predicates–e.g. *think*, *know*, *remember*, *forget*–emotive predicates–e.g. *hope*, *fear*, *love*, *hate*–and communicative predicates–e.g. *say*, *tell*, *notify*, *convince*–among others. We describe the data collection process in Section 8.3.

Next, we turn towards the clustering model that we use to induce a taxonomy of predicates on the basis of these inference and syntactic acceptability judgments in Section 8.4. We label and discuss the clusters that we discover and the prototypical inference patterns that they capture. Finally, in Section 8.5, we discuss an exploratory analysis where we attempt to uncover the semantic components underlying these inference patterns – i.e., foundational semantic features from which the inference patterns of each cluster are constructed – and find a mapping of these components to syntactic features.

8.2 Existing Lexical-Scale Datasets

Our work on measuring prototypical intensional inferences builds on two previous lexicon-scale datasets capturing veridicality (the MegaVeridicality dataset; (White and Rawlins, 2018)) and neg-raising inferences (the MegaNegRaising dataset; (An and White, 2020)) across a variety of clause-embedding verbs in a variety of syntactic contexts. Both datasets include syntactic frames selected on the basis of their acceptability as measured in the MegaAcceptability dataset (White and Rawlins, 2016, 2020), which contains acceptability judgments for 1,000 English clause-embedding verbs in 50 syntactic contexts.

A major challenge to collecting such inference judgments at scale lies in disentangling the lexical effects of a sentence's matrix predicate on the inference from the potentially confounding effects of world knowledge. For instance, if a respondent were to indicate that (8-b) follows from (8-a), it would be difficult as an experimenter to tell whether their judgment was due to the semantics of *know* (as desired) or to the respondent's knowledge of world history.

(8) a. Jo knew that Napoleon was defeated at Waterloo.
  b. Napoleon was defeated at Waterloo.

To isolate predicate-specific effects on veridicality inferences, White and Rawlins (2018) build on a method they developed in (White and Rawlins, 2016) for constructing low-content sentences. Specifically, they solicit judgments by presenting participants with a *bleached* sentence, as in (9-a), and then ask (9-b), where the possible responses are *yes*, *no*, and *maybe or maybe not*.

(9) a. Someone {knew, didn't know} that a particular thing happened.
  b. Did that thing happen?

This bleaching is carried out by instantiating all noun phrases in a syntactic context with indefinite pronouns and by replacing all verbs except for the target verb with *do*, *have*, or *happen*, as appropriate. White and Rawlins (2018) capture factivity in this paradigm by manipulating the negation on the verb and seeing whether participants judge that the inferences goes through in both conditions. The resulting MegaVeridicality dataset contains veridicality judgments for 517 finite clause-embedding English verbs in both transitive (passivized) and intransitive contexts.

An and White (2020) take a similar approach to investigating neg-raising inferences. Participants are presented with questions like (10) and respond using a bounded slider ranging from 0 (*not likely at all*) to 1 (*very likely*).

(10) If I were to say *I don't think that a particular thing happened*, how likely is it that I mean *I think that that thing didn't happen*?

As in (White and Rawlins, 2018), the authors select predicates based on their acceptability in different frames and with different tenses based on data from MegaAcceptability. The resulting MegaNegRaising dataset contains judgments for 925 clause-embedding verbs in six syntactic contexts (including multiple involving infinitival complements) in both past and present tenses, and with both first and third person subjects—e.g. the past analogue of (10) with a third person subject is (11).

(11) If I were to say *a particular person didn't think that a particular thing happened*, how likely is it that I mean *that person thought that that thing didn't happen*?

8.3 Collecting a Dataset of Intensionality Inferences

In addition to the veridicality and neg-raising inferences already captured by MegaVeridicality and MegaNegRaising, we aim to capture patterns of doxastic, bouletic, and intention inferences at scale. As with veridicality and neg-raising inferences, a major obstacle to collecting inference data at scale is that, using standard item construction methods, it can be difficult to ensure that one is isolating prototypical inferences triggered by the predicate of interest (in some syntactic context) rather than surrounding lexical material (in conjunction with world knowledge). For instance, *boast* tends to trigger an inference that the boaster believes the content of the boast, but this inference is defeasible in cases where the boaster is a willful liar, as in (12).

(12) Trump boasted that he won in 2020.

(13) Trump doubts that he won in 2020.

Conversely, *doubt* tends not to trigger bouletic inferences; but if given a sentence like (13) and asked how likely it is that Trump wants to have won the 2020 election, one would likely answer that it is highly likely—mainly on the basis of prior knowledge.

To mitigate this issue, we deploy a *templatic semantic bleaching* method for data collection wherein we strip away lexical material that may confound the prototypical judgments associated with the predicate itself. In this method, participants are presented with *templatic items* consisting of a *templatic antecedent*, as in (14), and a *templatic consequent*, as in (15), and are asked to judge the likelihood that the consequent is true given the antecedent using a slider with *extremely unlikely* on the left and *extremely likely* on the right[2].

(14)
 a. A boasted to B that C happened.
 b. A doubted that C happened.

(15)
 a. A believed that C happened.
 b. A wanted C to have happened.

[2]In a separate validation experiment, we compared inferences collected using this method to inferences collected using a contentful method – controlling for world knowledge using a norming task that assessed the *a priori* likelihood that each contentful item is true – and found that the templatic bleaching task reliably captures the same information about lexically triggered inferences (Kane et al., 2021; Gantt et al., in prep).

We collected data for the MegaIntensionality dataset over two separate experiments – one focused on *finite* frames (those taking a sentential complement, i.e., A ___ that S), and one focused on *nonfinite* frames (those taking a verb phrase complement, i.e., A ___ to VP). In the former case, we collect only doxastic and bouletic inferences, since we expect to find interesting intention inferences only in the case of nonfinite complements. We detail the finite data collection first, and then detail the modifications to this method that we made for collecting nonfinite data.

8.3.1 Collecting Finite Frames

Materials

We select 725 unique predicates for use in this experiment based on their normalized acceptability score in the MegaAcceptability dataset.[3] We focus on the 12 frames found in Table 8.1, manipulating (i) embedded tense/modality; (ii) the presence of a direct object (DO) or *to*-PP; and (iii) whether the matrix clause is passivized or not (to naturally capture predicates that take expletive subjects and direct objects, such as *amaze*, *surprise*, etc.). We manipulate tense/modality of the embedded clause—past (16-a), future (16-b), and tenseless (16-c)—to ensure good coverage of bouletic predicates, like *hope*, and deontic predicates, like *demand*; and we manipulate DO/PP-taking to ensure good coverage of communicative predicates, like *say*.

(16)
a. A knew that C happened.
b. A hoped that C would happen.
c. A demanded that C happen.
d. A said to B that C happened.

In all of these frames, the matrix predicate is in the simple past (for the active frames) or past participial form (for the passive frames). For each frame except the embedded

[3]These normalized scores are described in (White and Rawlins, 2020) and are available at megaattitude.io.

tenseless ones, we select predicates with a normalized acceptability score in that frame of $\geq$0.2. For embedded tenseless frames, we set the threshold at 1.5. These thresholds were determined by manual inspection of the least acceptable items that would be included. The 0.2 threshold roughly corresponds to an average rating of approximately 4.5 on the original ordinal scale (1-7), and the 1.5 threshold corresponds to an average rating of approximately 6.[4] The number of predicates that lie above this threshold for each frame can be found in Table 8.1. We additionally manipulate the polarity of the matrix clause in templatic antecedents, as in (17).

(17) A {wanted, didn't want} C to have happened.

And we construct templatic consequents for each antecedent that are conditioned on the tense/modality of the embedded clause—(18) for antecedents with embedded past tense and (19) for antecedents with tenseless or future embedded clauses.

(18)
a. A believed that C happened.
b. A wanted C to have happened.

(19)
a. A believed that C would happen.
b. A wanted C to happen.

We construct two sets of templatic consequents for each antecedent with a DO/PP.

(20)
a. A/B believed that C happened.
b. A/B wanted C to have happened.

We sort the resulting items into lists of 32, aiming to constrain the construction of these lists such that the distribution over the expected responses for the items list has a mean of

[4]We use a distinct threshold for the embedded tenseless frames because we found that the acceptability scores are significantly noisier for them than in other frames, resulting in many unnatural tenseless items being included when the threshold is set to 0.2. We believe this noise may be due to some MegaAcceptability participants missing the subtle difference between the embedded simple past and tenseless items—namely, the presence or lack of an *-ed* suffix.

Embedded Past		Embedded Future		Embedded Tenseless	
A ___ that C happened	534	A ___ that C would happen	498	A ___ that C happen	50
A ___ to B that C happened	156	A ___ to B that C would happen	160	A ___ to B that C happen	7
A was ___ that C happened	238	A was ___ that C would happen	213	A was ___ that C happen	24
A ___ B that C happened	33	A ___ B that C would happen	31	A ___ B that C happen	1

Table 8.1: Counts of unique predicates for each frame in the finite experiment.

approximately 0.5 and has high variance. The idea here is to avoid introducing "warping" into the participants' use of the response scale due to the underlying distribution of inferences associated with items in a particular list. For instance, if the underlying distribution of inferences for items in a list were to result in a heavy bias toward extremely high likelihood responses, any responses that might otherwise be moderately low likelihood or middling likelihood responses might be assigned extremely low likelihood in comparison to the majority of the other inferences in the list.

An obvious difficulty in achieving this goal is that we do not have access to the underlying distribution of inferences. Rather, this is exactly what we aim to measure, and so we must use proxy measures that are plausibly correlated. We use four such measures: the normalized veridicality and neg-raising scores from MegaVeridicality and MegaNegRaising, respectively; the normalized acceptability judgments from MegaAcceptability; and the frequency counts for the predicate in a particular item's templatic antecedent given by SUBTLEX (Brysbaert and New, 2009).[5]

We perform PCA on these scores and then bin items based on their score on each component. This binning was done sequentially: we first derive two bins based on a median split of scores on the first component; then for each of those bins, we derive two bins based on a median split of scores on the second component; continuing similarly for the remaining two components. This procedure results in 16 equally-sized bins. We

[5]We use the normalized veridicality and neg-raising scores described in (White and Rawlins, 2018) and (An and White, 2020), respectively. In cases where no veridicality or neg-raising score exists for a particular item, we randomly impute it.

construct lists such that every combination of PCA bin and consequent verb appear exactly once in each list. To fill a list with items, we choose frames and antecedent verbs proportionally to their frequency in that respective PCA bin, also enforcing a hard constraint that each antecedent verb appears no more than once in a list. For transitive frames, the subject of the consequent is toggled each time an item with a DO or PP in the templatic antecedent is added to a list, ensuring that we also obtain a balance of subject and object targets among the DO/PP frames in each list.

Finally, we add four sanity check questions to each list to verify participant reliability. These items are constructed in pairs, with one item in the pair having a clear-cut 0 response and the other having a clear-cut 1 response. Each item in the pair uses the same verb in both the antecedent and the consequent, with one item having a negated antecedent (creating a contradiction) and the other having a positive antecedent (creating a tautology). All such items use the A ___ that C happened frame, and we only use predicates with a very high acceptability in this frame ($\geq$3).

Participants

We recruited 272 native American English speakers on Amazon Mechanical Turk. Participants were allowed to respond to at most 20 lists, and each list was rated by 10 unique participants.

Results

Figure 8.1 plots the normalized judgments for the doxastic and bouletic inferences (for both subject target and object target, when applicable), with select predicates labeled. To obtain these normalized judgments for each item, we use a mixed effects beta model-based normalization procedure.

We examine the top two subplots—those showing judgments for items with embedded past tense—first. The top left subplot shows judgments for doxastic inferences. We

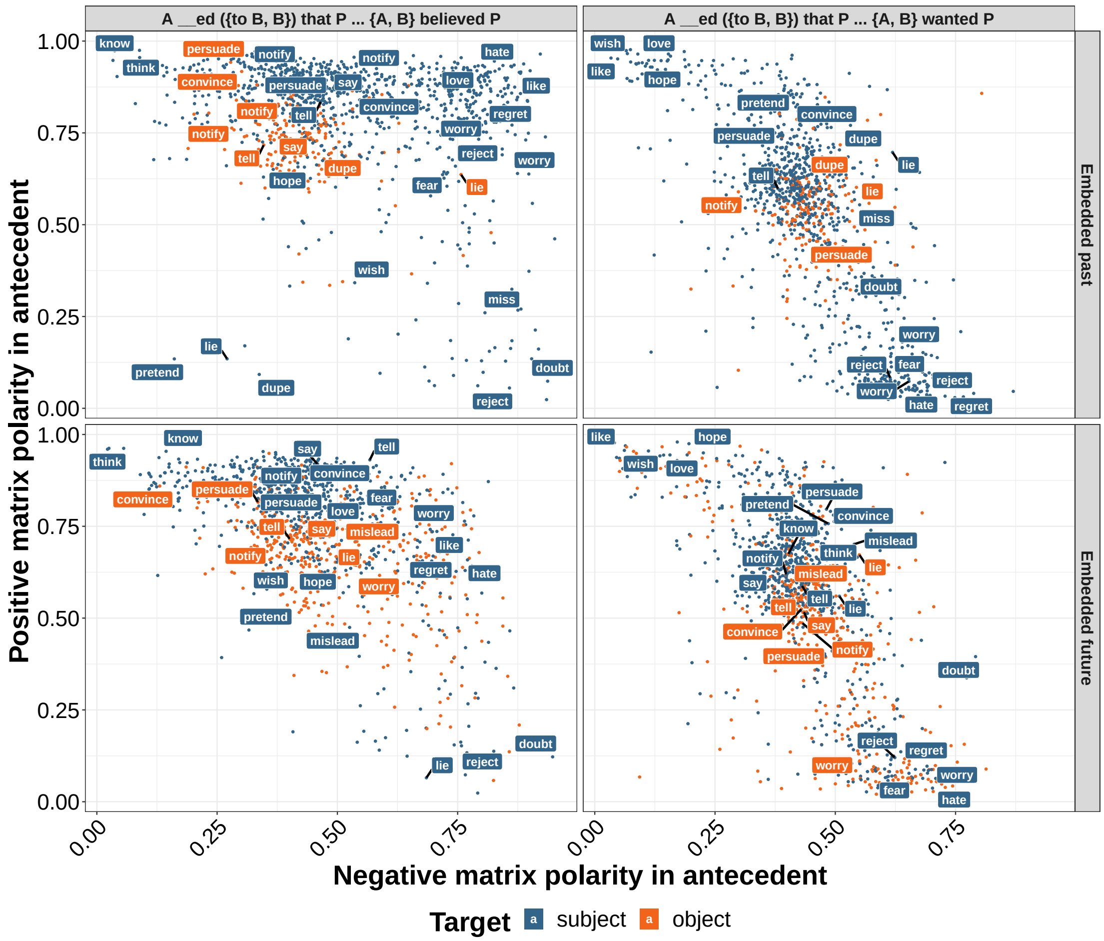

Figure 8.1: Distribution of verbs with respect to normalized doxastic and bouletic inference judgments, in finite syntactic contexts.

observe that our results correspond well with intuitions about a variety of commonly discussed predicates. For instance, cognitive predicates (such as *think* and *know*) and communicative predicates that trigger doxastic inferences about the recipient (such as *convince* and *persuade*) show up in the top left quadrant, indicating doxastic components that are "foregrounded" and targeted by negation. In contrast, emotive predicates such as *love*, *like*, and *hate* appear in the top right, indicating that the doxastic inferences are "backgrounded" and persist under negation. The center of the subplot shows predicates that don't trigger doxastic inferences, e.g. *wish* and *hope*. We also observe predicates that yield “backgrounded” negative doxastic inferences (deceitful communicatives, such as *lie* and *pretend*), and predicates that yield “foregrounded” negative doxastic inferences that are targeted by negation, such as *miss* and *doubt*.

The top right subplot shows judgments for bouletic inferences. As expected, we observe various positive emotive and preferential predicates such as *love*, *like*, *wish*, and *hope* in the top left quadrant, and negative emotive predicates such as *hate*, *regret*, *worry*, and *fear* in the bottom right quadrant. These indicate predicates that yield positive and negative bouletic inferences, respectively, that are “foregrounded” and targeted by negation. Many predicates are clustered around the center, indicating weak positive or negative bouletic inferences (such as *pretend* and *doubt*, respectively), or a lack of a bouletic component, as in the case of most cognitive predicates (e.g. *know* and *think*) and communicatives (e.g. *tell* and *say*). Notably, the overall pattern observed in this subplot suggests that no predicate has a positive bouletic inference when under both positive and negative matrix polarity. This pattern is consistent with the hypothesis that bouletic inferences (if present) are always at issue and targeted by negation (Anand and Hacquard, 2014).

The bottom two subplots show judgments for the same items with embedded future (when applicable). For the bouletic inferences for these items (bottom right subplot), the judgments we obtain are approximately the same as the corresponding items with embedded past tense. However, the doxastic inferences from these items (bottom left subplot) weaken significantly for any predicates where the doxastic inferences are backgrounded relative to the embedded past tense items—e.g. *love* yields a strong doxastic inference with a past tense embedded clause but only a weak doxastic inference with a future embedded clause.

8.3.2 Collecting Nonfinite Frames

Materials

We follow the same overall list construction process described in Section 8.3.1. We initially choose 738 unique predicates based on their normalized acceptability score in the MegaAcceptability dataset. We select 11 nonfinite frames, manipulating (i) stative versus eventive embedded clauses; (ii) the presence of a direct object (DO) or *to*-PP; (iii) passivization of the matrix clause; and (iv) for-PP, small clause, or gerund variants. These frames are shown in Table 8.2.

The acceptability of each predicate in each frame is taken directly from MegaAcceptability, except for the *to*-PP cases – A ____ TO B TO HAVE/DO C – which are missing from the dataset. We add these frames for predicates that appear in both A ____ TO B THAT C HAPPENED and A ____ TO HAVE/DO C frames, imputing the acceptability score as the minimum of either frame.

Stative		**Eventive**	
A ___ to have C	232	A ___ to do C	291
A ___ to B to have C	70	A ___ to B to do C	85
A was ___ to have C	348	A was ___ to do C	353
A ___ B to have C	188	A ___ B to do C	248
		A ___ for B to do C	266
		A ___ B do C	156
		A ___ doing C	327

Table 8.2: Counts of unique predicates for each frame in the nonfinite experiment.

For each frame, we select predicates with a normalized acceptability score of $\geq$0.8, resulting in the per-frame predicate counts shown in Table 8.2. We determined this threshold by manual inspection of the quality of the least acceptable items above this threshold. We manipulate the matrix polarity of the antecedent sentences, such as in example (21).

(21) A {prayed, didn't pray} for B to do C.

We create templatic consequent sentences pertaining to each of the three inference types, which were sorted into lists of 48 items each using the binning process described previously. In the case of transitive frames, we manipulate the target of the consequent sentence – that is, the subject of the consequent sentence is taken to be either the subject or object from the antecedent. Additionally, we add 4 sanity check questions to each list to allow validation of annotator responses. These questions consist of 2 pairs of items, where the consequent of the first item is identical to the antecedent – creating a tautology whose expected response is *extremely likely* – and the consequent of the second item is the negation of the antecedent – creating a contradiction whose expected response is *extremely unlikely*. Each sanity check item uses the A ___ TO DO C frame, and the predicates are randomly selected from those with very high normalized acceptability (≥ 3) in this frame.

In contrast with the finite frames, however, the experimental design for the nonfinite frames posed many additional challenges that required modifications to the collection method. We describe the challenges and corresponding modifications in the remainder of this section.

Filtering Small Clause Items Manual inspection of the items in the A ___ B DO C frame indicated that the acceptability scores for these items contained a high degree of noise, resulting in poor quality items in this frame above the base threshold. We hypothesize that, in the original MegaAcceptability study, some annotators interpreted these items as finite frames with an elided *that*, such as in (22).

(22) a. Someone advised someone do something.

b. Someone advised $_{\text{that}}$ someone do something.

We therefore manually select only 9 theoretically interesting predicates to include in the small clause frame: *feel*, *hear*, *imagine*, *make*, *observe*, *overhear*, *see*, *sense*, and *help*.

Selecting Eventive or Stative Frames A number of predicates are acceptable in both the eventive and stative versions of particular frames, which we expect would result in somewhat redundant annotations. Additionally, certain predicates have a preference for either an eventive or stative complement despite passing the base threshold for both, such as the examples in (23) and (24).

(23) a. Someone annoyed someone to do something.

b. ?Someone annoyed someone to have something.

(24) a. ?Someone haggled someone to do something.

b. Someone haggled someone to have something.

For each predicate that is acceptable in both the eventive and stative versions of a particular frame, we constrain it to appear in either the eventive or stative frame. We select the eventive version if the predicate has high acceptability in that frame (normalized acceptability ≥ 1), or the stative version otherwise.

Selecting Passivized, DO, for-PP, and to-PP Frames We further constrain the frames that each predicate appears in by selecting between direct object versus for-PP frames, passivized versus unpassivized frames, and to-PP versus intransitive frames. For each pair, we decide which frame to select based on their normalized acceptability scores. Specifically:

- If a predicate occurs in both A ____ FOR B TO DO C and A ____ B HAVE/DO C, we select the former frame only if the predicate does not have high acceptability in the latter frame, and has a higher acceptability in the former frame than the latter.

- If a predicate occurs in both A ___ B TO HAVE/DO C and A WAS ___ TO HAVE/DO C, we select the former frame only if the predicate has high acceptability in that frame; otherwise we select the latter frame.
- If a predicate occurs in both A ___ TO B TO DO C and A ___ TO DO C, we select the former frame only if the predicate has a higher acceptability in that frame; otherwise we select the latter frame.

A high acceptability is defined as having a normalized acceptability score ≥ 1.2, chosen by manual inspection of the least acceptable items above this threshold for each frame.

Manipulating Control Unlike in the case of finite frames, in the case of nonfinite frames we have to contend with control of unexpressed arguments in the embedded clause by the matrix predicate. For example, the embedded clause in (25-a) is under object control – the assumed argument of the *doing* is the object of the matrix clause, *B* – while the embedded clause in (25-b) is under subject control – the assumed argument of the *doing* is the subject of the matrix clause, *A*. The example in (25-c) allows for both subject and object control.

(25) a. A commanded B to do C.
b. A promised B to do C.
c. A helped B to do C.

It is therefore necessary to condition the constructed consequent sentences on the control of each predicate. To do this, two of the authors manually annotated all transitive frames for cases of subject control and subject+object control, assuming object control as the default[6].

We construct a pair of consequent sentences for each inference type – one with a subject target and one with an object target – for each possible argument of the embedded clause. For items annotated with object control, we use the consequents in (26), (28), and (30). For items annotated with subject control, we use the consequents in (27), (29), and (31). For items annotated with both, we use both pairs of consequents. In the case of intransitive items, we use only the consequents with both subject control and subject target.

(26) a. A believed that B did C.
b. B believed that B did C.

(27) a. A believed that A did C.
b. B believed that A did C.

(28) a. A wanted to do C.
b. B wanted A to do C.

(29) a. A wanted B to do C.
b. B wanted to do C.

