Learning to Interpret Utterances Using Dialogue History

David DeVault
Institute for Creative Technologies
University of Southern California
Marina del Rey, CA 90292
devault@ict.usc.edu

Matthew Stone
Department of Computer Science
Rutgers University
Piscataway, NJ 08845-8019
Matthew.Stone@rutgers.edu

Abstract

We describe a methodology for learning a disambiguation model for deep pragmatic interpretations in the context of situated task-oriented dialogue. The system accumulates training examples for ambiguity resolution by tracking the fates of alternative interpretations across dialogue, including subsequent clarificatory episodes initiated by the system itself. We illustrate with a case study building maximum entropy models over abductive interpretations in a referential communication task. The resulting model correctly resolves 81% of ambiguities left unresolved by an initial handcrafted baseline. A key innovation is that our method draws exclusively on a system's own skills and experience and requires no human annotation.

1 Introduction

In dialogue, the basic problem of interpretation is to identify the contribution a speaker is making to the conversation. There is much to recognize: the domain objects and properties the speaker is referring to; the kind of action that the speaker is performing; the presuppositions and implicatures that relate that action to the ongoing task. Nevertheless, since the seminal work of Hobbs et al. (1993), it has been possible to conceptualize pragmatic interpretation as a unified reasoning process that selects a representation of the speaker's contribution that is most preferred according to a background model of how speakers tend to behave.

In principle, the problem of pragmatic interpretation is qualitatively no different from the many problems that have been tackled successfully by data-driven models in NLP. However, while researchers have shown that it is sometimes possible to annotate corpora that capture *features* of interpretation, to provide empirical support for theories, as in (Eugenio et al., 2000), or to build classifiers that assist in dialogue reasoning, as in (Jordan and Walker, 2005), it is rarely feasible to fully annotate the interpretations themselves. The distinctions that must be encoded are subtle, theoretically-loaded and task-specific—and they are not always signaled unambiguously by the speaker. See (Poesio and Vieira, 1998; Poesio and Artstein, 2005), for example, for an overview of problems of vagueness, underspecification and ambiguity in reference annotation.

As an alternative to annotation, we argue here that dialogue systems can and should prepare their own training data by inference from underspecified models, which provide sets of candidate meanings, and from skilled engagement with their interlocutors, who know which meanings are right. Our specific approach is based on *contribution tracking* (DeVault, 2008), a framework which casts linguistic inference in situated, task-oriented dialogue in probabilistic terms. In contribution tracking, ambiguous utterances may result in alternative possible contexts. As subsequent utterances are interpreted in those contexts, ambiguities may ramify, cascade, or disappear, giving new insight into the pattern of activity that the interlocutor is engaged in. For example, consider what happens if the system initiates clarification. The interlocutor's answer may indicate not only what they mean now but also what they must have meant earlier when they used the original ambiguous utterance.

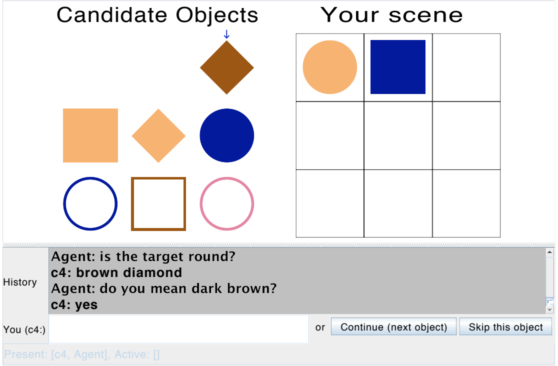
Contribution tracking allows a system to accumulate training examples for ambiguity resolution by tracking the fates of alternative interpretations across dialogue. The system can use these examples to improve its models of pragmatic interpretation. To demonstrate the feasibility of this approach in realistic situations, we present a system that tracks contributions to a referential communication task using an abductive interpretation model: see Section 2. A user study with this system, described in Section 3, shows that this system can, in the course of interacting with its users, discover the correct interpretations of many potentially ambiguous utterances. The system thereby automatically acquires a body of training data in its native representations. We use this data to build a maximum entropy model of pragmatic interpretation in our referential communication task. After training, we correctly resolve 81% of the ambiguities left open in our handcrafted baseline.

2 Contribution tracking

We continue a tradition of research that uses simple referential communication tasks to explore the organization and processing of human–computer and mediated human–human conversation, including recently (DeVault and Stone, 2007; Gergle et al., 2007; Healey and Mills, 2006; Schlangen and Fernández, 2007). Our specific task is a two-player object-identification game adapted from the experiments of Clark and Wilkes-Gibbs (1986) and Brennan and Clark (1996); see Section 2.1. To play this game, our agent, COREF, interprets utterances as performing sequences of task-specific problem-solving acts using a combination of grammar-based constraint inference and abductive plan recognition; see Section 2.2. Crucially, COREF's capabilities also include the ambiguity management skills described in Section 2.3, including policies for asking and answering clarification questions.

2.1 A referential communication task

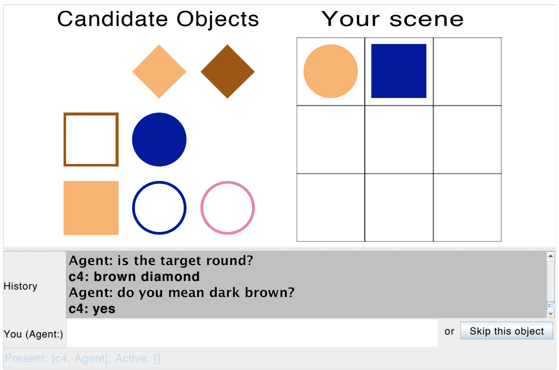
The game plays out in a special-purpose graphical interface, which can support either human–human or human–agent interactions. Two players work together to create a specific configuration of objects, or a *scene*, by adding objects into the scene one at a time. Their interfaces display the same set of candidate objects (geometric objects that differ in shape, color and pattern), but their locations are shuffled. The shuffling undermines the use of spatial expressions such as "the object at bottom left". Figures 1 and 2 illustrate the different views.[1]



Agent: is the target round?
c4: brown diamond
Agent: do you mean dark brown?
c4: yes

Present: [c4, Agent], Active: []

Figure 1: A human user plays an object identification game with COREF. The figure shows the perspective of the user (denoted c4). The user is playing the role of director, and trying to identify the diamond at upper right (indicated to the user by the blue arrow) to COREF.



Agent: is the target round?
c4: brown diamond
Agent: do you mean dark brown?
c4: yes

Present: [c4, Agent], Active: []

Figure 2: The conversation of Figure 1 from COREF's perspective. COREF is playing the role of matcher, and trying to determine which object the user wants COREF to identify.

As in the experiments of Clark and Wilkes-Gibbs (1986) and Brennan and Clark (1996), one of the players, who plays the role of *director*, instructs the other player, who plays the role of *matcher*, which object is to be added next to the scene. As the game proceeds, the next target object is automatically determined by the interface and privately indicated to the director with a blue arrow, as shown in Figure 1. (Note that the corresponding matcher's perspective, shown in Figure 2, does not include the blue arrow.) The director's job is then to get the matcher to click on (their version of) this target object.

To achieve agreement about the target, the two players can exchange text through an instant-messaging modality. (This is the only communi-

[1] Note that in a human–human game, there are literally two versions of the graphical interface on the separate computers the human participants are using. In a human–agent interaction, COREF does not literally use the graphical interface, but the information that COREF is provided is limited to the information the graphical interface would provide to a human participant. For example, COREF is not aware of the locations of objects on its partner's screen.

cation channel.) Each player's interface provides a real-time indication that their partner is "Active" while their partner is composing an utterance, but the interface does not show in real-time what is being typed. Once the `Enter` key is pressed, the utterance appears to both players at the bottom of a scrollable display which provides full access to all the previous utterances in the dialogue.

When the matcher clicks on an object they believe is the target, their version of that object is privately moved into their scene. The director has no visible indication that the matcher has clicked on an object. However, the director needs to click the `Continue (next object)` button (see Figure 1) in order to move the current target into the *director's* scene, and move on to the next target object. This means that the players need to discuss not just what the target object is, but also whether the matcher has added it, so that they can coordinate on the right moment to move on to the next object. If this coordination succeeds, then after the director and matcher have completed a series of objects, they will have created the exact same scene in their separate interfaces.

2.2 Interpreting user utterances

COREF treats interpretation broadly as a problem of abductive intention recognition (Hobbs et al., 1993).[2] We give a brief sketch here to highlight the content of COREF's representations, the sources of information that COREF uses to construct them, and the demands they place on disambiguation. See DeVault (2008) for full details.