(30) a. A intended to do C.
b. B intended for A to do C.

(31) a. A intended for B to do C.
b. B intended to do C.

Manipulating Embedded Tense Another challenge particular to nonfinite frames is that, while the MegaAcceptability dataset lacks the embedded tense markers present for the finite frames, certain nonfinite predicates have a preference for temporal orientation – i.e., whether the event of the embedded clause is located in the future, as in example (32), or the past, as in example (33).

(32) a. A promised to do C.
b. $\rightsquigarrow$ A intended to do C.
c. $\not\rightsquigarrow$ A intended to have done C.

(33) a. A admitted doing C.

[6] Each annotator independently annotated half of the transitive items.

b. $\not\rightsquigarrow$ A intended to do C.

c. $\rightsquigarrow$ A intended to have done C.

To select the appropriate temporal orientation to use for each consequent sentence, we use the MegaOrientation dataset, which contains judgments of the acceptability of various nonfinite frames with both past and future-oriented embedded clauses across 898 English verbs (Moon and White, 2020). For each item, we generate consequent sentences for whichever embedded tenses have a normalized MegaOrientation score ≥ 0.2 for that frame. If neither embedded tense is above that threshold, we select the embedded tense with the higher score.

Noise Reduction Manually inspecting the acceptability of a sample of the nonfinite items after the previous selection procedure, we observed that the nonfinite items were still far noisier than observed in the case of finite items[7]. Furthermore, many of the negative polarity items were ambiguous or otherwise difficult to answer.

To further filter difficult items out of our experiment, we randomly selected 10% of the candidate items – resulting in 24 lists of 48 items – and ran a separate experiment where annotators were asked to annotate both the likelihood *and* difficulty of each item. For example, after being shown an antecedent and asked “How likely is it that B intended to do C?”, a participant would then be asked to judge “How difficult is it to judge the likelihood that B intended to do C?” using a slider with *very easy* on the left and *very difficult* on the right.

After collecting these annotations, we trained a neural natural language inference model with mixed effects (Gantt et al., 2020) to separately predict the likelihood and difficulty scores of held-out items given the concatenated antecedent and consequent sentences as input. We then used the trained model to predict difficulty scores for the full

[7]While the exact cause of this is unclear, we hypothesize that it may be due to a greater degree of linguistic ambiguity in these inference judgments.

set of items, and removed items with predicted difficulty scores ≥ 0.375, determined by manual inspection[8].

We additionally modify the binning method during list creation to include the predicted difficulty and predicted likelihood scores in the PCA, ensuring that the items in the resulting lists are approximately balanced on difficulty and likelihood. We finally obtain 193 lists with 48 items per list.

Participants

For collecting difficulty annotations, we recruited 114 native American English speakers on Amazon Mechanical Turk. Each list was rated by 10 unique participants.

For the full experiment, we recruited 300 native American English speakers on Amazon Mechanical Turk. Participants were allowed to respond to at most 20 lists, and each list was rated by 10 unique participants.

Results

Figure 8.2 plots the normalized judgments for the doxastic, bouletic, and intention inferences for each embedded tense, as well as for each applicable combination of target and control conditions, with select predicates labeled. To obtain these normalized judgments for each item, we use a mixed effects beta model-based normalization procedure.

We first restrict ourselves to examining inferences targeting the subject, beginning with the leftmost two subplots, i.e., those showing doxastic judgments. We observe that commonly discussed cognitive predicates – including predicates that are future-oriented in infinitival contexts such as *know*, as well as predicates that are past-oriented such as *remember* – appear in the top left quadrant, suggesting doxastic inferences that are

[8]To ensure that each item still has an equal number of consequent verbs – as required for balancing the lists – we resample from the rejected items in order of increasing difficulty until the consequent verbs are exactly balanced.

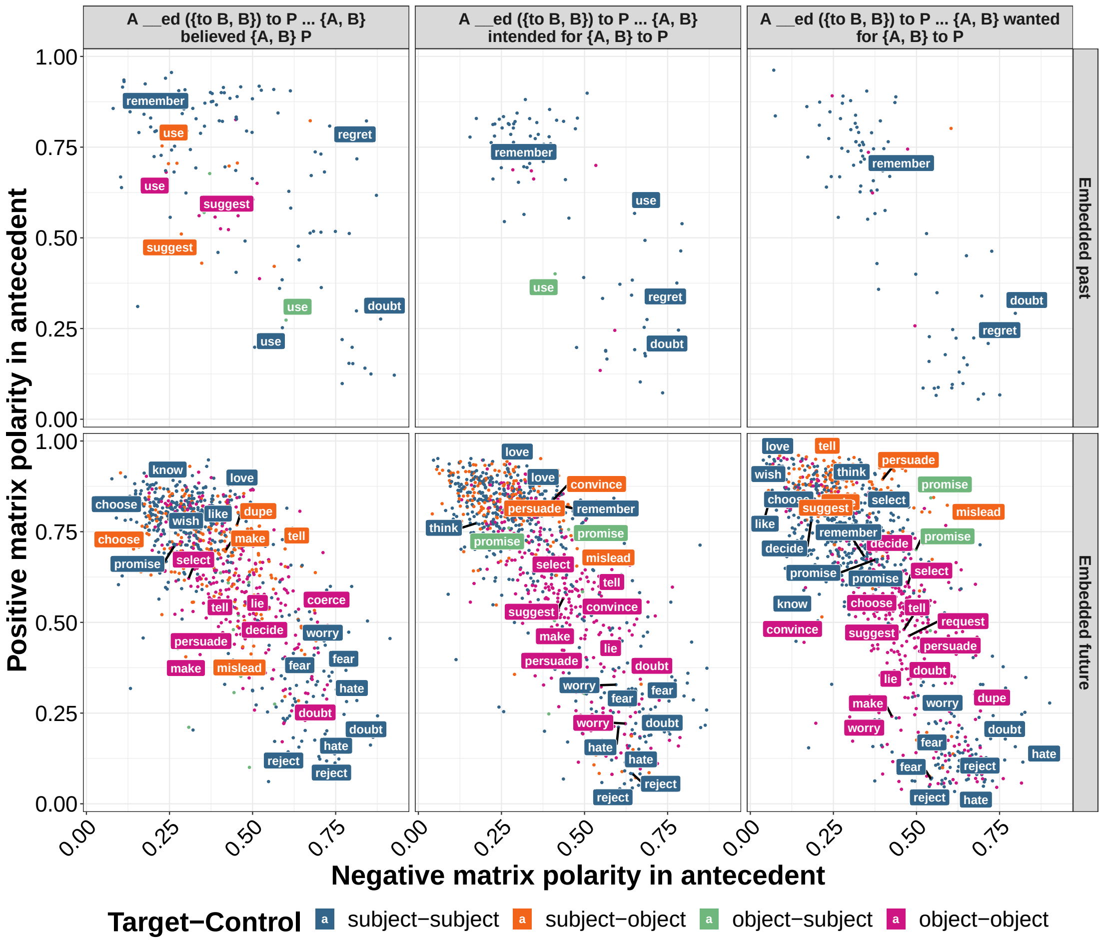

Figure 8.2: Distribution of verbs with respect to normalized doxastic, bouletic, and intention inference judgments, in nonfinite syntactic contexts.

targeted by negation, consistent with intuitions about such predicates. Past-oriented emotive predicates such as *regret*, appear in the top right, suggesting that doxastic inferences are presupposed by emotive predications pertaining to a prior event; in contrast, future-oriented emotive predicates such as *love* and *hate* appear in either the top left or bottom right, suggesting that these may trigger either a positive or negative doxastic inference respectively, and that these inferences are targeted by negation. Predicates with object control – e.g., communicative, authoritative, and persuasive predicates such as *tell*, *make*, and *choose* – yield weakly positive doxastic inferences about the object of the predication (excepting deceptives such as *mislead*) that may or may not be targeted by negation.

We next examine the rightmost two subplots showing bouletic judgments. As expected, we observe that both past-oriented and future-oriented emotive predicates trigger either positive bouletic inferences – such as in the case of *love* and *like* (as well as preferentials such as *wish*) – or negative bouletic inferences – such as in the case of *hate* or *fear*; furthermore, these inferences are targeted by negation. Cognitive predicates such as *remember* and *think* are associated with a moderate bouletic inference for the subject that does not project under negation; likewise, communicatives and other subject-control predicates such as *tell* and *persuade* are associated with moderate bouletic inferences for the object.

Finally, we turn to the central two subplots showing intention judgments. Intuitively, we expect that these judgments would pattern similarly to the bouletic judgments, but with predicates that implicate commitment to a particular event regardless of desire shifted in the positive direction. Indeed, we observe that predicates such as *remember* and *hate* are associated with somewhat higher judgments than in the case of bouletic inferences; conversely, predicates such as *doubt* and *fear* have weaker judgments.

We observe a surprising result when examining inference judgments instead targeting the object. While common predicates with subject control appear to conform to intuitions – such as the predicate *promise* triggering positive bouletic and intention inferences about the subject – we find that predicates with object control are heavily clustered around the center. Common persuasive predicates such as *persuade* and *convince* were judged to have very weak bouletic and intention inferences about the object of the persuasion, contrary to intuition. We manually selected a subset of these predicates and, in a small set of validation experiments, investigated them by collecting judgments with several alternative prompts (e.g., adding explicit “beforehand” or “afterward” modifiers). We found that, in the particular case of object target and object control, there appears to be high variability in the resulting judgments due to either true lexical ambiguity or inter-annotator meta-linguistic variability in how the items are interpreted. We leave this as an open question for future investigation.

8.4 Deriving a Taxonomy of Prototypical Inferences

Using these lexicon-scale datasets, we attempt to derive a taxonomy of English clause-embedding predicates according to distinct patterns in the prototypical inference judgments that they trigger. One sub-question that emerges in this analysis is: how many, and which, inference patterns and syntactic distributions are actually attested by the lexical-scale data?

To address this question, we fit a multiview mixture model simultaneously to each dataset – i.e., MegaAcceptability, MegaVeridicality, MegaNegRaising, and MegaIntensionality – treating each as a separate view. The result is a soft clustering of predicates according to common inference patterns, while allowing some morphing in order to ensure that the clusters are syntactically meaningful. Each cluster is associated with a central inference pattern (i.e., a distribution of scores in $[0, 1]$ across inference types and syntactic frames), as well as a central syntactic distribution. In some sense, this model allows us to determine the degree to which the full lexicon can be compressed on the basis of the observed inference and acceptability judgments.

8.4.1 Multiview Mixed-Effects Cluster Model

Model Specification

To obtain a soft clustering of the predicates in our datasets, we fit a *multiview mixed effects mixture model*. The underlying mixture model is similar to Latent Dirichlet Allocation for topic modeling (Blei et al., 2003): Each unique predicate v in the lexicon is associated with a categorical distribution over clusters with parameter $\boldsymbol{\theta}_v \sim \mathrm{Dirichlet}\left(\alpha\mathbf{1}_K\right)$

(where α is a dispersion hyperparameter); and the cluster assignment c_i for a particular datapoint i is sampled from the distribution corresponding to the predicate of that datapoint, $c_i \sim \text{Categorical}\left(\boldsymbol{\theta}_{\text{verb}(i)}\right)$.

Each dataset constitutes a different *view* of the predicates that we are clustering within the model. Following Gantt et al. (2020), we incorporate *mixed effects* into the model in order to account for variation in response behavior among participants. Specifically, each cluster c of the mixture model is associated with the fixed effect parameters $\boldsymbol{\beta}_{\langle\text{view},c\rangle}$ of the mixed effects response model for each view. The full model is as follows:

$$\begin{aligned}
\boldsymbol{\theta}_v &\sim \text{Dirichlet}\left(\alpha \mathbf{1}_K\right) \\
z_i &\sim \text{Categorical}\left(\boldsymbol{\theta}_{\text{verb}(i)}\right) \\
\boldsymbol{\beta}_{\langle\text{view}(i),\, z_i\rangle} &\sim \mathcal{N}(\mathbf{0}, 100\mathbf{I}) \\
\boldsymbol{\rho}_{\text{participant}(i)} &\sim \mathcal{N}(0, 100) \\
y_i &\sim f_{\text{view}(i)}\left(\boldsymbol{\beta}_{\langle\text{view}(i),\, z_i\rangle} \cdot \mathbf{x}_i, \boldsymbol{\rho}_{\text{participant}(i)}\right)
\end{aligned}$$

Unit Response Model The MegaNegRaising and MegaIntensionality datasets use a unit-valued response scale. Following Grove and White (under review), we use a unit mixed effects response model that assumes each response is a mixture of three truncated normal (TN) distributions: one centered around 1 representing "extremely likely" responses with some variance, another centered around 0 representing "extremely unlikely" responses, and a "prior" distribution centered around μ. $\beta^{(\boldsymbol{\theta})}_{c_i,k_i}$ and $\beta^{(\mu)}_{c_i,k_i}$ represent the fixed effects for the mixture weights and the location of the prior distribution, respectively, where c_i is the cluster assignment for an item and k_i is the inference type. $\rho^{(\boldsymbol{\theta})}_{p_i}$ and $\rho^{(\mu)}_{p_i}$ represent the respective random effects for participant p_i.

$$
\begin{aligned}
\boldsymbol{\theta}_i &= \text{softmax}\left(\beta^{(\boldsymbol{\theta})}_{c_i,k_i} + \rho^{(\boldsymbol{\theta})}_{p_i}\right) \\
\mu_i &= \text{logit}^{-1}\left(\beta^{(\mu)}_{c_i,k_i} + \rho^{(\mu)}_{p_i}\right) \\
y_i^{(1)} &\sim \text{TN}(1,1,0,1) \\
y_i^{(0)} &\sim \text{TN}(0,1,0,1) \\
y_i^{(\mu)} &\sim \text{TN}(\mu,1,0,1) \\
y_i &= \boldsymbol{\theta}_i \cdot \left[y_i^{(1)}, y_i^{(0)}, y_i^{(\mu)}\right]
\end{aligned}
$$

After fitting the model, we compute the mean inference values as $v_i = 1 * \theta_i^{(1)} + 0 * \theta^{(0)} + \mu * \theta_i^{(\mu)}$.

Ordinal Response Model The MegaVeridicality and MegaAcceptability datasets use a 3-point and 7-point ordinal response scale, respectively. For both datasets, we use an ordinal mixed effects response model that assumes that each item with a particular cluster c_i maps to some real-valued score β_{c_i}, and that each participant p has a different way of binning these scores into ordinal rankings. These bins are defined by random effects $\boldsymbol{\rho}_p$ that represent cutpoints, such that the worst rating corresponds to bin $(-\infty, \rho_{p1}]$, the best rating to bin $\left(\rho_{p(n-1)}, \infty\right)$, and all other ratings r to bins $\left(\rho_{p(r-1)}, \rho_{pr}\right)$.

The probability of a particular item i (with participant p_i and assigned cluster c_i) getting ordinal rating r is defined based on these cutpoints:

$$
\begin{aligned}
\mathbb{P}\left(y_i \leq r\right) &= \text{logit}^{-1}(c_{p_i,r} - \beta_{c_i,k_i}) \\
\mathbb{P}\left(y_i = r\right) = \theta_{ir} &= \mathbb{P}\left(y_i \leq r\right) - \mathbb{P}\left(y_i \leq (r-1)\right) \\
y_i &\sim \text{Categorical}(\boldsymbol{\theta}_i)
\end{aligned}
$$

After fitting the model, we compute the mean inference values using the expected distribution over the fixed ordinal cutpoints. We assign values to each ordinal rank such that the resulting expected values $v_i \in [0, 1]$[9], allowing them to be directly compared with the unit-valued inference model.

Model Fitting

We fit the model in PyTorch using maximum a posteriori estimation of the parameters, using the following loss function:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{\text{prior}} - \log P(\mathbf{y} \mid \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\rho}) \\ &= \mathcal{L}_{\text{prior}} - \sum_d \sum_i \operatorname*{logsumexp}_c \left[\log \theta_{\text{verb}(i),c} + \log f_d \left(y_i^{(d)} \mid \boldsymbol{\beta}_c^{(d)}, \boldsymbol{\rho}^{(d)}_{\text{participant}(i)}\right)\right]\end{aligned}$$

We minimize the loss function using minibatch stochastic gradient descent with the Adam optimizer with a learning rate of 0.02 and a batch size of 5%, for a maximum of 10000 epochs. We initialize the cluster weights and fixed effects parameters using K-Means with imputation of missing values. We use an early stopping criterion where, after 1000 epochs, we stop if the relative change in loss (over a window size of 100) is below a threshold of $1\mathrm{e}{-7}$. We set the cluster dispersion hyperparameter to be $\alpha = 1$.

Selecting the optimal number of clusters

A challenge that arises in our unsupervised clustering approach is that of selecting the optimal number of clusters. One way to select the number of clusters is to use an extrinsic metric; for instance, evaluating how well a particular number of inference patterns is able to *predict* syntactic acceptability judgments in a multivariate regression (Kane et al., 2021). In the present experiment, however, we incorporate the MegaAcceptability task into the model in order to find syntactically meaningful clusters, so we cannot use this dataset for an extrinsic evaluation.

[9] For example, the ordinals in MegaVeridicality would be assigned 0, 0.5, and 1.

Instead, we use cross-evaluation to select the optimal number of clusters. We split each dataset into 5 folds. For each held-out fold, we fit the model on the remaining datasets and compute the per-item log-likelihood of the held-out data. For each pairing of models (i.e., for each pairing of the number of clusters $|\mathcal{C}|$), we compute Bonferroni-corrected 95% confidence intervals for the likelihood ratio via non-parametric bootstrap over items. To choose the optimal number of clusters $|\mathcal{C}|^*$, we choose the smallest number of clusters such that no model with a greater number of clusters performs reliably better.

8.4.2 Results

After fitting models for each number of clusters up to $|\mathcal{C}| = 30$ using the cross-validation procedure described in Section 8.4.1, we find the optimal number of clusters to be $|\mathcal{C}|^* = 25$. This demonstrates that 25 clusters are sufficient to capture most of the variance in the observed inference patterns and syntactic distributions across the lexical-scale data. One important caveat, however, is that the inference data our model was trained on is not exhaustive; training the model on additional inference types may lead to a finer-grained clustering. In this section, we investigate the resulting clusters and the inference patterns that are associated with them.

Hierarchically Grouping Clusters

For expository purposes, we first hierarchically group the resulting clusters on the basis of correlations between their mean inference patterns and syntactic distributions. We perform an agglomerative hierarchical clustering in which we regard the "micro-clusters" from the mixture model as individual samples, and attempt to induce "macro-clusters" by iteratively merging clusters. We use a Pearson correlation distance metric, and define the distance between two clusters to be the maximum distance between any pair of data points within them (i.e., complete linkage).

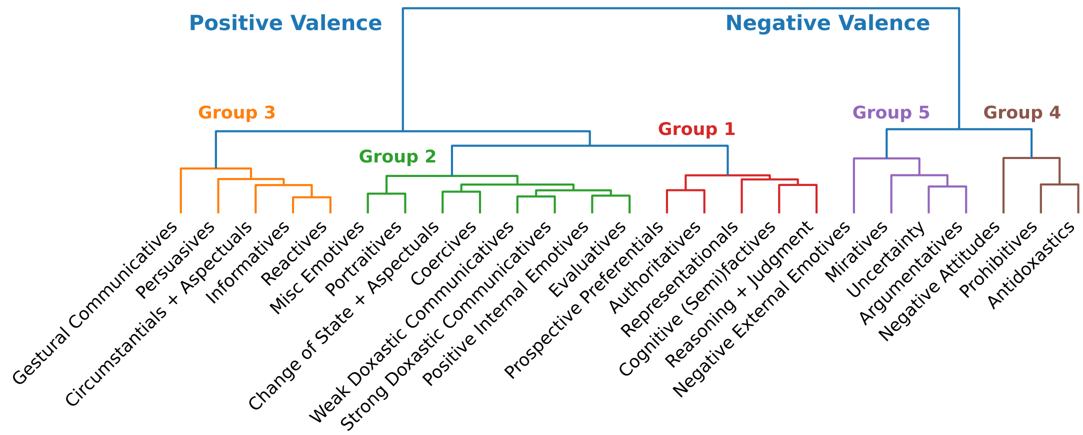

Figure 8.3: Through hierarchically clustering the "micro-clusters" from the mixture model based on correlations between inference patterns, we obtain several "macro-clusters" or groups, which in turn are divided between a group with broadly positive valence and a group with broadly negative valence.

The resulting dendrogram plot is shown in Figure 8.3. We manually label the most salient macro-clusters (i.e., groups), and we assign names to the micro-clusters on the basis of the highest-probability predicates contained within them. We additionally observe that the primary division – capturing most of the correlations between micro-clusters – corresponds to a broadly negative valence group of clusters (i.e., associated with various negated inference types) and a broadly positive valence group of clusters.

Clusters and Inference Patterns

In the following section, we provide a discussion of the salient inference patterns for each cluster in our taxonomy along with prototypical predicates in each cluster, comparing our results to prior literature when applicable. We investigate these inference patterns by analyzing the mean values of the fixed effects parameters learned by the mixed effects mixture model for each inference dataset.

Since we learn parameters for each inference type relative to each syntactic frame in a dataset, we first aggregate values across frames in order to aid interpretation. However,

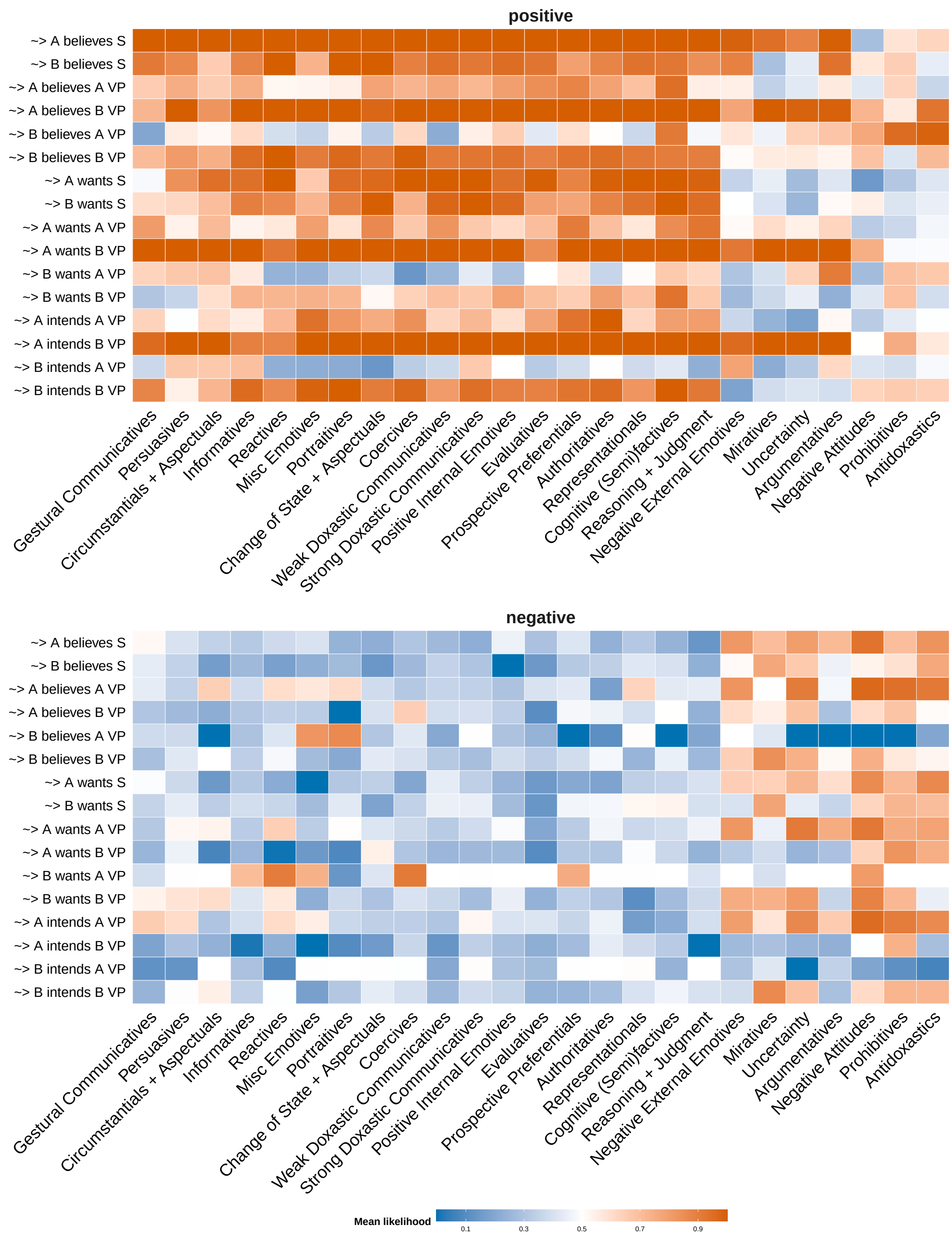

Figure 8.4: Prototypical doxastic, bouletic, and intention inference patterns for each cluster (weighted average across frames).

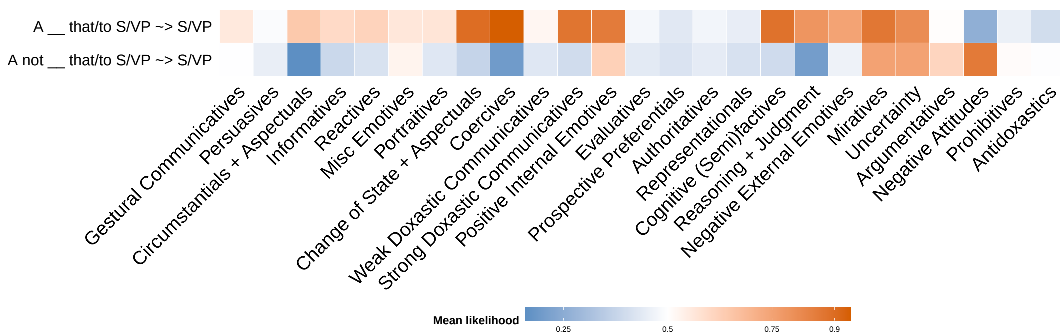

Figure 8.5: Prototypical veridicality inference patterns for each cluster (weighted average across frames).

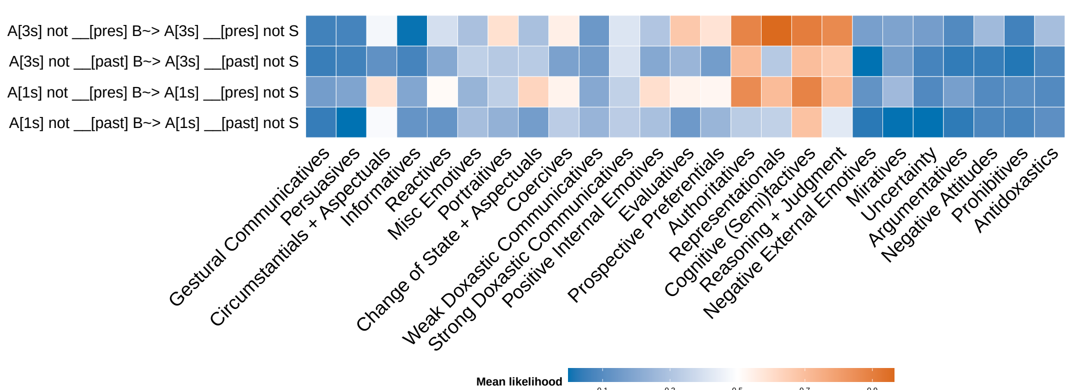

Figure 8.6: Prototypical neg-raising inference patterns for each cluster (weighted average across frames).

since not all inference type and frame combinations have sufficient datapoints available – due to low acceptability – some values learned by the model may not be meaningful. We account for this by taking a weighted average across frames for each inference type, weighting by the percentile rank of the acceptability values learned by the model for each frame. Figures 8.5–8.4 show the resulting inference patterns for veridicality inferences, negation-raising inferences, and doxastic, bouletic, and intention inferences, respectively. We primarily refer to these prototypical inference patterns throughout the following discussion of the observed clusters, except where we find that it is important to describe patterns relative to particular frames – in Appendix A.2 we provide plots showing the full disaggregated inference patterns. Furthermore, we provide full lists of the high-probability predicates within each cluster in Appendix A.1.

Group 1 broadly consists of clusters related to representation, reasoning, preferences, and authorization.

(34) **Reasoning + Judgment:** change in belief states through mental processes.
manage, glean, opt, presuppose, signify, conclude, etc.

(35) **Cognitive (Semi)factives:** cognitive states where truth is presupposed.
discover, realize, recognize, know, find out, understand, etc.

(36) **Representationals:** representations of belief states.
dream, hope, think, agree, assume, etc.

(37) **Authoritatives:** acts of authorizing some state or action.
expect, authorize, desire, request, allow, approve, etc.

(38) **Prospective Preferentials:** orientation towards bringing about some preferred future state.
aim, appear, compete, hunger, try, lust, attempt, start, etc.

We first compare and contrast the clusters of Representationals (36), Cognitive (Semi)factives (35), and Reasoning + Judgment (34). These clusters are all similar in that they primarily involve mental states and processes related to the representation of beliefs and knowledge – indeed, we observe from Figure 8.4 that these clusters are associated with similar inferences about the belief of the subject. However, they differ in whether the truth of their complement is in fact entailed or presupposed, i.e., whether they give rise to veridicality inferences in positive contexts, both positive and negative contexts, or neither. Predicates within the class of cognitive semifactives have been thought to weakly presuppose the truth of their clausal complements (Karttunen, 1971b; Kiparsky and Kiparsky, 1970); however, unlike true factives (such as in the case of emotive predicates, as we discuss later), these presuppositions are highly variable across contexts (Hooper and Thompson, 1973; Simons, 2007). Indeed, we observe from Figure 8.5 that this cluster is strongly veridical, but not systematically factive across all contexts – yet, we note that when we disaggregate inference types by individual frames (see Appendix A.2), this cluster is factive in the frame A ___ that S, consistent with prior literature.

The cluster of Reasoning + Judgment predicates are an example of what Karttunen (1971a) termed "two-way implicatives": they entail the truth of their complement in a positive context, and the falsity of their complement in a negative context. The Representational cluster, on the other hand, is nonveridical, consistent with previous findings (White and Rawlins, 2018).

The other two nonveridical clusters in this group – Authoritatives (37) and Prospective Preferentials (38) – primarily contain predicates that represent attitudes in relation to infinitival complements. The former – containing predicates that have to do with authorizing some action or state of affairs - tends to trigger inferences about either the intention of the subject or about the desire and intention of the object (in the case of transitive frames). The latter contains a subset of preferential predicates (Uegaki and Sudo, 2017) that involve preference for, and possibly an act to bring about, some future state or event – this cluster tends to trigger inferences that the subject believes, wants, and intends the embedded clause.

We note that this group as a whole is the only group that triggers strong neg-raising inferences (Figure 8.6) – with the exception of Prospective Preferentials, which trigger only weak neg-raising inferences in the present tense – suggesting that neg-raising is related to predicates with cognitive components. This is generally consistent with prior findings (An and White, 2020), although we surprisingly find the class of Cognitive (Semi)factives as a whole to be associated with neg-raising inferences, despite that predicates such as *realize* and *know* are standardly thought to be non-neg-raising.

Group 2 consists of clusters capturing a mixture of positive valence internal states, communicatives, and aspectual predicates.

(39) **Evaluatives:** acts of evaluating the truth of a belief.
check, investigate, contemplate, evaluate, examine, explore, etc.

(40) **Positive Internal Emotives:** positive internal emotional states.
content, relieve, amuse, charm, please, comfort, etc.

(41) **Strong Doxastic Communicatives:** acts of communicating a belief where truth is entailed.
affirm, confirm, verify, acknowledge, clarify, etc.

(42) **Weak Doxastic Communicatives:** acts of communicating a belief.
articulate, communicate, explain, write, etc.

(43) **Coercives:** acts of compelling another to perform some action.
coerce, enlist, manipulate, allow, assign, bribe, coax, compel, force, etc.

(44) **Change of State + Aspectuals:** expressions involving a change in state over time.

buy, delete, alter, attempt, insert, operate, cause, etc.

(45) **Portraitives:** acts of depicting some state or action.
defend, imitate, portray, address, conceal, depict, describe, detail, etc.

(46) **Misc Emotives:** expressions of emotion that didn't cleanly fit in other clusters.
faze, cloud, gladden, affront, bear, bless, chide, etc.

Starting with the Evaluatives (39) and Positive Internal Emotives (40), we find predicates involving internal evaluation of doxastic states and predicates representing positive internal emotional states, respectively. The former cluster is nonveridical and is associated with generally positive doxastic and bouletic inferences, although may result in negative inferences about the subject in particular nonfinite transitive frames (see Appendix A.2). Unsurprisingly, the latter cluster is factive[10] – as emotive predicates are typically thought to be (Kiparsky and Kiparsky, 1970) – and associated with positive bouletic inferences with finite complements.

Interestingly, we find two separate classes of communicative predicates involving a communication of beliefs – Strong Doxastic Communicatives (41) and Weak Doxastic Communicatives (42) – that both trigger positive doxastic inferences for either participant; however, the former cluster is veridical – entailing the truth of the complement – while the latter cluster is nonveridical.

We next find a cluster of Coercive (43) predicates; consistent with intuition, these predicates trigger a positive intention inference for the object of the coercion, but a much weaker bouletic inference. This cluster is also the only other two-way implicative cluster (along with Reasoning + Judgment), triggering a positive veridicality inference in positive contexts, and a negative inference in negative contexts. Intriguingly, our model also finds a couple clusters of aspectual predicates[11], one of which – the Change

[10]The factivity inferences are particularly strong in passivized frames; refer to Appendix A.2.

[11]Plausibly, these clusters are differentiated solely by their typical syntactic distributions, since we do not measure inferences related to aspect – though this would be an interesting future analysis.

of State + Aspectuals (44) cluster – is closely related to Coercives and contains veridical predicates that involve a change in state over time.

Finally, we observe clusters of Portraitives (45) (predicates depicting some state or action) as well as Misc Emotives (46) ("miscellaneous" expressions of emotion that didn't fit cleanly in other emotive clusters). The former cluster fails to trigger doxastic inferences about the subject in nonfinite contexts. Plausibly, this is due to the fact that *depicting* an action tends to carry an implication that the action itself was not performed.

Group 3 consists of primarily communicative clusters that involve transmission of information.

(47) **Reactives:** acts of communicating an attitude in relation to a state or event.
audit, discipline, approach, chastise, compliment, praise, demean, etc.

(48) **Informatives:** acts of informing another about a state or event.
inform, remind, tell, advise, alert, ask, email, notify, warn, etc.

(49) **Circumstantials + Aspectuals:** expressions involving the result of some process over time.
bury, come, function, happen, invite, result, etc.

(50) **Persuasives:** communicative acts intended to bring about a change in desires of the target.
hound, pester, educate, interview, badger, brief, bug, challenge, etc.

(51) **Gestural Communicatives:** non-contentful acts of communication.
glare, grin, strut, talk, beam, frown, scowl, sulk, etc.

First, we find two related communicative clusters – Reactives (47) and Informatives (48) – that respectively involve the communication of a bouletic attitude towards a state or event, and the act of informing another about a state or event. As one might expect, reactives trigger slightly positive inferences about the subject's desire for the object to perform some action, but do not trigger inferences about the subject's desire to perform the action themselves. Informatives, on the other hand, are associated with doxastic inferences for both participants (in the finite case); and inferences about both the object's intention to perform an action and the subject's intention for the object to perform an action (in the nonfinite case). Both clusters are nonveridical, or very weakly implicative.

We next observe a second cluster of aspectual predicates – Circumstantials + Aspectuals (49). One interesting finding is that this cluster triggers only weak veridicality inferences in positive contexts, but strong negative veridicality inferences in negative contexts – the converse of the Change of State + Aspectuals cluster. Plausibly, this difference is related to the fact that the predicates in the former set of aspectuals are generally past-oriented, whereas the latter set of aspectuals are generally future-oriented (Moon and White, 2020). An experiment that incorporates temporal orientation judgments into the model may provide evidence for or against this hypothesis; we leave this open as a possible future analysis.

Finally, we find two other communicative clusters – Persuasives (50) and Gestural Communicatives (51). The former, like the cluster of Coercives, contains predicates that involve an intention of the subject for the object of the act to perform some action; unlike Coercives, however, the predicates in this cluster are indirectly mediated through an attempt to bring about a change in the desires of the target. Correspondingly, we observe that this cluster is nonveridical and tends to be associated with weaker bouletic and intention inferences for the object. The latter cluster contains predicates that involve communication of internal states through gestures or other non-contentful speech acts, and triggers a bouletic inference about the desire of the subject to perform some action in nonfinite contexts.

Group 4 consists of clusters that primarily contain *negative attitude* predicates, expressing internal attitudes that involve some sort of negated component.

(52) **Antidoxastics:** expressions of disbelief or doubt in some state or action.
doubt, question, dispute, fear, neglect, worry, etc.

(53) **Prohibitives:** acts of disallowing or rejecting some state or action.
denounce, disallow, face, prohibit, repress, abhor, admonish, etc.

(54) **Negative Attitudes:** negative cognitive attitudes towards some state or action.
decline, fail, hate, neglect, refuse, regret, detest, dislike, etc.

We first discuss the two nonveridical clusters in this group: Antidoxastics (52) and Prohibitives (53). The former cluster – which contains predicates that express disbelief, anti-belief, or doubt in some state or action – inverts the doxastic pattern associated with positive doxastic clusters such as Representationals: it does not trigger doxastic inferences about the subject in a positive context[12], and triggers positive doxastic inferences about the subject when under negation. Interestingly, we also find that, when used in nonfinite transitive constructions, this cluster is associated with positive doxastic inferences for each participant *about the other participant* – suggesting that these predicates may also involve a misalignment of belief states. The Prohibitives cluster, on the other hand, contains predicates that express rejection of some state or action, and is associated with negative bouletic inferences that are targeted by negation.

The third cluster in this group – Negative Attitudes (54) – contain other predicates that express negative internal states. This cluster is the only cluster as a whole to be anti-implicative – triggering negative veridicality inferences that are flipped by negation – however, when we disaggregate inferences by frame (see Appendix A.2), we find that this cluster is strongly factive in the A ___ that S frame. Therefore, this cluster appears to agglomerate several types of predicates – including negative internal emotives (such as *hate* and *dislike*) that are known to be factive (Kiparsky and Kiparsky, 1970;

[12] We observe a weak positive inference with finite complements; however, looking at the disaggregated results in Appendix A.2, we note that this is primarily driven by certain tenseless constructions.

Giannakidou, 2006), strongly anti-implicative predicates such as *fail* and *decline*, and predicates such as *refuse* that have sometimes been called “adversatives” (Klima, 1964). One possible cause of this outcome is that the inclusion of both finite and nonfinite frames in the model weakens the predicate-level distinctions between inference patterns within this cluster.

Group 5 consists of the remaining negative valence clusters that tend to contain predicates expressing negative externalized states or actions.

(55) **Argumentatives:** communicative acts that express disagreement over some desire.
fret, lie, agonize, bitch, brood, object, quarrel, quibble, whine, etc.

(56) **Uncertainty:** expressions of lack of certainty in some belief or action.
confuse, distress, freak out, panic, perplex, puzzle, stump, baffle, etc.

(57) **Miratives:** expressions of displeasure or surprise in some state or action.
depress, disappoint, shock, disgust, displease, embarrass, frustrate, surprise, etc.

(58) **Negative External Emotives:** externalized expressions of negative emotion.
apologize, complain, cry, gloat, growl, sob, weep, etc.

First, we find clusters of Argumentatives (55) and Uncertainty (56) that appear to contain communicative predicates involving disagreement, and external expressions of uncertainty. In particular, the former cluster appears to entail the anti-desire of the subject for some state or action – i.e., it triggers negative bouletic inferences that are flipped under negation – but generates a doxastic presupposition for the subject. The latter cluster appears to trigger similar inference patterns (though generating stronger negative bouletic and intention inferences for the subject), but is additionally factive –

distinguishing it from the cluster of Antidoxastics as well.

Next, we find a cluster of Miratives (57) containing predicates that involve expressions of displeasure or surprise. Standard accounts of mirativity suggest that this attitude involves both a factive, emotive component (Sadock and Zwicky, 1985; Chernilovskaya et al., 2012; Bustamante, 2015) as well as an element of surprise, suddenness, or mental unpreparedness (DeLancey, 1997, 2001; Rett, 2011). Indeed, we observe that this cluster appears to pattern similarly to other emotive clusters: it is factive, triggers negative bouletic inferences about the subject that are targeted by negation, and generates a doxastic presupposition[13]. Interestingly, we find that negative emotive miratives tend to be more prototypical members of this cluster than predicates with weaker emotive components such as *surprise*.

Finally, we find a cluster of Negative External Emotives (58) containing predicates involving externalized expressions of negative emotion. This cluster has a similar pattern of bouletic and doxastic inferences as other emotive clusters, but surprisingly, we find that this cluster is only veridical rather than factive. One possible explanation for this difference may be that external emotives carry an additional actuality inference that may be targeted by negation (e.g., to say that "A didn't *complain* that C happened" is to say that C did not, in fact, happen). We leave this as an open question for future work.

8.4.3 Mapping Clusters to Syntax

We now consider the syntactic distributions of the clusters in our taxonomy learned by the view of the model that was fit using MegaAcceptability. However, rather than directly investigating the relationship between clusters and syntactic frames, we ultimately aim to extract relationships between clusters and the underlying constituents (or syntactic features) that the frames are built from. For example, the frame A was ___ whether S may be parsed into the following constituents: a direct object (NP_obj), an interrogative

[13] These inferences become stronger when looking at the specific frame A was ___ that S.

complement (whether S), and potentially a direct subject (NP_subj) depending on interpretation.

We assume that there is a *linear mapping* $M \in \mathbb{R}^{|\mathcal{C}|\times|\mathcal{S}|}$ from clusters to a set of syntactic features $\mathcal{S}$ such that the syntactic features in a particular frame can be *predicted* from the distribution of clusters that are acceptable in that frame. We treat this as a multi-label classification problem where the objective is to predict for each frame a multi-hot vector where the ith index is 1 if that frame contains feature $\mathcal{S}_i$, and 0 otherwise. One challenge that emerges is that particular frames may have multiple constituent parses; for instance, the example above is ambiguous between two interpretations with and without a direct subject. We account for this by treating two distinct parses as separate data points, but taking the *minimum* loss over all parses p_f for a particular frame f. We therefore use the following binary cross-entropy loss function:

$$\begin{aligned}
&\underset{M}{\text{argmin}} - \sum_{f,s} \text{min}_{p_f} \left(\hat{\phi}_{f,s} \text{log} \phi_{f,p_f,s} + (1-\phi_{f,p_f,s}) \text{log}(1-\hat{\phi}_{f,s}) \right) \\
&\hat{\phi}_{f,s} = \text{logit}^{-1} \left(\sum_{c} a_{f,c} m_{c,s} \right)
\end{aligned}$$

Where $A \in [0,1]^{|\mathcal{F}|\times|\mathcal{C}|}$ is the matrix of cluster acceptability loadings for each frame computed from the fixed effects of the mixture model[14], and $\phi_f \in \{0,1\}^{|\mathcal{P}_f|\times|\mathcal{S}|}$ contains a multi-hot vector of features for each parse of f. We manually create the latter matrices for $|\mathcal{S}| = 27$ constituent types and $|\mathcal{F}| = 49$ frames, with a maximum of 3 parses per frame.

We fit this model using batch gradient descent with a learning rate of 0.1 for 2000 epochs. We show the resulting syntactic feature probabilities $\text{logit}^{-1}(M)$ for each cluster in Figure 8.7, ordering the axes according to a hierarchical clustering. In the remainder of this section, we discuss several interesting distinctions that we find.

[14]The full loadings are included in Appendix A.2.

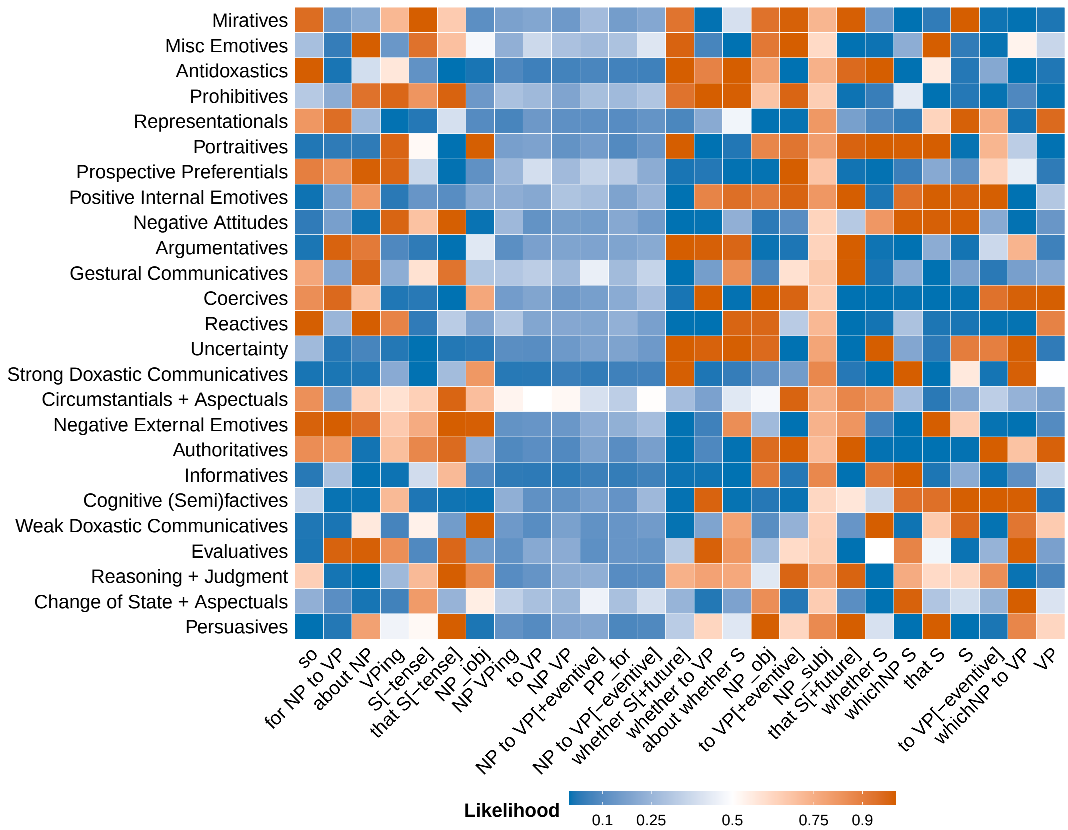

Figure 8.7: Likelihoods that each cluster maps onto a particular syntactic feature.