COREF's utterance interpretations take the form of action sequences that it believes would constitute coherent contributions to the dialogue task in the current context. Interpretations are constructed abductively in that the initial actions in the sequence need not be directly tied to observable events; they may be *tacit* in the terminology of Thomason et al. (2006). Examples of such tacit actions include clicking an object, initiating a clarification, or abandoning a previous question. As a concrete example, consider utterance (1b) from the dialogue of Figure 1, repeated here as (1):

(1) a. COREF: is the target round?
b. `c4`: brown diamond
c. COREF: do you mean dark brown?
d. `c4`: yes

In interpreting (1b), COREF hypothesizes that the user has tacitly abandoned the agent's question in (1a). In fact, COREF identifies two possible interpretations for (1b):

$i_{2,1}$= ⟨ `c4:tacitAbandonTasks[2],`
`c4:addcr[t7,rhombus(t7)],`
`c4:setPrag[inFocus(t7)],`
`c4:addcr[t7,saddlebrown(t7)]`⟩

$i_{2,2}$= ⟨ `c4:tacitAbandonTasks[2],`
`c4:addcr[t7,rhombus(t7)],`
`c4:setPrag[inFocus(t7)],`
`c4:addcr[t7,sandybrown(t7)]`⟩

Both interpretations begin by assuming that user `c4` has tacitly abandoned the previous question, and then further analyze the utterance as performing three additional dialogue acts. When a dialogue act is preceded by tacit actions in an interpretation, the speaker of the utterance *implicates* that the earlier tacit actions have taken place (DeVault, 2008). These implicatures are an important part of the interlocutors' coordination in COREF's dialogues, but they are a major obstacle to annotating interpretations by hand.

Action sequences such as $i_{2,1}$ and $i_{2,2}$ are coherent only when they match the state of the ongoing referential communication game and the semantic and pragmatic status of information in the dialogue. COREF tracks these connections by maintaining a probability distribution over a set of dialogue states, each of which represents a possible thread that resolves the ambiguities in the dialogue history. For performance reasons, COREF entertains up to three alternative threads of interpretation; COREF strategically drops down to the single most probable thread at the moment each object is completed. Each dialogue state represents the stack of processes underway in the referential communication game; constituent activities include problem-solving interactions such as identifying an object, information-seeking interactions such as question–answer pairs, and grounding processes such as acknowledgment and clarification. Dialogue states also represent pragmatic information including recent utterances and referents which are salient or in focus.

COREF abductively recognizes the intention I of an actor in three steps. First, for each dialogue state s_k, COREF builds a *horizon graph* of possible tacit action sequences that could be assumed coherently, given the pending tasks (DeVault, 2008).

Second, COREF uses the horizon graph and other resources to solve any constraints associ-

[2] In fact, the same reasoning interprets utterances, button presses and the other actions COREF observes!

ated with the observed action. This step instantiates any free parameters associated with the action to contextually relevant values. For utterances, the relevant constraints are identified by parsing the utterance using a hand-built, lexicalized tree-adjoining grammar. In interpreting (1b), the parse yields an ambiguity in the dialogue act associated with the word "brown", which may mean either of the two shades of brown in Figure 1, which COREF distinguishes using its `saddlebrown` and `sandybrown` concepts.

Once COREF has identified a set of interpretations $\{i_{t,1}, ..., i_{t,n}\}$ for an utterance o at time t, the last step is to assign a probability to each. In general, we conceive of this following Hobbs et al. (1993): the agent should weigh the different assumptions that went into constructing each interpretation.[3] Ultimately, this process should be made sensitive to the rich range of factors that are available from COREF's deep representation of the dialogue state and the input utterance—this is our project in this paper. However, in our initial implemented prototype, COREF assigned these probabilities using a simple hand-built model considering only N_{T}, the number of tacit actions abductively assumed to occur in an interpretation:

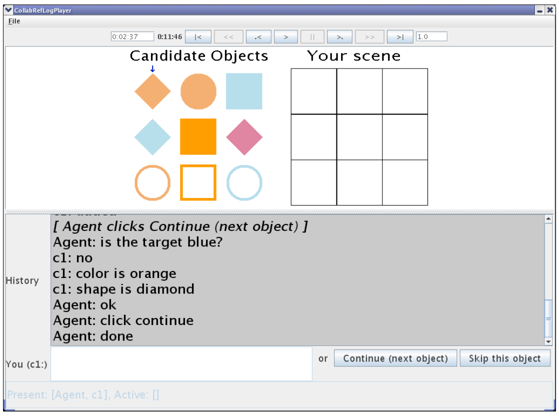
$$P(I = i_{t,j} | o, S_t = s_k) \propto \frac{1}{N_{\mathrm{T}}(i_{t,j}) + 1} \qquad (1)$$

In effect, this is a "null hypothesis" that assigns relatively uniform weights to different abductive hypotheses.