NP-taking behavior We observe fairly clear distinctions in NP-taking behavior between clusters, with emotive clusters (such as Negative External Emotives and Positive Internal Emotives) and communicative clusters (such as Strong/Weak Doxastic Communicatives, Persuasives, and Informatives – but interestingly, not Gestural Communicatives or Argumentatives) having strong preferences for either a direct object (NP_obj) or indirect object (NP_iobj), and with cognitive/representational clusters showing the weakest preferences for either. The preference for a direct or indirect object among communicatives is likely indicative of the fact that communicatives tend to entail transfer of information from a source (usually characterized by the subject) to a goal (usually characterized by the direct or indirect object); a similar generalization likely holds for external emotives. The tendency of internal emotive clusters to prefer a direct object is perhaps more surprising – however, this is likely attributable to the tendency of internal emotives to appear in expletive subject + experiencer object constructions, such as "It pleased someone that something happened".

We find a further distinction within communicative clusters between those that strongly prefer to realize the goal as an indirect object – e.g., Strong/Weak Doxastic Communicatives, as well as Negative External Emotives and Reasoning + Judgment – those that prefer to realize the goal as a direct object – e.g., Reactives, Informatives, Authoritatives, and Prohibitives – and those that are acceptable with both – e.g., Portraitives and Coercives. These findings broadly replicate prior work by Kane et al. (2021); however, we find clearer evidence for communicative classes that prefer direct objects at the exclusion of indirect objects, including clusters such as Miratives and Antidoxastics that were previously found to have a weak preference for direct objects. It is likely that we observe stronger associations in these cases due to the inclusion of nonfinite transitive constructions, such as (59), as well as passivized experiencer object constructions as mentioned previously.

(59)
a. ?Someone {questioned, worried, embarrassed, surprised} someone that something happened.
b. Someone {questioned, worried, embarrassed, surprised} someone to do something.

Interestingly, we also find a distinction in NP-taking between the two aspectual classes, with the more future-oriented Change of State + Aspectuals cluster preferring a direct object, and the more past-oriented Circumstantials + Aspectuals cluster preferring an indirect object. This appears to attest to the hypothesis that NP-taking behavior tracks underlying distinctions in event structure or aspect (Levin and Rappaport Hovav, 2005), though this relationship requires further exploration.

Finiteness We also observe clear trends in the relationships between clusters and the types of clausal complements that they prefer. For instance, we observe that many clusters containing so-called "assertive" predicates (Hooper, 1975) – e.g., Cognitive (Semi)factives, Representationals, Reasoning + Judgment, Portraitives, and Weak Doxastic Communicatives (but, interestingly, not their Strong counterpart) – as well as several other emotive or bouletic clusters – tend to prefer finite complements (that/whether S variants). On the other hand, clusters that primarily involve intention components – e.g., Authoritatives, Coercives, Prospective Preferentials, and Prohibitives – tend to prefer nonfinite complements (to VP variants). In particular, the distinction between representationals and preferentials mirrors prior generalizations (see Bolinger, 1968). We note, however, that this trend is nowhere near categorical; many clusters, including Representationals, can take both finite and nonfinite complements to some degree.

Eventivity Looking now within clusters that can take nonfinite complements, we also observe interesting distinctions in the eventivity of the embedded verb phrase. For example, though it was noted above that Representationals (and Cognitive (Semi)factives) can take some nonfinite complements, these are heavily biased towards stative (-eventive) complements – whereas Prospective Preferentials are acceptable with both types of complements. This pattern may indicate a structural distinction between the sorts of infinitivals that representationals take and those that preferentials take (see Wurmbrand, 2014).

Embedded Tense In the case of finite complements, we distinguish between the two embedded tenses – past (default that S) and future (that S[+future]) – in the MegaAcceptability dataset. While many clusters are acceptable with both embedded tenses, we find some clusters that tend to prefer future-oriented complements – such as Authoritatives, Gestural Communicatives, Argumentatives, and Miratives – and others that tend to prefer past-oriented complements – such as Weak Doxastic Communicatives and Negative Attitudes. It is likely that these differences mirror distinctions in the temporal interpretations of the underlying events (Moon and White, 2020).

Interrogativity Finally, we turn towards the distinction between interrogative (which/whether variants) and declarative (that/to variants) complements. Many generalizations have been provided relating hypothesized semantic components to alternation between interrogatives and declaratives (Hintikka, 1975; Grimshaw, 1979; Zuber, 1982; Egré, 2008; Theiler et al., 2017, 2019; Uegaki and Sudo, 2019; Roberts, 2019). One common alternation, for instance, draws attention to the distinction between *responsive* predicates – those that are good with both complements, such as (60-b) – and *antirogative* predicates – those that are only good with declarative complements. This is commonly taken to indicate a relationship between veridicality and interrogativity.

(60) a. Someone {thinks, believes, hopes, fears} (that / *whether) S
     b. Someone {knows, understands, doubts, fears} (that / whether) S

We find, indeed, that there are several responsive clusters that allow for both complement types – including Cognitive (Semi)factives and Antidoxastics – and clusters that are antirogative – including Representationals and Persuasives. However, both the Cognitive (Semi)factives and Representationals exhibit weaker preferences than one might expect. This finding is consistent with recent work that finds the support for such generalizations to be considerably weaker when investigating corpus-wide data (White, 2021).

Interestingly, we also find several clusters that exhibit a strong preference for interrogative complements over declarative complements, including intuitive clusters such as Reactives as well as some counter-intuitive clusters that merit further investigation – for example, Uncertainty, Informatives and Strong Doxastic Communicatives.

8.5 Discovering Semantic Components

As an exploratory post hoc analysis – intended to lay the groundwork for future investigation and testing of linguistic generalizations – we try to decompose the clusters into a set of *semantic components* based on shared correlations within the clusters. If the clusters in our taxonomy correspond to distinct inference patterns and syntactic distributions, the components that we aim to uncover represent, in some sense, aspects of the underlying denotational features that combine to form the truth conditions of predicates within each cluster (see e.g. Dowty, 1979) – for instance, bouletic or doxastic modalities.

8.5.1 Model

We induce the semantic components using a fuzzy logic matrix factorization model, following White and Rawlins (2016) and An and White (2020). The intuition behind our approach is that we assume a particular cluster c triggers a particular inference type i – denoted by the boolean indicator variable $r_{c,i}$ – if it has at least one semantic component k that is associated with that inference type – denoted by the boolean indicator variables $l_{c,k}$ and $h_{k,i}$, respectively. More formally, $r_{c,i} \equiv \bigvee_k [l_{c,k} \wedge h_{k,i}]$.

Since our clustering model learns the *probabilities* that each cluster triggers a particular inference type, we convert this definition to a probabilistic fuzzy logic formula (Meghdadi and Akbarzadeh-T, 2001), choosing an axiomatization where $P(x \wedge y) \equiv P(x)P(y)$ and $P(\neg x) \equiv 1 - P(x)$. Specifically, given the mean fixed effect values learned by the mixture model for the three inference datasets – $T \in [0,1]^{|\mathcal{C}| \times |\mathcal{I}|}$ – we derive the following model:

$$
\begin{aligned}
\hat{t}_{c,i} = P(r_{c,i}) &= P\left(\bigvee_k [l_{c,k} \wedge h_{k,i}]\right) \\
&= 1 - \prod_k 1 - u_{c,k} n_{k,i} \\
U^*, N^* &= \underset{U,N}{\text{argmin}} \left(|T - \hat{T}|\right)
\end{aligned}
$$

Where $U \in [0,1]^{|\mathcal{C}| \times |\mathcal{K}|}$ and $N \in [0,1]^{|\mathcal{K}| \times |\mathcal{I}|}$ are the factorized matrices representing the probability that a cluster has a particular semantic component, and that that semantic component gives rise to a particular inference type, respectively[15].

We impose additional regularization on the components by using a modified version of the syntactic feature classifier discussed in Section 8.4.3 to predict the syntactic features associated with each component. Instead of learning a linear mapping from clusters to features, we instead learn a mapping $M \in \mathbb{R}^{|\mathcal{K}|\times|\mathcal{S}|}$ from components to features. We compute the frame probabilities for each component by composing the cluster frame probabilities (i.e., $A \in [0,1]^{|\mathcal{F}|\times|\mathcal{C}|}$) with the component probabilities for each cluster learned by the model, i.e., U:

$$\hat{\phi}_{f,s} = \text{logit}^{-1}\left(\sum_k \left(\sum_c a_{f,c}u_{c,k}\right) m_{c,s}\right)$$
$$M^*, U^* = \underset{M,U}{\text{argmin}} - \sum_{f,s} \text{min}_{p_f} \left(\hat{\phi}_{f,s}\text{log}\phi_{f,p_f,s} + (1-\phi_{f,p_f,s})\text{log}(1-\hat{\phi}_{f,s})\right)$$

We jointly optimize both loss functions during each epoch of training by iteratively optimizing the fuzzy logic inference model (using minibatch gradient descent with a batch size of 20%) and the feature classification model (using batch gradient descent). We fit the model using the Adam optimizer for a maximum of 10000 epochs, using a learning rate of 0.01, stopping if after 1000 epochs the relative change in loss for both models is below a threshold of $1\text{e}{-5}$.

8.5.2 Results

After fitting models with up to 20 components, we choose to analyze the model with $|\mathcal{K}|^* = 10$ components – this number is chosen heuristically by finding the "elbow point"

[15]One caveat with this model definition is that it fails to account for the fact that a particular component may be *negated* by a particular cluster – instead implicitly viewing this as two separate components that give rise to negative or positive inferences. In developing this model, we initially attempted to model negation of a component by a cluster explicitly; however, we found that this model was not able to capture meaningful component negations.

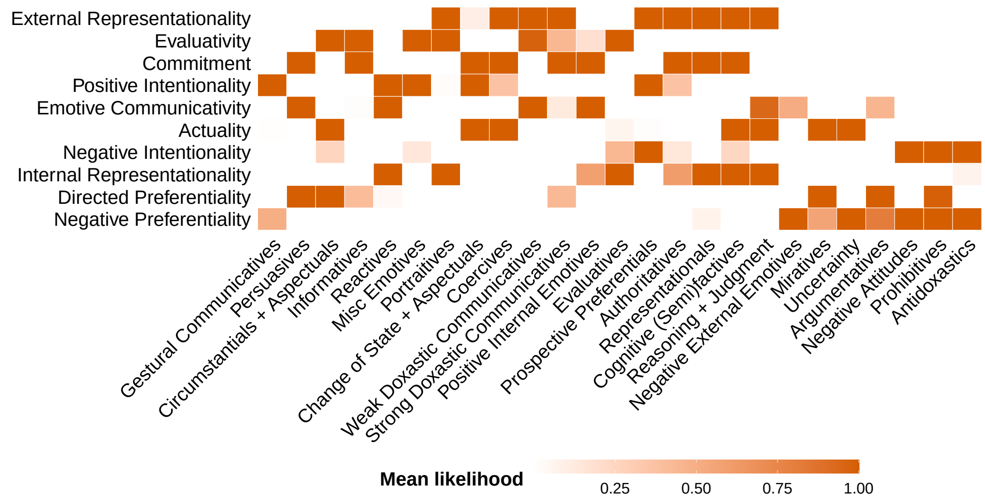

Figure 8.8: Semantic component membership probabilities for each cluster.

in the syntax classification loss, i.e., the point at which introducing an additional component begins to provide marginal explanatory power towards the syntactic distributions of clusters.

Figures 8.8 and 8.9 respectively plot the probabilities that a cluster has a particular semantic component (U) and the probabilities that a particular semantic component maps to a particular syntactic feature (M). In Figures 8.10 – 8.12 we show the probabilities that a particular semantic component gives rise to a particular inference type, averaging across frames for each inference type. The components in each plot are ordered according to a hierarchical clustering of syntactic feature and inference probabilities.

Since the clusters in our model are built from several semantic components, the inferences associated with each cluster reflect superpositions of the underlying component mappings – in other words, the inferential properties of each cluster can be "built from" distinct combinations of components. Similarly, the distributional profile of a particular cluster of predicates will be a function of the syntactic preferences of the semantic components from which that cluster is built – for example, a cluster having two components, one with a preference for direct objects and one with a preference for indirect objects, may be acceptable with either a direct or indirect object.

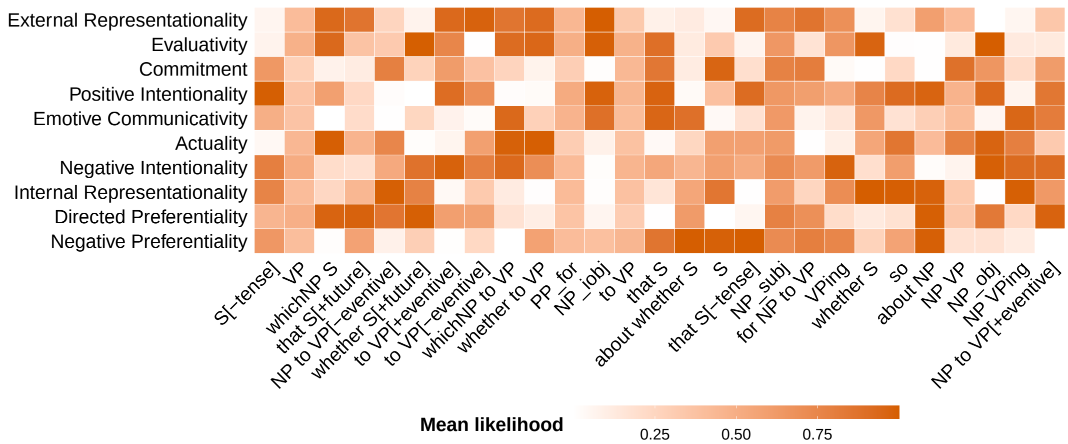

Figure 8.9: Mappings from semantic components to syntactic features.

We form several generalizations from the resulting semantic components that are consonant with prior literature. For example, replicating a finding in (Kane et al., 2021)[16], we find that components associated with neg-raising are anti-correlated with those that are veridical under negation, although this correlation is weaker than previously observed (An and White, 2020). Unsurprisingly, we observe from Figure 8.8 that these components most strongly associate with emotive clusters, which have long been known to be anti-neg-raising as well as factive.

Below, we describe interpretations of the 10 semantic components that result from our model, along with further tentative generalizations. We suggest names for each component on the basis of the clusters that have that component (Figure 8.8); the predicates with highest probability for that component[17] (see Appendix A.3); and the inference patterns associated with each component (Figures 8.10 – 8.12).

Negative Preferentiality represents a negative preference for some state or event.

[16]In contrast to the matrix factorization approach used here, which results in "local" semantic components that combine to form the inference patterns, Kane et al. (2021) use PCA to find "global" components that explain most of the variance in inference patterns.

[17]We compute these probabilities by composing the component probabilities for each cluster with the predicate membership probabilities for each cluster obtained from the mixture model.

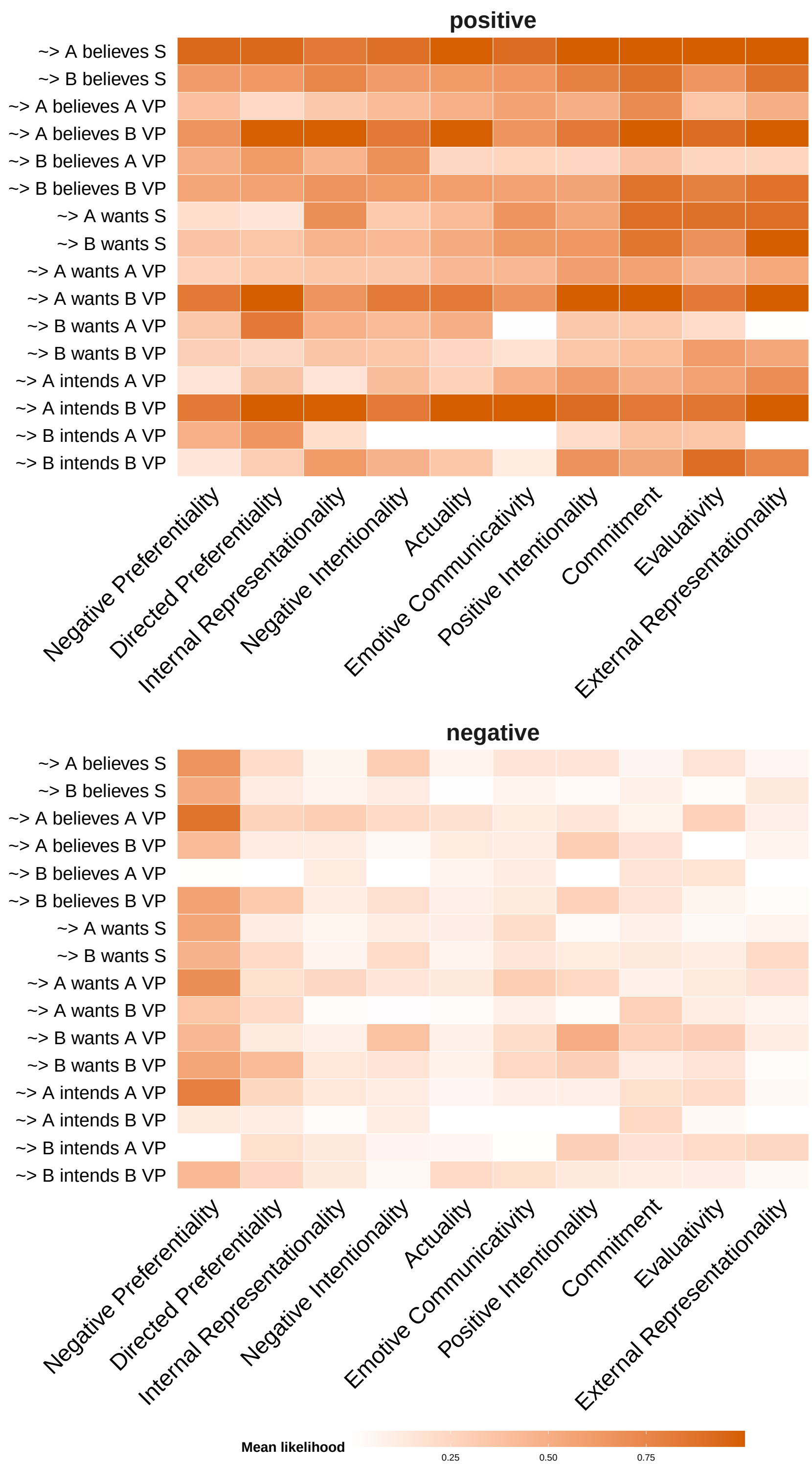

Figure 8.10: Mappings from semantic components to doxastic, bouletic, and intention inferences (averaged across frames).

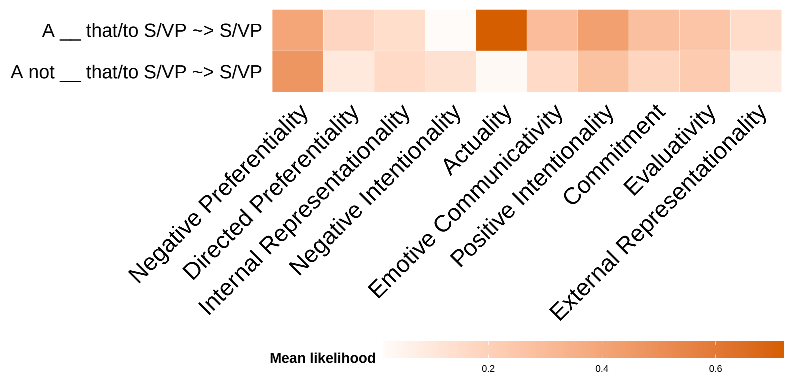

Figure 8.11: Mappings from semantic components to veridicality inferences (averaged across frames).

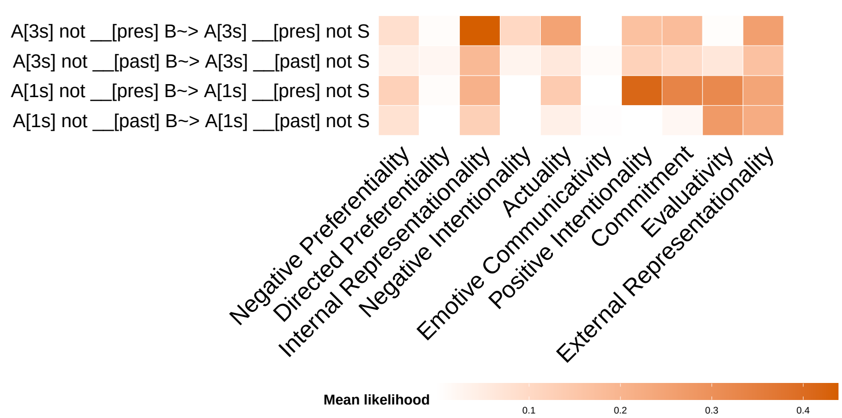

Figure 8.12: Mappings from semantic components to neg-raising inferences (averaged across frames).

(61) detest, loathe, neglect, oppose, reject, resent, disapprove, dislike

Broadly, this component captures negative emotivity, and is shared among all negative emotive clusters (as well as, to a lesser extent, Gestural Communicatives). It tends to prefer intransitive syntactic contexts with finite complements. It is associated with veridicality and doxastic presuppositions, as well as anti-desire and anti-intention inferences that are reversed in negative contexts.

Directed Preferentiality captures preferences that are directed from a source to a target.

(62) result, happen, aggravate, annoy, dishearten, pester, admonish, agitate

This component is shared by a subset of negative valence clusters that involve communication of negative preferential attitudes – namely, Miratives, Argumentatives, and Prohibitives – as well as positive valence clusters such as Persuasives and, curiously, Cirumstantials + Aspectuals. It tends to prefer syntactic contexts with a direct object, and either nonfinite complements or future-oriented finite complements. In contrast with Negative Preferentiality, it is non-veridical, but is associated with bouletic and intention inferences for the target about the source.

Internal Representationality captures representation of the "inner" belief state of an agent.

(63) consider, deem, evaluate, formulate, perceive, picture, recognize

This component is shared by the various "cognitive" clusters that we identified – such as Representationals, Cognitive (Semi)factives, and Reasoning + Judgment – as well as Reactives, Portraitives, and Evaluatives. It tends to prefer intransitive contexts with interrogative or stative nonfinite complements. It is associated with strong neg-raising inferences with a 3rd-person subject under present tense.

Negative Intentionality represents an intention of an agent *against* some event.

(64) neglect, refuse, detest, dislike, hate, reject, resent, decline, disallow

This component is shared primarily by a subset of clusters relating to negative attitudinal states – Antidoxastics, Prohibitives, and Negative Attitudes – but also Prospective Preferentials, which notably contains both Negative and Positive Intentionality components. It tends to prefer syntactic contexts with direct objects and either nonfinite or gerund complements. It is associated with anti-intention inferences for an agent, or in the case of transitive contexts, for the target about the source.

Actuality captures the entailment that some state or event actually occurred (or will occur).

(65) result, delete, store, bury, discover, disgust, find, happen, evidence

This component is shared by both aspectual clusters, as well as Coercives, Cognitive (Semi)factives, Reasoning + Judgment, Miratives, and Uncertainty. It tends to prefer contexts with a direct object and interrogative complements. It is associated with strong veridicality inferences.

Emotive Communicativity represents the communication of bouletic attitudes from a source to a target.

(66) comfort, soothe, aggrieve, attest, commend, congratulate

This component is shared by many emotive and communicative clusters, such as Persuasives, Reactives, Weak Doxastic Communicatives (but, interestingly, not Strong Doxastic Communicatives), Positive Internal Emotives, Negative External Emotives. It also appears for Reasoning + Judgment, possibly suggesting that some predicates of judgment involve both representational and emotive components. It tends to prefer indirect objects.

Positive Intentionality represents an intention of an agent *for* some event.

(67) start, try, aim, appear, attempt, buy, steer, apply

This component is shared by clusters such as Gestural Communicatives, Reactives, Misc Emotives, Change of State + Aspectuals, Prospective Preferentials, as well as (to a lesser degree) Coercives and Authoritatives. It tends to prefer either direct or indirect objects, and nonfinite or tenseless complements. It is associated with reflexive intention inferences for either the subject or object (in the transitive case), as well as neg-raising inferences with a 1st-person subject under present tense.

Commitment broadly encompasses different types of commitments, such as discourse commitment or doxastic commitment.

(68) advise, bet, command, discover, guarantee, instruct, petition, promise, trust, approve

This component is shared by communicative clusters that involve some sort of discourse commitment or obligation – such as Persuasives, Informatives, Change of State + Aspectuals, Coercives, Strong Doxastic Communicatives, Authoritatives, and Positive Internal Emotives – or doxastic commitment, such as Representationals and Cognitive (Semi)factives. It tends to weakly prefer direct objects, and either finite, sentential, or small clause complements. It is associated with doxastic inferences about the subject and object (in the case of transitive contexts) and neg-raising inferences with a 1st-person subject under present tense.

Evaluativity captures the evaluation of the truth-value of a proposition (or intention towards an action) by some agent.

(69) result, cloud, discuss, happen, outline, recount, repeat, review, summarize

The semantic interpretation of this component is somewhat unclear relative to the other components that we observe; however, it is shared by clusters that each involve some sort of evaluation or portrayal by an agent – including Evaluatives, Informatives, Portraitives, and Weak Doxastic Communicatives (as well as, curiously, Circumstantials + Aspectuals and Misc Emotives). This component tends to prefer both direct and indirect objects, and interrogative or finite complements. It is associated with neg-raising inferences with a 1st-person subject, and in the case of transitive nonfinite contexts, with intention inferences about the object.

External Representationality captures communication of an attitude state from a source to a target.

(70) admit, affirm, agree, allege, announce, claim, communicate, confirm, convey, describe

This component is shared by several communicative clusters involving information transfer – Portraitives, Weak and Strong Doxastic Communicatives, and Coercives – as well as cognitive attitude clusters – Prospective Preferentials, Authoritatives, Representationals, Cognitive (Semi)factives, and Reasoning + Judgment. It tends to prefer indirect objects and either nonfinite, interrogative, tenseless, or future-oriented finite complements. It is associated with doxastic and bouletic inferences about both the subject and object (in finite contexts), or about the subject towards the object (in nonfinite contexts). It is also weakly neg-raising across all contexts.

Many of these components are interesting and clearly align with lexical-semantic components that have been proposed in theoretical literature. We also find several components that appear to align with relatively understudied relationships, such as those relating the source of a communicative act with the target of a communicative act. Interestingly, we find that most components are somewhat associated with doxastic, bouletic, and intention inferences – the weakest overall associations appearing to be with the Negative Preferentiality and Emotive Communicativity components – potentially suggesting a root modality that is active in the interpretation of most clusters of predicates. It is also possible, however, that a more fine-grained categorization of inference patterns – including additional types of inferences besides those used for the present analysis – may result in more varied components. We leave this as an interesting direction for future exploration.

8.6 Discussion

In this chapter, we have presented the results of a lexical-scale collection of prototypical belief, desire, and intention inferences, as well as a taxonomy of English predicates derived on the basis of several types of prototypical inferences and syntactic distributions. We concluded with an exploratory analysis that revealed the underlying semantic components that these prototypical inference patterns are constructed from. This work represents, to our knowledge, the first systematic, lexical-scale taxonomy of predicates according to their lexicosemantic inferential properties. The taxonomy and component decomposition provide a fertile ground for linguists to test generalizations about the interface between syntax and semantics.

Apart from the insights into lexical semantics gained through this analysis, such a taxonomy may be useful in the development of dialogue systems such as Eta, and in NLP pipelines more generally. One popular framework for intelligent agents proposed by Bratman (1987) explicitly models agents through their belief, desire, and intention states, while a preponderance of dialogue schemas and plan actions contain conditions related to these states. In both cases, creating precise and semantically-grounded models for inferring belief, desire, and intention states from natural language would allow for more robust and more explainable dialogue management.

We suggest two potentially promising routes for the incorporation of our results into NLP systems. The first route involves using the inference patterns that we uncovered to inform the design of *Natural Logic* (NLog) reasoning models. NLog was initially proposed by Lakoff (1970) as an approach to characterize inferences in natural language (or structures resembling natural language) through lexical entailment relations – for example, monotonic inferences based on generalization and specialization relations, as implemented by MacCartney and Manning (2007). These monotonic inferences are known to be sensitive to semantic contexts including surrounding polarity items and the veridicality or factivity of a matrix predicate (see e.g. Giannakidou, 2006). Stratos et al. (2011) and Kim et al. (2019) demonstrate NLog systems for Episodic Logic and ULF, respectively, that augment monotonic entailment with the ability to generate inferences from a subset of clause-embedding factive and implicative predicates. Our work can build on such systems by (i) greatly expanding the subset of factive and implicative predicates that can modulate monotonic inferences or generate projection inferences; (ii) allowing for the incorporation of other types of lexical entailments; particularly, bouletic, doxastic, intention, and neg-raising entailments; (iii) allowing for the quantification of uncertainty by using our model to assign probabilities to particular inferences.

A second, and more direct, route involves fine-tuning statistical natural language inference (NLI) models on the MegaIntensionality dataset to enable prediction of belief, desire, and intention entailments. In this direction, Gantt et al. (2020) propose a method for inserting mixed effects adapters into pretrained language models (such as

BERT) that allows them to model prototypical inference judgments while accounting for annotator variability, and show that this method substantially improves performance on the MegaVeridicality and MegaNegRaising datasets. A similar model trained on the MegaIntensionality dataset instead of (or in addition to) these datasets may be sufficient to predict the belief, desire, and intention states of agents from natural language text; however, we leave this question to a future investigation.

9 Conclusion

In this dissertation I presented Eta, a general dialogue framework for the creation of conversational agents that uses an explicit dialogue schema representation and a novel schema instantiation method to dynamically drive dialogue. Eta maintains a dialogue state including an episodic memory, abstract schema knowledge, records of partially instantiated schemas, a dialogue plan, and dialogue history. Four parallel processes – perception, reasoning, planning, and execution – operate over this dialogue state. Within each process, abstract *transduction* methods are employed that transform certain entities in the dialogue state; for instance, interpreting an input as a logical form, or expanding a step in the dialogue plan. The functional implementations of these transduction methods are relegated to the domain of application, allowing for seamless integration of pattern-based methods, LLM-based methods, or other methods depending on the needs of the particular domain.

I have provided a comprehensive overview of three separate case studies of conversational agents created with the Eta framework across diverse domains, detailing the design of the schemas and transduction methods that enabled each agent. First, I discussed the LISSA virtual human – a friendly peer for social skill assistance – and our efforts to enable topically broad and personable casual conversation that lead to the genesis of the Eta framework. I also presented more recent work demonstrating how schemas may be combined with LLMs in order to enable robust and engaging persona-based response generation.

Next, I discussed a spatially situated virtual agent, DAVID, that is able to hold collaborative conversations with a user about a physical "blocks world" domain. I described the schema design, transduction methods, and specialist reasoning methods that were required to enable advanced question-answering capabilities, and provided an open-ended evaluation of the system's performance. I also discussed an extension of the system to interactive concept tutoring sessions.

As a final case study, I presented the SOPHIE virtual human – a simulated cancer patient that medical professionals can use to practice essential communication skills in end-of-life scenarios. I presented results from an initial pilot experiment that was conducted with the system, as well as a post-hoc analysis of the transcripts that were obtained from this pilot. I discussed an extension of the SOPHIE agent that incorporates LLMs for interpretation and generation, improving on the limitations observed in the transcript analysis while preserving the strengths of the schema-based framework. This system represents the most elaborate and impactful application of Eta to date, and to our knowledge, there is no other dialogue framework that allows for conversational fluency in various domains to the extent of Eta while still allowing control of the structure and goals of dialogue.

In the final chapter of this dissertation, I shift from the topic of schema-based dialogue management toward a descriptive question: as opposed to interpreting natural language to fit a pre-specified representation of prototypical knowledge within a domain, what sorts of prototypical knowledge can be *inferred* from natural language itself? I described the MegaIntensionality project: an effort to create a lexical-scale dataset of prototypical judgments of belief, desire, and intention inferences from English predicates. I present a comprehensive taxonomy of predicates and corresponding inference patterns derived from this data using a soft clustering model, and an exploratory analysis of the semantic components underlying these clusters.

Several avenues of future work open up as a result of this dissertation. The current state of the Eta framework presented within, although adaptable, is still dependent on domain-specific transducer plugins as well as external servers for specialist reasoning methods. The extensions that we discussed illustrate some steps that we have taken to employing more general domain-independent transduction methods, e.g., using statistical semantic parsers or LLMs such as those proposed by (Kim et al., 2021; Gibson and Lawley, 2022), and likewise using LLMs for controllable generation, as discussed in Chapters 5 and 7. More extensive evaluations of these improvements are necessary, and an effort to carry out a large-scale experiment with the SOPHIE system is currently underway.

Apart from this, several other potential extensions appear to be promising for further closing the gap between domain-specific plugins and domain-independent transduction methods. For instance, LLM or search-based methods for reasoning and plan modification may enhance Eta's capabilities in domains involving complex logistical planning. The data collected in the MegaIntensionality dataset, and the resulting analysis, may be useful in developing models for natural language inference that can allow Eta to more precisely track the belief, desire, and intention states of a user. Finally, enabling automatic acquisition of schemas – e.g., from stories (Lawley et al., 2019) or dialogue examples – and incorporating efficient data-driven methods of schema matching – e.g., attention models for schema alignment, such as those proposed by Mehri and Eskénazi (2021) – would pave the way for a generalist agent competent in diverse domains without the need for explicit schema design.

Bibliography

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. Towards a human-like open-domain chatbot, 2020.

Mohammad Rafayet Ali, Dev Crasta, Li Jin, Agustin Baretto, Joshua Pachter, Ronald D. Rogge, and Mohammed Ehsan Hoque. Lissa — live interactive social skill assistance. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 173–179, 2015. doi: 10.1109/ACII.2015.7344568.

Mohammad Rafayet Ali, Seyedeh Zahra Razavi, Raina Langevin, Abdullah Al Mamun, Benjamin Kane, Reza Rawassizadeh, Lenhart K. Schubert, and Ehsan Hoque. A virtual conversational agent for teens with autism spectrum disorder: Experimental results and design lessons. In *Proceedings of the 20th ACM International Conference on Intelligent Virtual Agents*, IVA '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375863. doi: 10.1145/3383652.3423900. URL https://doi.org/10.1145/3383652.3423900.

James F. Allen and C.Raymond Perrault. Analyzing intention in utterances. *Artificial Intelligence*, 15(3):143–178, 1980. ISSN 0004-3702. doi: https://doi.org/10.1016/0004-3702(80)90042-9. URL https://www.sciencedirect.com/science/article/pii/0004370280900429.

James F. Allen, G. Ferguson, and Amanda Stent. An architecture for more realistic conversational systems. In *IUI '01*, 2001.

Hannah An and Aaron White. The lexical and grammatical sources of neg-raising inferences. *Society for Computation in Linguistics (SCiL)*, 3(1):220–233, January 2020. doi: 10.7275/yts0-q989.

Pranav Anand and Valentine Hacquard. Epistemics and attitudes. *Semantics and Pragmatics*, 6(8):1–59, 2013. doi: 10.3765/sp.6.8.

Pranav Anand and Valentine Hacquard. Factivity, belief and discourse. In Luka Crnič and Uli Sauerland, editors, *The Art and Craft of Semantics: A Festschrift for Irene Heim*, volume 1, pages 69–90. MIT Working Papers in Linguistics, Cambridge, MA, 2014.

Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, Jayant Krishnamurthy, Theo Lanman, Percy Liang, Christopher H. Lin, Ilya Lintsbakh, Andy McGovern, Aleksandr Nisnevich, Adam Pauls, Dmitrij Petters, Brent Read, Dan Roth, Subhro Roy, Jesse Rusak, Beth Short, Div Slomin, Benjamin Snyder, Stephon Striplin, Yu Su, Zachary Tellman, Sam Thomson, Andrei Vorobev, Izabela Witoszko, Jason Wolfe, Abby Wray, Yuchen Zhang, and Alexander Zotov. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571, 09 2020.

David H Arnold, Christopher J Lonigan, Grover J Whitehurst, and Jeffery N Epstein. Accelerating language development through picture book reading: replication and extension to a videotape training format. *Journal of educational psychology*, 86(2): 235, 1994.

Frederic C Bartlett. *Remembering: A study in experimental and social psychology*, volume 14. Cambridge University Press, 1932.

Frederic C. Bartlett and Walter Kintsch. *Remembering: A Study in Experimental and Social Psychology*. Cambridge University Press, 2 edition, 1995. doi: 10.1017/CBO9780511759185.

Jon Barwise and Robin Cooper. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4(2):159–219, June 1981. ISSN 1573-0549. doi: 10.1007/BF00350139.

Tahmina Begum. Doctor patient communication: a review. *Journal of Bangladesh College of Physicians and Surgeons*, 32(2):84–88, 2014.

S.R. Berman. *On the semantics and logical form of wh-clauses*. PhD thesis, University of Massachusetts at Amherst, Amherst, MA, 1991.

Sara Bernardini, Kaska Porayska-Pomsta, and Harini Sampath. Designing an intelligent virtual agent for social communication in autism. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 9(1):9–15, Jun. 2021. doi: 10.1609/aiide.v9i1.12688. URL https://ojs.aaai.org/index.php/AIIDE/article/view/12688.

Timothy W. Bickmore, Daniel Schulman, and Candace L. Sidner. A reusable framework for health counseling dialogue systems based on a behavioral medicine ontology. *Journal of Biomedical Informatics*, 44(2):183–197, 2011. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2010.12.006. URL https://www.sciencedirect.com/science/article/pii/S1532046411000025.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003. doi: 10.5555/944919.944937.

D. Bohus and Alexander I. Rudnicky. The ravenclaw dialog management framework: Architecture and systems. *Comput. Speech Lang.*, 23:332–361, 2009.

Dwight Bolinger. Postposed main phrases: An English rule for the Romance subjunctive. *Canadian Journal of Linguistics*, 14(1):3–30, 1968. doi: 10.1017/S0008413100019629.

Michael Bratman. *Intention, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press, Cambridge, 1987.

Marc Brysbaert and Boris New. Moving beyond Kučera and Francis. *Behavior research methods*, 41(4):977–990, 2009. doi: 10.3758/BRM.41.4.977.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

Teresa Bustamante. Real tense and real aspect in mirativity. *Semantics and Linguistic Theory*, 22:347, 04 2015. doi: 10.3765/salt.v0i0.2650.

Phyllis Butow and Ehsan Hoque. Using artificial intelligence to analyse and teach communication in healthcare. *The Breast*, 50:49–55, 2020. ISSN 0960-9776. doi: https://doi.org/10.1016/j.breast.2020.01.008.

Greg N. Carlson and Beverly Spejewski. Generic passages. *Natural Language Semantics*, 5(2):101–165, 1997. ISSN 0925854X, 1572865X. URL http://www.jstor.org/stable/23748023.

Stephanie Carnell, Shivashankar Halan, Michael Crary, Aarthi Madhavan, and Benjamin Lok. Adapting virtual patient interviews for interviewing skills training of novice healthcare students. In *Proceedings of the 15th International Conference on Intelligence Virtual Agents (IVA 2015)*, volume 9238, Delft, Netherlands, 08 2015. doi: 10.1007/978-3-319-21996-7_5.

Thomas Carroll, Christopher Mooney, and Robert Horowitz. Re-act: remote advanced communication training in a time of crisis. *Journal of pain and symptom management*, 61(2):364–368, 2021.

Nathanael Chambers. Event schema induction with a probabilistic entity-driven model. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, 2013.

Anna Chernilovskaya, Cleo Condoravdi, and Sven Lauer. On the discourse effects of wh-exclamatives. In *Proceedings of the 30th West Coast Conference on Formal Linguistics*. Cascadilla Press, 2012. URL https://api.semanticscholar.org/CorpusID:10714239.

Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. Decontextualization: Making sentences stand-alone. *Transactions of the Association for Computational Linguistics*, 9:447–461, 2021. doi: 10.1162/tacl_a_00377. URL https://aclanthology.org/2021.tacl-1.27.

Herbert H. Clark. *Using Language*. 'Using' Linguistic Books. Cambridge University Press, 1996. doi: 10.1017/CBO9780511620539.

Herbert H. Clark. Context and common ground. In Keith Brown, editor, *Encyclopedia of Language and Linguistics*, pages 105–108. Elsevier, 2006.

Herbert H. Clark and Susan E. Brennan. Grounding in communication. In Lauren Resnick, Levine B., M. John, Stephanie Teasley, and D., editors, *Perspectives on Socially Shared Cognition*, pages 13–1991. American Psychological Association, 1991.

Philip R. Cohen and C. Raymond Perrault. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3(3):177–212, 1979. ISSN 0364-0213. doi: https://doi.

org/10.1016/S0364-0213(79)80006-3. URL https://www.sciencedirect.com/science/article/pii/S0364021379800063.

Judith Degen and Judith Tonhauser. Are there factive predicates? an empirical investigation. *Language*, 98:552 – 591, 2022. URL {h}ttps://ling.auf.net/lingbuzz/005360.

Scott DeLancey. Mirativity: The grammatical marking of unexpected information. *Linguistic Typology*, 1(1):33–52, 1997. doi: doi:10.1515/lity.1997.1.1.33. URL https://doi.org/10.1515/lity.1997.1.1.33.

Scott DeLancey. The mirative and evidentiality. *Journal of Pragmatics*, 33(3): 369–382, 2001. ISSN 0378-2166. doi: https://doi.org/10.1016/S0378-2166(01)80001-1. URL https://www.sciencedirect.com/science/article/pii/S0378216601800011. Evidentiality.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

MR1304975 DiMatteo. The role of the physician in the emerging health care environment. *Western Journal of Medicine*, 168(5):328, 1998.

David Dowty. Thematic proto-roles and argument selection. *Language*, 67(3):547–619, 1991. doi: 10.2307/415037.

David R. Dowty. *Word meaning and Montague grammar: the semantics of verbs and times in generative semantics and in Montague's PTQ*. Number v. 7 in Synthese language library. D. Reidel Pub. Co, Dordrecht ; Boston, 1979. ISBN 978-90-277-1008-6 978-90-277-1009-3.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. Shortcut learning of large language models in natural language understanding. *Commun. ACM*, 67

(1):110–120, dec 2023. ISSN 0001-0782. doi: 10.1145/3596490. URL https://doi.org/10.1145/3596490.

Robin Dunbar, Anna Marriott, and Neill Duncan. Human conversational behavior. *Human nature (Hawthorne, N.Y.)*, 8:231–246, 09 1997. doi: 10.1007/BF02912493.

Paul Egré. Question-embedding and factivity. *Grazer Philosophische Studien*, 77(1): 85–125, 2008. doi: 10.1163/18756735-90000845.

Patrick D Elliott, Nathan Klinedinst, Yasutada Sudo, and Wataru Uegaki. Predicates of relevance and theories of question embedding. *Journal of Semantics*, 34(3):547–554, 2017. doi: 10.1093/jos/ffx008.

Donka Farkas. *Intensional Descriptions and the Romance Subjunctive Mood*. Garland Publishing, New York, 1985. ISBN 0-8240-5426-1.

G. Ferguson and James F. Allen. Trips: An integrated intelligent problem-solving assistant. In *AAAI/IAAI*, 1998.

Richard E. Fikes and Nils J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3): 189–208, 1971. ISSN 0004-3702. doi: https://doi.org/10.1016/0004-3702(71)90010-5. URL https://www.sciencedirect.com/science/article/pii/0004370271900105.

Kevin Fiscella, Peter Franks, Malathi Srinivasan, Richard L Kravitz, and Ronald Epstein. Ratings of physician communication by real and standardized patients. *The Annals of Family Medicine*, 5(2):151–158, 2007.

Nicholas T. Franklin, Kenneth A. Norman, Charan Ranganath, Jeffrey M. Zacks, and Samuel J. Gershman. Structured event memory: a neuro-symbolic model of event cognition. *bioRxiv*, 2019.

Lucian Galescu, Choh Man Teng, James Allen, and Ian Perera. Cogent: A generic dialogue system shell based on a collaborative problem solving model. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 400–409, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5048. URL https://www.aclweb.org/anthology/W18-5048.

William Gantt, Benjamin Kane, and Aaron Steven White. Natural language inference with mixed effects. In Iryna Gurevych, Marianna Apidianaki, and Manaal Faruqui, editors, *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 81–87, Barcelona, Spain (Online), December 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.starsem-1.9.

William Gantt, Benjamin Kane, and Aaron Steven White. The interaction of lexical knowledge and world knowledge in role-anchored inferences. in prep.

Mary M. Garrett. Classical chinese conceptions of argumentation and persuasion. *Argumentation and Advocacy*, 29(3):105–115, 1993. doi: 10.1080/00028533.1993.11951559. URL https://doi.org/10.1080/00028533.1993.11951559.

Benjamin Ross George. *Question embedding and the semantics of answers*. PhD thesis, University of California Los Angeles, 2011.

Anastasia Giannakidou. *The landscape of polarity items*. PhD thesis, University of Groningen, 1997.

Anastasia Giannakidou. Only, emotive factive verbs, and the dual nature of polarity dependency. *Language*, 82(3):575–603, 2006. ISSN 00978507, 15350665. URL http://www.jstor.org/stable/4490205.

Anastasia Giannakidou and Alda Mari. *Truth and Veridicality in Grammar and Thought*. University of Chicago Press, Chicago, 2021.

Erin Gibson and Lane Lawley. Language-model-based parsing and english generation for unscoped episodic logical forms. *The International FLAIRS Conference Proceedings*, 2022. URL https://api.semanticscholar.org/CorpusID:248603501.

Daniel Gildea, Giorgio Satta, and Xiaochang Peng. Cache transition systems for graph parsing. *Computational Linguistics*, 44(1):85–118, April 2018. doi: 10.1162/COLI_a_00308. URL https://aclanthology.org/J18-1004.

Matthew L. Ginsberg. Universal planning: An (almost) universally bad idea. *AI Magazine*, 10(4):40–44, 1989. doi: https://doi.org/10.1609/aimag.v10i4.964. URL https://onlinelibrary.wiley.com/doi/abs/10.1609/aimag.v10i4.964.

Jonathan Ginzburg. Resolving questions, II. *Linguistics and Philosophy*, 18(6):567–609, 1995. doi: 10.1007/BF00983299.

Alessandra Giorgi and Fabio Pianesi. *Tense and Aspect: Form Semantics to Morphosyntax*. Oxford University Press, Oxford, 1997. ISBN 0-19-509192-2.

Lucrezia Grassi, Carmine Tommaso Recchiuto, and Antonio Sgorbissa. Knowledge-grounded dialogue flow management for social robots and conversational agents. *International Journal of Social Robotics*, 14:1273 – 1293, 2021.

Jane Grimshaw. Complement selection and the lexicon. *Linguistic Inquiry*, 10(2):279–326, 1979.

B. Grosz and C. Sidner. Plans for discourse. In *Intentions in Communication*, pages 416–443. MIT Press, 1990.

Barbara J. Grosz and Candace L. Sidner. Attention, intentions, and the structure of discourse. *Comput. Linguist.*, 12(3):175–204, July 1986. ISSN 0891-2017.

Julian Grove and Aaron Steven White. Factivity, presupposition projection, and the role of discrete knowlege in gradient inference judgments. https://lingbuzz.net/lingbuzz/007450/current.pdf, under review.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2023.

Jennifer Fong Ha and Nancy Longnecker. Doctor-patient communication: a review. *Ochsner Journal*, 10(1):38–43, 2010.

Vrindavan Harrison, Juraj Juraska, Wen Cui, Lena Reed, Kevin K. Bowden, Jiaqi Wu, Brian Schwarzmann, Abteen Ebrahimi, Rishi Rajasekaran, Nikhil Varghese, Max Wechsler-Azen, Steve Whittaker, Jeffrey Flanigan, and Marilyn Walker. Athena: Constructing dialogues dynamically with discourse constraints, 2020.

Kurtis Haut, Caleb Wohn, Benjamin Kane, Thomas Carroll, Catherine Guigno, Varun Kumar, Ronald Epstein, Lenhart Schubert, and Ehsan Hoque. Validating a virtual human and automated feedback system for training doctor-patient communication skills. In *Proc. of Affective Computing and Intelligent Interaction (ACII)*, 9 2023.

Irene Heim. Presupposition projection and the semantics of attitude verbs. *Journal of Semantics*, 9(3):183–221, 1992. doi: 10.1093/jos/9.3.183.

Jaakko Hintikka. Different Constructions in Terms of the Basic Epistemological Verbs: A Survey of Some Problems and Proposals. In *The Intentions of Intentionality and Other New Models for Modalities*, pages 1–25. Dordrecht: D. Reidel, 1975.

Jerry R. Hobbs. Ontological promiscuity. In *Annual Meeting of the Association for Computational Linguistics*, 1985. URL https://api.semanticscholar.org/CorpusID:298585.

Michael Hoerger, Ronald M Epstein, Paul C Winters, Kevin Fiscella, Paul R Duberstein, Robert Gramling, Phyllis N Butow, Supriya G Mohile, Paul R Kaesberg, Wan Tang, et al. Values and options in cancer care (voice): study design and rationale for a patient-centered communication and decision-making intervention for physicians, patients with advanced cancer, and their caregivers. *BMC cancer*, 13(1):1–14, 2013.

Joan B. Hooper. On assertive predicates. In John P. Kimball, editor, *Syntax and Semantics*, volume 4, pages 91–124. Academy Press, New York, 1975.

Joan B. Hooper and Sandra A. Thompson. On the applicability of root transformations. *Linguistic Inquiry*, 4(4):465–497, 1973. ISSN 00243892, 15309150. URL http://www.jstor.org/stable/4177789.

Ingrid Hopkins, Michael Gower, Tanuz Pérez, Dana Smith, Franklin Amthor, Casey Wimsatt, and Fred Biasini. Avatar assistant: Improving social skills in students with an asd through a computer-based intervention. *Journal of autism and developmental disorders*, 41:1543–55, 02 2011. doi: 10.1007/s10803-011-1179-z.

Laurence Horn. *On the semantic properties of logical operators in English*. PhD thesis, UCLA, 1972.

Robert Horowitz, Laura Hogan, and Thomas Carroll. Mvp-medical situation, values, and plan: A memorable and useful model for all serious illness conversations. *Journal of Pain and Symptom Management*, 60, 07 2020. doi: 10.1016/j.jpainsymman.2020.07.022.

Kiran Ijaz, Anton Bogdanovych, and Tomas Trescak. Virtual worlds vs books and videos in history education. *Interactive Learning Environments*, 25(7):904–929, 2017.

Léo Jacqmin, Lina M. Rojas Barahona, and Benoit Favre. "do you follow me?": A survey of recent approaches in dialogue state tracking. In Oliver Lemon, Dilek Hakkani-Tur, Junyi Jessy Li, Arash Ashrafzadeh, Daniel Hernández Garcia, Malihe Alikhani, David

Vandyke, and Ondřej Dušek, editors, *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 336–350, Edinburgh, UK, September 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.sigdial-1.33. URL https://aclanthology.org/2022.sigdial-1.33.

Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Kaya Stechly, Mudit Verma, Siddhant Bhambri, Lucas Saldyt, and Anil Murthy. Llms can't plan, but can help planning in llm-modulo frameworks, 2024.

B. Kane, C. Giugno, L. Schubert, K. Haut, C. Wohn, and E. Hoque. Managing emotional dialogue for a virtual cancer patient: A schema-guided approach. *IEEE Transactions on Affective Computing*, PP:1–12, sep 2023. ISSN 1949-3045. doi: 10.1109/TAFFC.2023.3317805.

Benjamin Kane and Lenhart Schubert. We are what we repeatedly do: Inducing and deploying habitual schemas in persona-based responses. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10998–11016, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.678. URL https://aclanthology.org/2023.emnlp-main.678.

Benjamin Kane, Georgiy Platonov, and Lenhart K. Schubert. History-aware question answering in a blocks world dialogue system, 2020.

Benjamin Kane, William Gantt, and Aaron S. White. Intensional gaps: Relating veridicality, factivity, doxasticity, bouleticity, and neg-raising. *Semantics and Linguistic Theory 31*, 2021. URL https://par.nsf.gov/biblio/10300471.

Lauri Karttunen. Implicative verbs. *Language*, 47(2):340–358, 1971a. ISSN 00978507, 15350665. URL http://www.jstor.org/stable/412084.

Lauri Karttunen. Some observations on factivity. *Paper in Linguistics*, 4(1):55–69, 1971b. doi: 10.1080/08351817109370248.

Lauri Karttunen. Presuppositions of compound sentences. *Linguistic Inquiry*, 4(2): 167–193, 1973.

Lauri Karttunen. Syntax and semantics of questions. *Linguistics and Philosophy*, 1(1): 3–44, 1977. doi: 10.1007/BF00351935.

Gene Kim and Lenhart Schubert. Intension, attitude, and tense annotation in a high-fidelity semantic representation. In *Proc. the Workshop Computational Semantics Beyond Events and Roles*, pages 10–15. ACL, April 2017.

Gene Kim, Benjamin Kane, Viet Duong, Muskaan Mendiratta, Graeme McGuire, Sophie Sackstein, Georgiy Platonov, and Lenhart Schubert. Generating discourse inferences from unscoped episodic logical formulas. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 56–65, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3306. URL https://aclanthology.org/W19-3306.

Gene Kim, Viet Duong, Xin Lu, and Lenhart Schubert. A transition-based parser for unscoped episodic logical forms. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 184–201, Groningen, The Netherlands (online), June 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.iwcs-1.18.

Gene Louis Kim and Lenhart Schubert. A type-coherent, expressive representation as an initial step to language understanding. In Simon Dobnik, Stergios Chatzikyriakidis, and Vera Demberg, editors, *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 13–30, Gothenburg, Sweden, May 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-0402. URL https://aclanthology.org/W19-0402.

Paul Kiparsky and Carol Kiparsky. *FACT*, pages 143–173. De Gruyter Mouton, Berlin, Boston, 1970. ISBN 9783111350219. doi: doi:10.1515/9783111350219.143. URL https://doi.org/10.1515/9783111350219.143.

Nikita Kitaev and Dan Klein. Constituency parsing with a self-attentive encoder. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1249. URL https://aclanthology.org/P18-1249.

Edward Klima. Negation in English. In Jerry Fodor and Jerrold Katz, editors, *The Structure of Language: Readings in the Philosophy of Language*. Prentice-Hall, 1964.

Barbara M Korsch and Vida Francis Negrete. Doctor-patient communication. *Scientific American*, 227(2):66–75, 1972.

Utpal Lahiri. *Questions and Answers in Embedded Contexts*. Oxford University Press, 2002. ISBN 0-19-824133-X.

George Lakoff. Linguistics and natural logic. *Synthese*, 22(1):151–271, 1970.

Lane Lawley and Lenhart Schubert. Mining logical event schemas from pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 332–345, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-srw.25. URL https://aclanthology.org/2022.acl-srw.25.

Lane Lawley, Gene Louis Kim, and Lenhart Schubert. Towards natural language story understanding with rich logical schemas. In *Proceedings of the Sixth Workshop on Natural Language and Computer Science*, pages 11–22, 2019.

Lane Lawley, Benjamin Kuehnert, and Lenhart Schubert. Learning general event schemas with episodic logic. In *Proceedings of the 1st and 2nd Workshops on Natural Logic Meets Machine Learning (NALOMA)*, pages 1–6, Groningen, the

Netherlands (online), June 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.naloma-1.1.

Neal Lesh, Charles Rich, and Candace L. Sidner. Collaborating with focused and unfocused users under imperfect communication. In Mathias Bauer, Piotr J. Gmytrasiewicz, and Julita Vassileva, editors, *User Modeling 2001*, pages 64–73, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. ISBN 978-3-540-44566-1.

Anton Leuski and David Traum. Npceditor: Creating virtual human dialogue using information retrieval techniques. *AI Magazine*, 32(2):42–56, 03 2011. doi: 10.1609/aimag.v32i2.2347.

Beth Levin and Malka Rappaport Hovav. Wiping the slate clean: A lexical semantic exploration. *Cognition*, 41(1-3):123–151, December 1991. ISSN 0010-0277. doi: 10.1016/0010-0277(91)90034-2.

Beth Levin and Malka Rappaport Hovav. *Argument Realization*. Cambridge University Press, Cambridge, 2005. ISBN 0-521-66376-8.

Mosh Levy, Shauli Ravfogel, and Yoav Goldberg. Guiding LLM to fool itself: Automatically manipulating machine reading comprehension shortcut triggers. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8495–8505, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.569. URL https://aclanthology.org/2023.findings-emnlp.569.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1014. URL https://aclanthology.org/N16-1014.

Manling Li, Sha Li, Zhenhailong Wang, Lifu Huang, Kyunghyun Cho, Heng Ji, Jiawei Han, and Clare Voss. The future is not one-dimensional: Complex event schema induction by graph modeling for event prediction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5203–5215, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.422. URL https://aclanthology.org/2021.emnlp-main.422.

Karen E. Lochbaum. A collaborative planning model of intentional structure. *Computational Linguistics*, 24(4):525–572, 1998. URL https://www.aclweb.org/anthology/J98-4001.

Bill MacCartney and Christopher D. Manning. Natural logic for textual inference. In Satoshi Sekine, Kentaro Inui, Ido Dagan, Bill Dolan, Danilo Giampiccolo, and Bernardo Magnini, editors, *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200, Prague, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/W07-1431.

Kyle Mahowald, Anna A Ivanova, Idan A Blank, Nancy Kanwisher, Joshua B Tenenbaum, and Evelina Fedorenko. Dissociating language and thought in large language models: a cognitive perspective. *arXiv*, arXiv:2301.06627, 2023.

Kellen Maicher, Douglas Danforth, Alan Price, Laura Zimmerman, Bruce Wilcox, Beth Liston, Holly Cronau, Laurie Belknap, Cynthia Ledford, David Way, Doug Post, Allsion Macerollo, and Milisa Rizer. Developing a conversational virtual standardized patient to enable students to practice history-taking skills. *Simulation in healthcare: journal of the Society for Simulation in Healthcare*, 12:124–131, 04 2017. doi: 10.1097/SIH.0000000000000195.

Bodhisattwa Prasad Majumder, Taylor Berg-Kirkpatrick, Julian J. McAuley, and Harsh

Jhamtani. Unsupervised enrichment of persona-grounded dialog with background stories. In *ACL*, 2021. URL https://arxiv.org/pdf/2106.08364.pdf.

A.H. Meghdadi and M.-R. Akbarzadeh-T. Probabilistic fuzzy logic and probabilistic fuzzy systems. In *10th IEEE International Conference on Fuzzy Systems. (Cat. No.01CH37297)*, volume 3, pages 1127–1130 vol.2, 2001. doi: 10.1109/FUZZ.2001.1008853.

Shikib Mehri and Maxine Eskénazi. Schema-guided paradigm for zero-shot dialog. In *SIGdial*, 2021.

Marvin Minsky. A framework for representing knowledge. Technical report, Department of Computer Science, MIT, Cambridge, MA, USA, 1974.

Ellise Moon and Aaron Steven White. The source of nonfinite temporal interpretation. In Mariam Asatryan, Yixiao Song, and Ayana Whitmal, editors, *Proceedings of the 50th Annual Meeting of the North East Linguistic Society*, volume 3, pages 11–24, Amherst, MA, 2020. GLSA Publications.

Fabrizio Morbini, David DeVault, Kenji Sagae, Jillian Gerten, Angela Nazarian, and David Traum. FLoReS: A Forward Looking, Reward Seeking, Dialogue Manager. In *Natural Interaction with Robots, Knowbots and Smartphones - Putting Spoken Dialog Systems into Practice*, pages 313–325. Springer New York, November 2013. URL http://www.amazon.com/Natural-Interaction-Robots-Knowbots-Smartphones/dp/1461482798/ref=sr_1_1?s=books&ie=UTF8&qid=1409179426&sr=1-1&keywords=Natural+Interaction+with+Robots%2C+Knowbots+and+Smartphones.

Johannes E. M. Mosig, Shikib Mehri, and Thomas Kober. Star: A schema-guided dialog dataset for transfer learning, 2020.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior, 2023.

Ian Perera, James F. Allen, Lucian Galescu, Choh Man Teng, Mark Burstein, Scott Friedman, David McDonald, and Jeffrey Rye. Natural language dialogue for building and learning models and structures. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 5103–5104. AAAI Press, 2017.

Ian Perera, James Allen, Choh Man Teng, and Lucian Galescu. A situated dialogue system for learning structural concepts in blocks world. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 89–98, 2018.

Jean Piaget and Margaret Cook. *The origins of intelligence in children*, volume 8. International Universities Press New York, 1952.

Emmanouil Antonios Platanios, Adam Pauls, Subhro Roy, Yuchen Zhang, Alexander Kyte, Alan Guo, Sam Thomson, Jayant Krishnamurthy, Jason Wolfe, Jacob Andreas, and Dan Klein. Value-agnostic conversational semantic parsing. In *ACL-IJCNLP 2021*, 08 2021.

Georgiy Platonov and Lenhart Schubert. Computational models for spatial prepositions. In *Proceedings of the First International Workshop on Spatial Language Understanding*, pages 21–30, 2018.

Georgiy Platonov, Lenhart Schubert, Benjamin Kane, and Aaron Gindi. A spoken dialogue system for spatial question answering in a physical blocks world. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 128–131, 1st virtual meeting, July 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.sigdial-1.16.

Georgiy Platonov, Benjamin Kane, and Lenhart Schubert. Generating justifications in a spatial question-answering dialogue system for a blocks world. In *Proceedings of the Reasoning and Interaction Conference (ReInAct 2021)*, pages 53–57, Gothenburg, Sweden, October 2021a. Association for Computational Linguistics. URL https://aclanthology.org/2021.reinact-1.8.

Georgiy Platonov, Yifei Yang, Haoyu Wu, Jonathan Waxman, Marcus Hill, and Lenhart Schubert. Modeling semantics and pragmatics of spatial prepositions via hierarchical common-sense primitives. In *Proceedings of Second International Combined Workshop on Spatial Language Understanding and Grounded Communication for Robotics*, pages 32–41, Online, August 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.splurobonlp-1.4. URL https://aclanthology.org/2021.splurobonlp-1.4.

Massimo Poesio and David Traum. Towards an axiomatization of dialogue acts. In *Formal Semantics and Pragmatics of Dialogue (Twendial)*, 05 1998.

Massimo Poesio and David R. Traum. Conversational actions and discourse situations. *Computational Intelligence*, 13(3):309–347, 1997. doi: https://doi.org/10.1111/0824-7935.00042. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/0824-7935.00042.

Martha E. Pollack. Plans as complex mental attitudes. In *Intentions in Communication*, pages 77–103. MIT Press, 1990.

Kathryn I. Pollak, Maren K. Olsen, Hongqiu Yang, Neil Prose, II Jackson, Larry R., Sandro O. Pinheiro, T. Kayla Dunbar, and Kimberly S. Johnson. Effect of a Coaching Intervention to Improve Cardiologist Communication: A Randomized Clinical Trial. *JAMA Internal Medicine*, 04 2023. ISSN 2168-6106. doi: 10.1001/jamainternmed.2023.0629. URL https://doi.org/10.1001/jamainternmed.2023.0629.

Paul Portner. *Situation theory and the semantics of propositional expressions*. PhD thesis, University of Massachusetts, Amherst, 1992.

S. G. Pulman, J. Boye, M. Cavazza, C. Smith, and R. S. de la Cámara. 'how was your day?'. In *Proceedings of the 2010 Workshop on Companionable Dialogue Systems*, CDS '10, page 37–42. Association for Computational Linguistics, 2010. ISBN 9781932432817.

Josep Quer. *Mood at the interface*. PhD thesis, Utrecht Institute of Linguistics, OTS, 1998.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *Proc. of the AAAI Conference on Artificial Intelligence*, 34:8689–8696, 04 2020. doi: 10.1609/aaai.v34i05.6394.

S. Z. Razavi, Lenhart K. Schubert, B. Kane, M. Ali, K. V. Orden, and Tianyi Ma. Dialogue design and management for multi-session casual conversation with older adults. *ArXiv*, abs/1901.06620, 2019.

Seyedeh Zahra Razavi, Mohammad Rafayet Ali, Tristram H. Smith, Lenhart K. Schubert, and Mohammed (Ehsan) Hoque. The lissa virtual human and asd teens: An overview of initial experiments. In David Traum, William Swartout, Peter Khooshabeh, Stefan Kopp, Stefan Scherer, and Anton Leuski, editors, *Intelligent Virtual Agents*, pages 460–463, Cham, 2016. Springer International Publishing. ISBN 978-3-319-47665-0.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

URL https://openreview.net/forum?id=1ikK0kHjvj. Featured Certification, Outstanding Certification.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410.

Drew Reisinger, Rachel Rudinger, Francis Ferraro, Craig Harman, Kyle Rawlins, and Benjamin Van Durme. Semantic proto-roles. *Transactions of the Association for Computational Linguistics*, 3:475–488, 2015. doi: 10.1162/tacl_a_00152. URL https://aclanthology.org/Q15-1034.

Ronald Rensink. *Scene Perception*, pages 151–155. Oxford University Press, 01 2001.

Jessica Rett. Exclamatives, degrees and speech acts. *Linguistics and Philosophy*, 34(5): 411–442, 2011. ISSN 01650157, 15730549. URL http://www.jstor.org/stable/41486885.

Charles Rich, Candace Sidner, and Neal Lesh. Collagen – applying collaborative discourse theory to human-computer interaction. *AI Magazine*, 22:15–26, 12 2001.

David Riedl and Gerhard Schüßler. The influence of doctor-patient communication on health outcomes: a systematic review. *Zeitschrift für Psychosomatische Medizin und Psychotherapie*, 63(2):131–150, 2017.

Albert Rizzo, Belinda Lange, J. Buckwalter, Eric Forbell, Julia Kim, Kenji Sagae, Josh Williams, Barbara Rothbaum, Joann Difede, Greg Reger, Thomas Parsons, and Patrick Kenny. Simcoach: An intelligent virtual human system for providing healthcare information and support. *Studies in health technology and informatics*, 163:503–9, 11 2011. doi: 10.1515/IJDHD.2011.046.

Tom Roberts. I can't believe it's not lexical: Deriving distributed veridicality. *Semantics and Linguistic Theory*, 29(0):665–685, December 2019. ISSN 2163-5951. doi: 10.3765/salt.v29i0.4634.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M. Smith, Y-Lan Boureau, and Jason Weston. Recipes for building an open-domain chatbot, 2020.

Brent Rossen, David Lind, and Benjamin Lok. Human-centered distributed conversational modeling: Efficient modeling of robust virtual human conversations. In *Intelligent Virtual Agents*, pages 474–481, 09 2009. ISBN 978-3-642-04379-6. doi: 10.1007/978-3-642-04380-2_52.

Jonas Rothfuss, Fabio Ferreira, Eren Aksoy, Zhou You, and Tamim Asfour. Deep episodic memory: Encoding, recalling, and predicting episodic experiences for robot action execution. *IEEE Robotics and Automation Letters*, 3, 01 2018.

Jerrold M. Sadock and Arnold M. Zwicky. Speech act distinctions in syntax. In Timothy Shopen, editor, *Language Typology and Syntactic Description*, volume 1, pages 155–196. Cambridge University Press, Cambridge, 1985.

R. Schank and M. Burstein. Artificial intelligence: Modeling memory for language understanding. In T. A. van Dijk, editor, *Handbook of Discourse Analysis. Volume 1: Disciplines of Discourse*, pages 145–166. Appleton Century Crofts, London, 1985.

R.C. Schank and R.P. Abelson. *Scripts, Plans, Goals, and Understanding: An Inquiry Into Human Knowledge Structures*. The Artificial Intelligence Series. Lawrence Erlbaum Associates, 1977. ISBN 9780470990339. URL https://books.google.com/books?id=YZ99AAAAMAAJ.

Lenhart K. Schubert. *The Situations We Talk About*, pages 407–439. Springer US, Boston, MA, 2000. ISBN 978-1-4615-1567-8. doi: 10.1007/978-1-4615-1567-8_18. URL https://doi.org/10.1007/978-1-4615-1567-8_18.

Lenhart K Schubert and Chung Hee Hwang. Episodic logic meets little red riding hood: A comprehensive, natural representation for language understanding. *Natural language processing and knowledge representation: Language for Knowledge and Knowledge for Language*, pages 111–174, 2000.

John R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, 1969. doi: 10.1017/CBO9781139173438.

Chris Segrin. Indirect effects of social skills on health through stress and loneliness. *Health communication*, 34(1):118–124, January 2019. ISSN 1041-0236. doi: 10.1080/10410236.2017.1384434. URL https://doi.org/10.1080/10410236.2017.1384434.

Harry Shum, Xiaodong He, and Di Li. From eliza to xiaoice: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19:10–26, 2018.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.320. URL https://aclanthology.org/2021.findings-emnlp.320.

Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, et al. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188*, 2022.

Mandy Simons. Observations on embedding verbs, evidentiality, and presupposition. *Lingua*, 117(6):1034–1056, 2007. ISSN 0024-3841. doi: https://doi.org/10.1016/j.

lingua.2006.05.006. URL https://www.sciencedirect.com/science/article/pii/S0024384106001288.

Moira A Stewart. Effective physician-patient communication and health outcomes: a review. *CMAJ: Canadian medical association journal*, 152(9):1423, 1995.

Karl Stratos, Lenhart Schubert, and Jonathan Gordon. Episodic logic: Natural logic + reasoning. *KEOD 2011 - Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, 01 2011.

Thomas Talbot, Kenji Sagae, Bruce John, and Albert Rizzo. Designing useful virtual standardized patient encounters. In *Proc. of Interservice/Industry Training, Simulation, and Education Conference*, 01 2012.

Arianne Teherani, Karen E Hauer, and Patricia O'Sullivan. Can simulations measure empathy? considerations on how to assess behavioral empathy via simulations. *Patient education and counseling*, 71(2):148–152, 2008.

Nadine Theiler, Floris Roelofsen, and Maria Aloni. What's wrong with believing whether. *Semantics and Linguistic Theory*, 27:248–265, 2017. doi: 10.3765/salt.v27i0.4125.

Nadine Theiler, Floris Roelofsen, and Maria Aloni. Picky predicates: why believe doesn't like interrogative complements, and other puzzles. *Natural Language Semantics*, pages 95–134, March 2019. ISSN 1572-865X. doi: 10.1007/s11050-019-09152-9.

M. Inés Torres, Javier Mikel Olaso, Neil Glackin, Raquel Justo, and Gérard Chollet. A spoken dialogue system for the empathic virtual coach. In Luis Fernando D'Haro, Rafael E. Banchs, and Haizhou Li, editors, *9th International Workshop on Spoken Dialogue System Technology*, pages 259–265, Singapore, 2019. Springer Singapore. ISBN 978-981-13-9443-0.

D. Traum and James F. Allen. Discourse obligations in dialogue processing. In *ACL*, 1994.

David R. Traum and Staffan Larsson. *The Information State Approach to Dialogue Management*, pages 325–353. Springer Netherlands, Dordrecht, 2003. ISBN 978-94-010-0019-2. doi: 10.1007/978-94-010-0019-2_15. URL https://doi.org/10.1007/978-94-010-0019-2_15.

Roy M. Turner. *Adaptive Reasoning for Real-World Problems: A Schema-Based Approach*. Lawrence Erlbaum, Hillsdale, NJ, 1994.

Wataru Uegaki. *Interpreting questions under attitudes*. PhD thesis, Massachusetts Institute of Technology, 2015.

Wataru Uegaki and Yasutada Sudo. The anti-rogativity of non-veridical preferential predicates. Amsterdam Colloquium 2017, 2017. URL https://semanticsarchive.net/Archive/WNmZDFmM/paper.pdf.

Wataru Uegaki and Yasutada Sudo. The *hope-wh puzzle. *Natural Language Semantics*, 27(4):323–356, December 2019. ISSN 1572-865X. doi: 10.1007/s11050-019-09156-5.

Karthik Valmeekam, Sarath Sreedharan, Matthew Marquez, Alberto Olmo, and Subbarao Kambhampati. On the planning abilities of large language models (a critical investigation with a proposed benchmark), 2023.

Teun Adrianus Van Dijk and Walter Kintsch. *Strategies of discourse comprehension*. Academic press New York, 1983.

Laura Pfeifer Vardoulakis, Lazlo Ring, Barbara Barry, Candace L. Sidner, and Timothy Bickmore. Designing relational agents as long term social companions for older adults. In Yukiko Nakano, Michael Neff, Ana Paiva, and Marilyn Walker, editors, *Intelligent Virtual Agents*, pages 289–302, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-33197-8.

Elisabeth Villalta. Spanish subjunctive clauses require ordered alternatives. *Semantics and Linguistic Theory*, 10:239–256, 2000. doi: 10.3765/salt.v10i0.3123.

Elisabeth Villalta. Mood and gradability: an investigation of the subjunctive mood in Spanish. *Linguistics and Philosophy*, 31(4):467–522, 2008. doi: 10.1007/s10988-008-9046-x.

Marilyn Walker and Steve Whittaker. Mixed initiative in dialogue: An investigation into discourse segmentation. In *Proceedings of the 28th Annual Meeting on Association for Computational Linguistics*, ACL '90, page 70–78. Association for Computational Linguistics, 1990. doi: 10.3115/981823.981833. URL https://doi.org/10.3115/981823.981833.

Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

Aaron Steven White. On believing and hoping whether. *Semantics and Pragmatics*, 14:6, 2021. doi: 10.3765/sp.14.6.

Aaron Steven White and Kyle Rawlins. A computational model of S-selection. *Semantics and Linguistic Theory*, 26:641–663, October 2016. ISSN 2163-5951. doi: 10.3765/salt.v26i0.3819.

Aaron Steven White and Kyle Rawlins. The role of veridicality and factivity in clause selection. In Sherry Hucklebridge and Max Nelson, editors, *Annual Meeting of the North East Linguistic Society (NELS)*, pages 221–234, Amherst, MA, 2018. GLSA Publications.

Aaron Steven White and Kyle Rawlins. Frequency, acceptability, and selection: A case study of clause-embedding. *Glossa: a journal of general linguistics*, 5(1):105, November 2020. ISSN 2397-1835. doi: 10.5334/gjgl.1001.

Aaron Steven White, Kyle Rawlins, and Benjamin Van Durme. The semantic proto-role linking model. In Mirella Lapata, Phil Blunsom, and Alexander Koller, editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 92–98, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-2015.

Terry Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972.

Susi Wurmbrand. Tense and aspect in English infinitives. *Linguistic Inquiry*, 45(3): 403–447, 2014.

Ramin Yaghoubzadeh, Marcel Kramer, Karola Pitsch, and Stefan Kopp. Virtual agents as daily assistants for elderly or cognitively impaired people. In Ruth Aylett, Brigitte Krenn, Catherine Pelachaud, and Hiroshi Shimodaira, editors, *Intelligent Virtual Agents*, pages 79–91, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-40415-3.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of Thoughts: Deliberate problem solving with large language models, 2023.

Zhou Yu, Alan W. Black, and Alexander I. Rudnicky. Learning conversational systems that interleave task and non-task content. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 4214–4220. AAAI Press, 2017. ISBN 9780999241103.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1205. URL https://aclanthology.org/P18-1205.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-demos.30. URL https://aclanthology.org/2020.acl-demos.30.

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and Xiaoyan Zhu. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, 63:1–17, 09 2020b. doi: 10.1007/s11431-020-1692-3.

Chujie Zheng and Minlie Huang. Exploring prompt-based few-shot learning for grounded dialog generation, 2022.

Richard Zuber. Semantic restrictions on certain complementizers. In S. Hattori and K. Inoue, editors, *International Congress of Linguists (ICL)*, pages 434–436, Tokyo, 1982.

A Full MegaIntensionality Results

A.1 Complete Cluster Verb Memberships

In Table A.1, we provide a full list of high-probability predicates falling within each cluster in our taxonomy. We show verbs falling within three cumulative probability thresholds – i.e., we abbreviate $0.5 \leq p > 0.4$ as $p > 0.4$. Note that, due to the nature of our soft clustering model, the same predicate may appear in multiple clusters if it has a particularly entropic distribution (possibly suggesting a high degree of polysemy).

A.2 By-frame Inference Patterns

Figures A.1 – A.5 show the full inference patterns for veridicality, neg-raising, intention, doxastic, and bouletic inferences, respectively. Each facet of a plot represents a particular inference type, while individual frames are displayed on the y-axis.

To make the plots more interpretable, we filter out inferences in frames that aren't acceptable in a particular cluster, based on the model's predicted values for the frames in the MegaAcceptability dataset. Specifically, we filter out frames that are in the bottom quarter percentile for ranked acceptability scores.

We additionally show the full syntactic distributions learned by the MegaAcceptability view of our model – for present tense, past tense, and past progressive tense, respectively – in Figures A.6 – Figure A.8.

A.3 Complete Component Verb Memberships

In Table A.2, we provide a full list of high-probability predicates falling within each semantic component in our matrix factorization model. These probabilities are obtained by composing the component probabilities for each cluster with the predicate membership probabilities for each cluster obtained from the mixture model. We show verbs falling within four cumulative probability thresholds – i.e., we abbreviate $0.7 \leq p > 0.6$ as $p > 0.6$.

Cluster	Threshold	Predicates
Gestural Communicatives	$p > 0.5$	glare, grin, strut, talk
	$p > 0.4$	beam, feud, frown, repent, scowl, smirk, snivel, sulk, wallow, wheeze
	$p > 0.3$	chime, cough, cringe, listen, meditate, mope, pout, rankle, sigh
Persuasives	$p > 0.5$	hound, pester
	$p > 0.4$	educate, interview
	$p > 0.3$	badger, brief, bug, challenge, consult, correct, counsel, deceive, frighten, harass, interest, misinform, mislead, pressure, probe, scare, taunt, torment
Circumstantials + Aspectuals	$p > 0.5$	bury, carp, come, function, happen, invite, punt, result, skirmish, turn out
	$p > 0.4$	bleat, bore, cloud, evidence, forgo, hire, seem, shoot, sober, store, taste, wound
	$p > 0.3$	accredit, cease, compete, configure, delete, disquiet, end, get, hinder, inscribe, jade, meet, nauseate, nonplus, remain, resume, serve, snub, stand, take, whoop
Informatives	$p > 0.5$	inform, remind, tell
	$p > 0.4$	advise, alert, ask, email, notify, warn
	$p > 0.3$	assure, caution, instruct, lecture, reassure, show, teach
Reactives	$p > 0.5$	audit, discipline
	$p > 0.4$	approach, chastise, distract, need, soothe

	$p > 0.3$	compliment, demean, frame, hustle, praise, recruit, reprimand, serve, slander, stereotype, summon, tantalize, thank, worship
Misc Emotives	$p > 0.5$	faze
	$p > 0.4$	cloud, gladden
	$p > 0.3$	affront, bear, bless, chide, embitter, jar, malign, set, sorrow
Portraitives	$p > 0.5$	
	$p > 0.4$	defend, imitate, portray
	$p > 0.3$	address, conceal, depict, describe, detail, endorse, expose, identify, justify, mention, point out, recount, showcase, televise, uncover
Change of State + Aspectuals	$p > 0.5$	buy, delete
	$p > 0.4$	alter, attempt, calibrate, insert, keep, operate, shape, smell, store
	$p > 0.3$	back, catch, cause, commence, cover, dig, disprefer, dub, end, glimpse, gurgle, handle, help, inspect, isolate, manufacture, reconstruct, rediscover, relearn, snort, spot, start, steer, taste, underscore, view
Coercives	$p > 0.5$	coerce, enlist, manipulate
	$p > 0.4$	allow, assign, bribe, coax, compel, force, hire, motivate, recruit, train, trick
	$p > 0.3$	appoint, bully, choose, commission, direct, dispatch, dupe, entice, guide, help, influence, inspire, invite, oblige, permit, provoke, require, rush, select, sign up, spur, summon, tempt
Weak Doxastic Communicatives	$p > 0.5$	

	$p > 0.4$	articulate, blog, brag, communicate, explain, mumble, murmur, mutter, preach, type, whisper, write
	$p > 0.3$	babble, boast, clarify, comment, confess, demonstrate, divulge, elaborate, embellish, exclaim, express, gossip, holler, insinuate, joke, narrate, post, read, reiterate, remark, repeat, reply, restate, share, shout, shriek, sing, transmit, utter, yell
Strong Doxastic Communicatives	$p > 0.5$	
	$p > 0.4$	affirm, confirm, verify
	$p > 0.3$	acknowledge, clarify, declare, disclose, ensure, guarantee, indicate, note, point out, prove, reaffirm, report, reveal, signal, specify, stress
Positive Internal Emotives	$p > 0.5$	content, relieve
	$p > 0.4$	amuse, awe, charm, comfort, delight, elate, enthrall, enthuse, exhilerate, flatter, gratify, humble, please, satisfy, thrill
	$p > 0.3$	aggrieve, electrify, enchant, energize, excite, hearten, intrigue, invigorate, rile, spellbind, stimulate
Evaluatives	$p > 0.5$	check, investigate
	$p > 0.4$	contemplate, evaluate, examine, explore, reconsider, research, review
	$p > 0.3$	analyze, brainstorm, calculate, consider, discuss, ponder, reevaluate, reexamine, study, test, wonder
Prospective Preferentials	$p > 0.5$	aim, appear, come around, compete, hunger, long, look, lust, try, yearn

	$p > 0.4$	apply, attempt, come out, scramble, seem, stand, start, struggle
	$p > 0.3$	bargain, be, begin, come, conspire, continue, crave, hope, love, pause, pine, plan, plot, procrastinate, scheme, set about, smile, start off, thirst
Authoritatives	$p > 0.5$	expect
	$p > 0.4$	authorize, desire, request
	$p > 0.3$	allow, approve, arrange, command, demand, fancy, forbid, mandate, need, order, permit, prefer, recommend, seek, want
Representationals	$p > 0.5$	
	$p > 0.4$	dream, hope, think
	$p > 0.3$	agree, assume, bet, concur, daydream, deserve, figure, guess, insist, pray, presume, pretend, proclaim, say, scream, shout, theorize, wish
Cognitive (Semi)factives	$p > 0.5$	discover, realize, recognize
	$p > 0.4$	find, find out, know, notice, see, understand
	$p > 0.3$	accept, comprehend, detect, figure out, hear, identify, observe, overhear, remember
Reasoning + Judgment	$p > 0.5$	manage
	$p > 0.4$	glean, opt, presuppose, signify
	$p > 0.3$	attest, chronicle, conclude, deem, derive, discern, expound, formulate, make out, reason out, reckon, submit, venture, warrant
Negative External Emotives	$p > 0.5$	
	$p > 0.4$	apologize, cackle, complain, cry, gloat, growl, sob, stutter, weep

	$p > 0.3$	banter, chirp, chuckle, cringe, curse, fume, gasp, giggle, grimace, groan, grunt, gurgle, laugh, moan, pout, rant, sigh, snap, snicker, snitch, spout, squeal, whimper, whine
Miratives	$p > 0.5$	depress, disappoint, disgruntle, disgust, displease, embarrass, frustrate, horrify, outrage, sadden, shock
	$p > 0.4$	amaze, anger, annoy, appall, astonish, astound, baffle, devastate, dishearten, dismay, dissatisfy, enrage, infuriate, irk, irritate, mortify, offend, overwhelm, perturb, sicken, spook, stun, surprise, traumatize, upset
	$p > 0.3$	aggravate, alarm, demoralize, disgrace, disturb, fascinate, floor, fluster, humiliate, madden, pain, perplex, petrify, please, startle, terrify
Uncertainty	$p > 0.5$	
	$p > 0.4$	confuse, distress, freak out, panic, perplex, puzzle, stump
	$p > 0.3$	anguish, appall, baffle, befuddle, bewilder, concern, devastate, disillusion, disturb, fascinate, fluster, grieve, irritate, mystify, petrify, quiz, strain, stress, terrify, trouble, unsettle, vex, worry
Argumentatives	$p > 0.5$	
	$p > 0.4$	fret, lie
	$p > 0.3$	agonize, bitch, brood, object, quarrel, quibble, scoff, whine
Negative Attitudes	$p > 0.5$	decline, fail, hate, neglect, refuse, regret

	$p > 0.4$	cease, detest, dislike, forget, resent
	$p > 0.3$	adore, dismiss, dread, enjoy, fear, loathe, oppose, reject
Prohibitives	$p > 0.5$	
	$p > 0.4$	denounce, disallow, face, prohibit, repress
	$p > 0.3$	abhor, admonish, belittle, blast, constrain, demystify, detest, dislike, disparage, forbid, loathe, mistrust, pardon, reproach
Antidoxastics	$p > 0.5$	doubt, question
	$p > 0.4$	dispute
	$p > 0.3$	fear, neglect, worry

Table A.1: A full list of high-probability predicates within each cluster, shown at three cumulative probability thresholds.

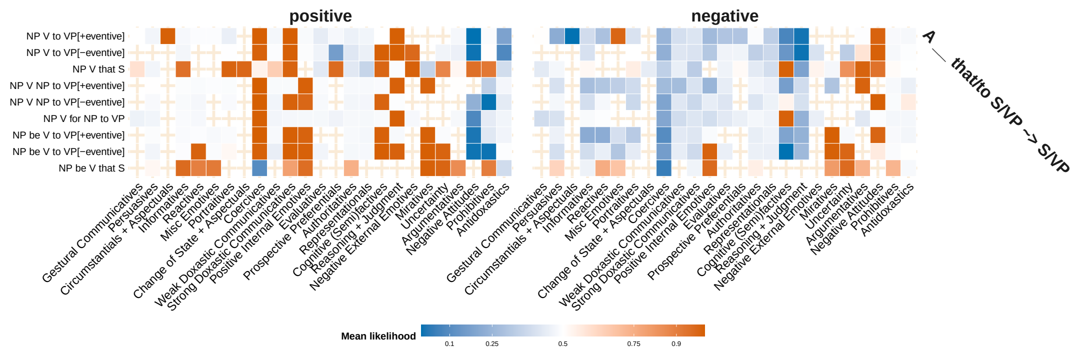

Figure A.1: Prototypical veridicality inference patterns for each cluster.

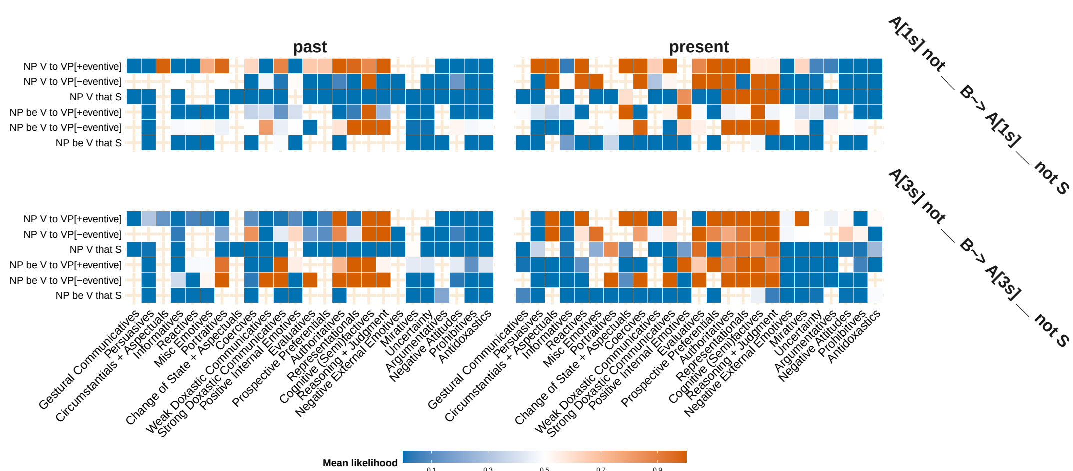

Figure A.2: Prototypical neg-raising inference patterns for each cluster.

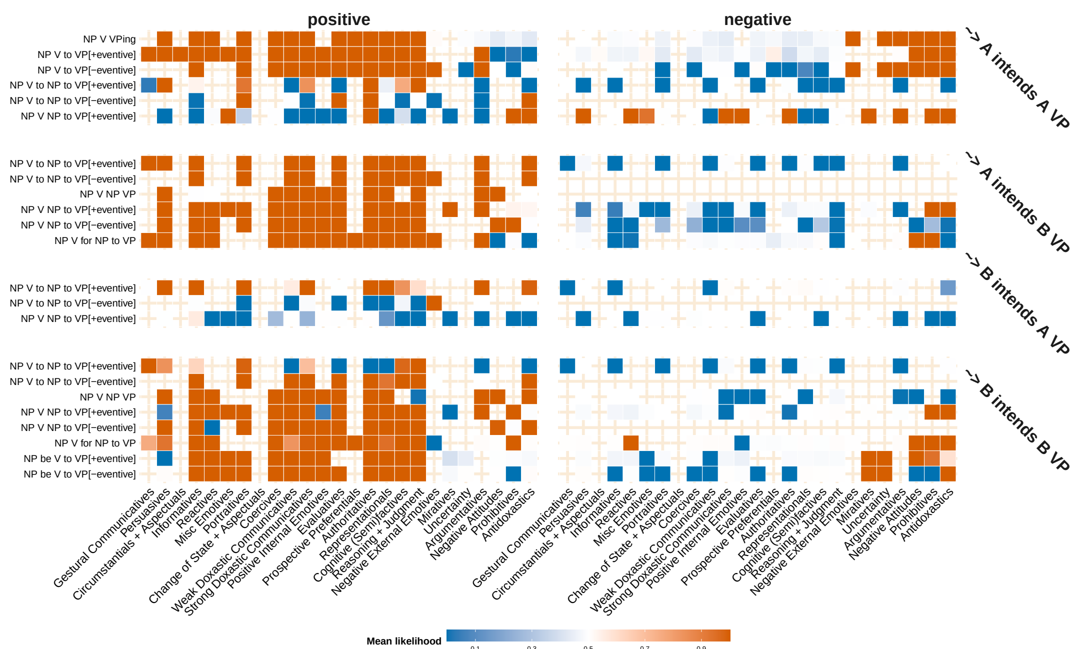

Figure A.3: Prototypical intention inference patterns for each cluster.

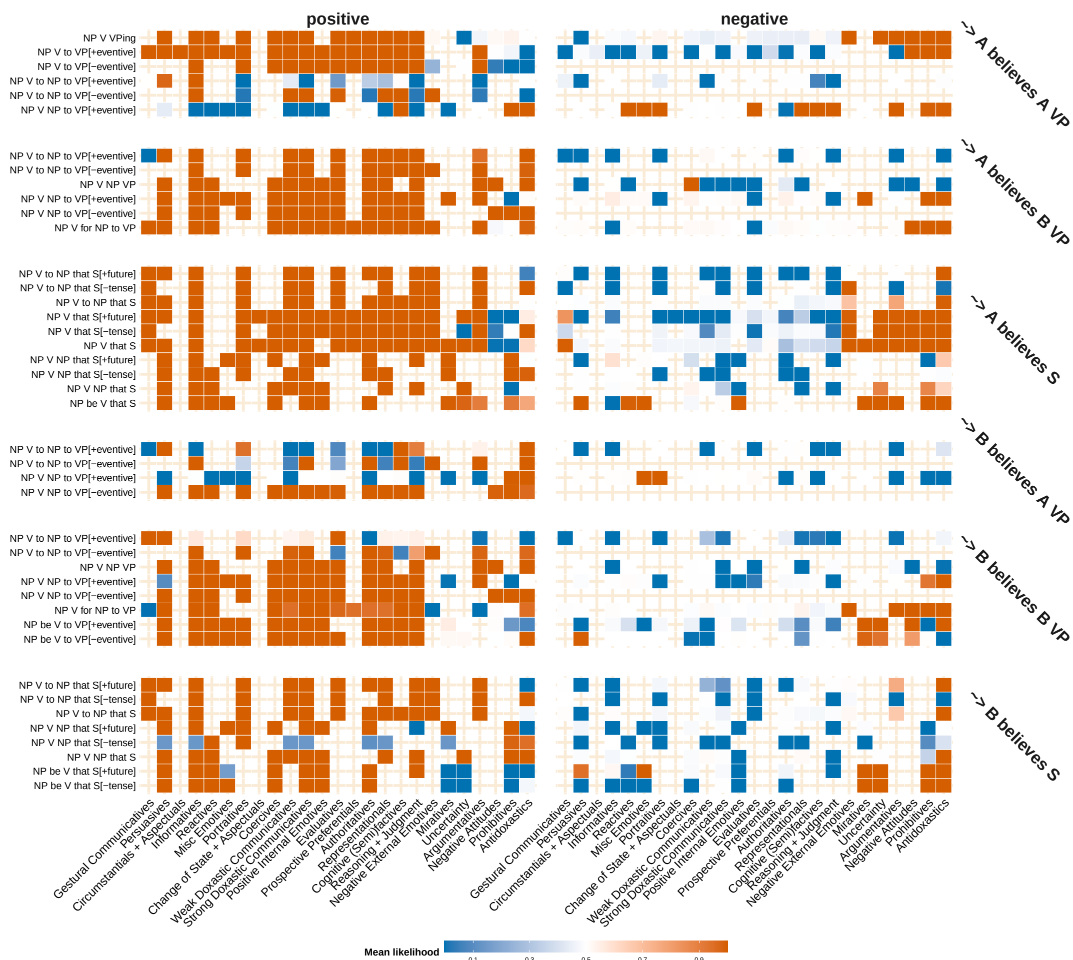

Figure A.4: Prototypical doxastic inference patterns for each cluster.

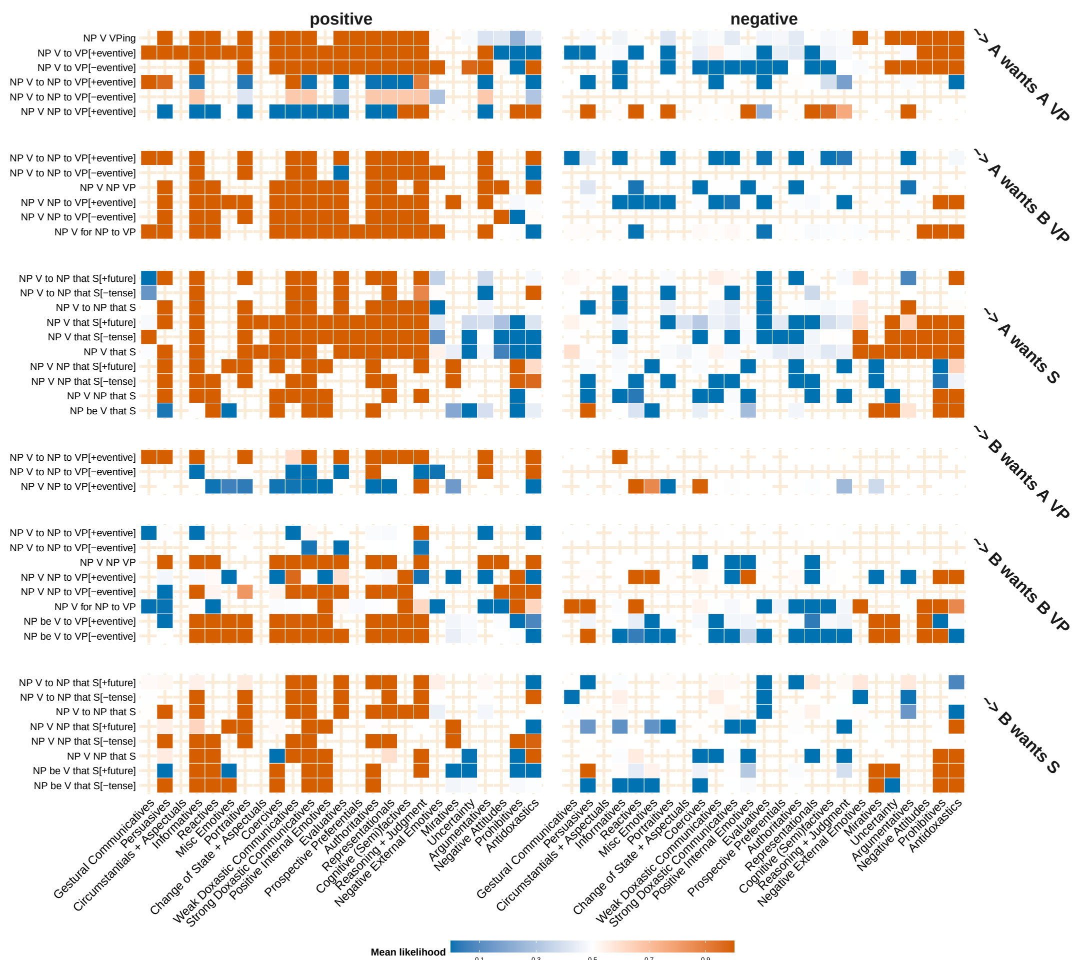

Figure A.5: Prototypical bouletic inference patterns for each cluster.

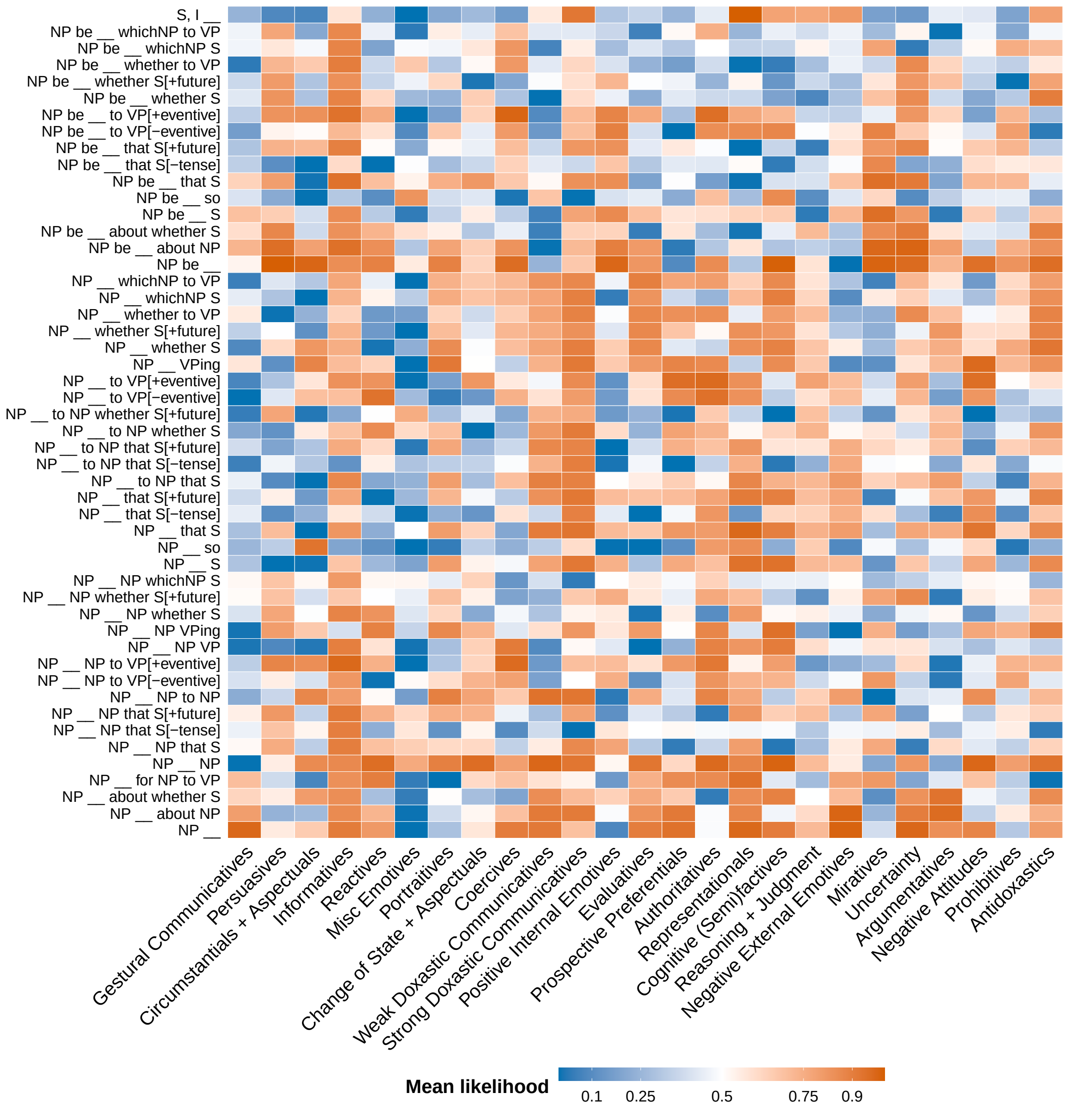

Figure A.6: Prototypical present tense syntactic distributions for each cluster.

Figure A.7: Prototypical past tense syntactic distributions for each cluster.

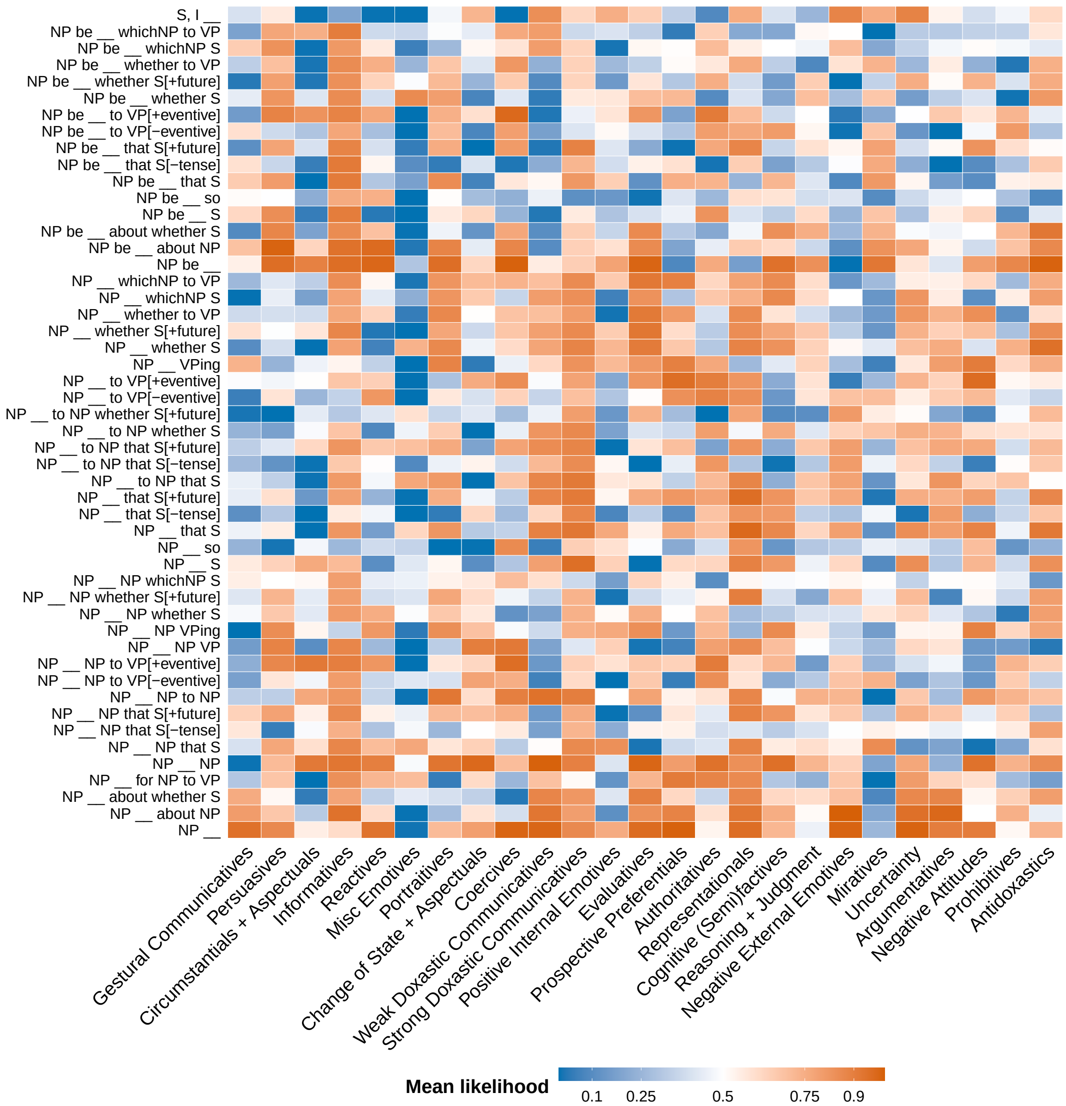

Figure A.8: Prototypical past progressive tense syntactic distributions for each cluster.

Component	Threshold	Predicates
Internal Representation-ality	$p > 0.9$	
	$p > 0.8$	consider, deem, evaluate, formulate, perceive, picture, recognize, register, visualize
	$p > 0.7$	analyze, appreciate, ascertain, audit, calculate, check, comprehend, conceive, contemplate, deduce, defend, depict, detect, document, envision, examine, foresee, gauge, hear, imagine, investigate, make_out, manage, mark, measure, monitor, pinpoint, portray, presume, record, reevaluate, sense, showcase, surmise, televise, think, witness
	$p > 0.6$	accept, address, agree, allege, anticipate, applaud, arrange, assess, assume, celebrate, certify, characterize, cheer, chronicle, commend, conclude, control, decide, demean, derive, deserve, detail, devise, discipline, discover, discuss, distract, endorse, envy, establish, estimate, expect, feel, figure, figure_out, find_out, forecast, glean, glorify, guess, hope, identify, imitate, include, justify, know, name, need, observe, opt
Emotive Communicativity	$p > 0.9$	
	$p > 0.8$	comfort, soothe
	$p > 0.7$	aggrieve, attest, commend, congratulate, distract, rouse, submit

	$p > 0.6$	approach, audit, charm, chastise, console, content, discipline, enchant, engage, enthrall, enthuse, exhilarate, humble, intrigue, presuppose, relieve, reprimand, rile, satisfy, spellbind, stimulate, tantalize, terrorize
Negative Preferentiality	$p > 0.9$	detest, loathe, neglect, oppose, reject, resent
	$p > 0.8$	anguish, curse, decline, disapprove, dislike, dismiss, disparage, disregard, distress, fail, grieve, hate, refuse, regret, repress, weep
	$p > 0.7$	abhor, agitate, baffle, brood, complain, cringe, denounce, deplore, despair, devastate, disconcert, dismay, distrust, doubt, dread, fear, flip_out, forget, fume, grimace, ignore, mourn, object, panic, puzzle, revolt, scoff, worry
	$p > 0.6$	agonize, apologize, appall, befuddle, condemn, confuse, cry, deny, disallow, disbelieve, discourage, dispute, disturb, exasperate, fluster, freak_out, fret, frustrate, fuss, gloat, gripe, groan, growl, hesitate, hush_up, irk, irritate, laugh, lie, misjudge, mistrust, outrage, overlook, perplex, perturb, pout, question, sadden, scowl, sigh, sneer, sob, spook, stew, stump, stun, sulk, terrify, trouble, underestimate
Evaluativity	$p > 0.9$	result
	$p > 0.8$	cloud, discuss, happen
	$p > 0.7$	fax, faze, outline, recount, repeat, review, sober, summarize, write

	$p > 0.6$	advise, articulate, ask, bear, blog, bore, brainstorm, broadcast, carp, clarify, contemplate, describe, detail, disquiet, divulge, email, embellish, endorse, establish, explain, express, fake, function, illustrate, imitate, inscribe, insinuate, mention, narrate, overestimate, ponder, punt, reevaluate, reiterate, remind, research, share, shoot, show, skirmish, televise, tell, transmit, turn_out, type, utter, weigh
Negative Intentionality	$p > 0.9$	neglect, refuse
	$p > 0.8$	detest, dislike, hate, reject, resent
	$p > 0.7$	decline, disallow, dismiss, disregard, fail, fear, ignore, loathe, oppose, repress
	$p > 0.6$	appear, cease, compete, condemn, denounce, dispute, distrust, doubt, dread, enjoy, forget, misjudge, mistrust, overlook, prohibit, regret, underestimate
Commitment	$p > 0.9$	advise, bet, command, discover, guarantee, instruct, petition, promise, prompt, see, trust
	$p > 0.8$	alert, allow, approve, arrange, ask, authorize, believe, bribe, brief, caution, choose, coax, coerce, compel, confirm, convince, demand, designate, determine, direct, email, encourage, expect, find, force, guide, help, hound, inform, know, manipulate, notify, observe, permit, pester, press, pressure, rediscover, remind, report, require, show, signal, teach, tell, think, threaten, understand, urge, warn

	$p > 0.7$	affirm, allege, assure, badger, beg, cajole, contract, counsel, dare, declare, delight, dupe, embolden, ensure, entice, find_out, gratify, hear, implore, incite, insist, inspire, intrigue, license, make, mandate, motivate, move, note, order, persuade, phone, pick, presume, proclaim, propose, provoke, radio, reaffirm, realize, reassure, recognize, relieve, remember, request, reveal, say, select, specify, suggest
	$p > 0.6$	accept, acknowledge, assign, assume, back, bug, buy, cause, certify, challenge, charm, claim, coach, comfort, comprehend, consult, content, correct, decree, deduce, dictate, educate, elate, elect, enforce, enlighten, enlist, enthrall, envision, feel, figure, figure_out, foresee, foretell, gather, glimpse, hint, hope, humble, identify, imagine, imply, indicate, induce, influence, insure, interest, lecture, lobby, misinform
Actuality	$p > 0.9$	delete, hire, result, store
	$p > 0.8$	bury, configure, depress, discover, disgust, disillusion, displease, embarrass, evidence, find, get, happen, help, horrify, invite, make, manage, perplex, sadden, taste, unsettle, view

	$p > 0.7$	accredit, appall, baffle, calibrate, cause, comprehend, crush, devastate, disappoint, disgruntle, dishearten, dissatisfy, disturb, end, fluster, frustrate, infuriate, insert, irk, irritate, manufacture, mortify, nauseate, nonplus, offend, outrage, print, punt, puzzle, realize, recognize, rediscover, resume, select, shock, shoot, stump, stun, surprise, tackle, terrify, turn_out, upset, use, vex
	$p > 0.6$	aggravate, alter, anger, annoy, astonish, astound, befuddle, begin, bore, carp, choose, come, contrive, cover, diagnose, dismay, distress, enlist, enrage, fascinate, fool, force, function, glean, glimpse, jade, know, label, manipulate, measure, mystify, operate, overwhelm, pain, perturb, petrify, pick, remain, see, shape, sicken, skirmish, smell, spook, spot, spur, startle, take, trick, witness
Positive Intentionality	$p > 0.9$	start, try
	$p > 0.8$	aim, appear, attempt, buy, steer
	$p > 0.7$	apply, commence, credential, discriminate, hunger, hustle, isolate, look, lust, mope, operate, smell, sorrow, start_off, suffer

	$p > 0.6$	audit, be, bear, begin, bless, calibrate, chime, come_around, crave, design, dig, disprefer, dub, endure, frown, galvanize, gladden, glimpse, grin, handle, help, insert, keep, legislate, long, malign, mean, meditate, need, scramble, set_out, shape, shatter, smile, smirk, snort, soothe, stand, stereotype, strut, swoon, tap, undertake, want, wheeze, worship, yearn
External Representationality	$p > 0.9$	admit, affirm, agree, allege, announce, claim, communicate, conclude, confirm, convey, declare, demonstrate, describe, disclose, emphasize, exclaim, hope, identify, imply, indicate, infer, long, manage, murmur, note, proclaim, propose, reiterate, report, restate, say, scream, shout, specify, speculate, state, verify, voice, whisper
	$p > 0.8$	accept, acknowledge, approve, arrange, articulate, assert, attest, boast, broadcast, certify, choose, clarify, conceive, confess, conspire, corroborate, deem, defend, demand, depict, dictate, divulge, dream, expect, explain, expose, express, foresee, guess, hint, illustrate, imagine, insinuate, know, love, maintain, mandate, mention, mumble, narrate, notice, observe, perceive, permit, picture, point_out, portray, pray, predict, presume

	$p > 0.7$	advertise, advocate, aim, allow, appear, ascertain, assess, assign, assume, authorize, brag, celebrate, come_around, come_out, command, comment, comprehend, concede, concur, confide, consent, crave, decide, decree, deduce, designate, desire, devise, diagnose, discover, document, elect, establish, extrapolate, fabricate, fantasize, figure, figure_out, find, flaunt, force, gamble, hear, highlight, holler, insist, intend, interpret, justify, learn
	$p > 0.6$	address, annotate, anticipate, appreciate, argue, attempt, babble, beg, believe, blog, care, change, chant, chronicle, commission, compete, conceal, consider, contend, contribute, dare, daydream, detect, direct, discern, discuss, display, elaborate, embellish, endorse, enlist, ensure, envision, estimate, expound, fancy, feel, find_out, formulate, gauge, glean, gossip, grant, guarantee, hanker, hunger, initiate, joke, leak, like
Directed Preferentiality	$p > 0.9$	result
	$p > 0.8$	happen
	$p > 0.7$	aggravate, annoy, dishearten, function, pester

	$p > 0.6$	admonish, agitate, anger, belittle, bore, bother, carp, counsel, deplore, depress, disappoint, disgruntle, disgust, disparage, displease, dissatisfy, enrage, face, forgo, frighten, frustrate, grouse, horrify, humiliate, irk, misinform, nauseate, nonplus, notify, repress, ridicule, sadden, spook, surprise, torment, trouble, upset

Table A.2: A full list of high-probability predicates within each semantic component, shown at four cumulative probability thresholds.