2.3 Interactive disambiguation

COREF uses its probabilistic model of context in order to tolerate ambiguity as it moves forward with its dialogues and to resolve ambiguity over time. We have put particular effort into COREF's skills with three kinds of ambiguity: word-sense ambiguities, where COREF finds multiple resolutions for the domain concept evoked by the use of a lexical item, as in the interaction (1) of Figure 1; referential ambiguities, where COREF takes a noun phrase to be compatible with multiple objects from the display; and speech act ambiguities, where alternative interpretations communicate or implicate different kinds of contributions to the ongoing task.

The resolution of ambiguity may involve some combination of asking questions of the user, aggregating information provided by the user across multiple turns of dialogue, and strategically dropping threads of interpretation. For example, COREF represents the context resulting from (1b) in terms of two states: one from interpretation $i_{2,1}$ and one from $i_{2,2}$. COREF asks a clarification question (1c); the user's explicit answer *yes* allows COREF to discard one of the possible states and allocate all its probability mass to the other one.



[Agent clicks Continue (next object)]
Agent: is the target blue?
c1: no
c1: color is orange
c1: shape is diamond
Agent: ok
Agent: click continue
Agent: done

Figure 3: A human user plays the director role in an object identification game with COREF. The figure shows the perspective of the user.

The dialogue in (2) shows an alternative case.

(2) a. `c1`: color is orange
b. `c1`: shape is diamond
c. COREF: ok

The example is taken from the setting shown in Figure 3. In this case, COREF finds two colors on the screen it thinks the user could intend to evoke with the word *orange*; the peachy orange of the diamond and circle on the top row and the brighter orange of the solid and empty squares in the middle column. COREF responds to the ambiguity by introducing two states which track the alternative colors. Immediately COREF gets an additional description from the user, and adds the constraint that the object is a diamond. As there is no bright orange diamond, there is no way to interpret the user's utterance in the bright orange state; COREF discards this state and allocates all its probability mass to the other one.

3 Inferring the fates of interpretations

Our approach is based on the observation that COREF's contribution tracking can be viewed as assigning a *fate* to every dialogue state it entertains as part of some thread of interpretation. In

[3] Though note that Hobbs et al. do not explicitly construe their weights in terms of probabilities.

particular, if we consider the agent's contribution tracking retrospectively, every dialogue state can be assigned a fate of *correct* or *incorrect*, where a state is viewed as correct if it or some of its descendants eventually capture all the probability mass that COREF is distributing across the viable surviving states, and incorrect otherwise.

In general, there are two ways that a state can end up with fate *incorrect*. One way is that the state and all of its descendants are eventually denied any probability mass due to a failure to interpret a subsequent utterance or action as a coherent contribution from any of those states. In this case, we say that the incorrect state was *eliminated*. The second way a state can end up incorrect is if COREF makes a strategic decision to drop the state, or all of its surviving descendants, at a time when the state or its descendants were assigned nonzero probability mass. In this case we say that the incorrect state was *dropped*. Meanwhile, because COREF drops all states but one after each object is completed, there is a single hypothesized state at each time t whose descendants will ultimately capture all of COREF's probability mass. Thus, for each time t, COREF will retrospectively classify exactly one state as correct.

Of course, we really want to classify interpretations. Because we seek to estimate $P(I = i_{t,j}|o, S_t = s_k)$, which conditions the probability assigned to $I = i_{t,j}$ on the correctness of state s_k, we consider only those interpretations arising in states that are retrospectively identified as correct. For each such interpretation, we start from the state where that interpretation is adopted and trace forward to a correct state or to its last surviving descendant. We classify the interpretation the same way as that final state, either *correct*, *eliminated*, or *dropped*.

We harvested a training set using this methodology from the transcripts of a previous evaluation experiment designed to exercise COREF's ambiguity management skills. The data comes from 20 subjects—most of them undergraduates participating for course credit—who interacted with COREF over the web in three rounds of the referential communication each. The number of objects increased from 4 to 9 to 16 across rounds; the roles of director and matcher alternated in each round, with the initial role assigned at random.

Of the 3275 sensory events that COREF interpreted in these dialogues, from the (retrospectively) correct state, COREF hypothesized 0 interpretations for 345 events, 1 interpretation for 2612 events, and more than one interpretation for 318 events. The overall distribution in the number of interpretations hypothesized from the correct state is given in Figure 4.

N	Percentage	N	Percentage
0	10.53	5	0.21
1	79.76	6	0.12
2	7.79	7	0.09
3	0.85	8	0.06
4	0.58	9	0.0

Figure 4: Distribution of degree of ambiguity in training set. The table lists percentage of events that had a specific number N of candidate interpretations constructed from the correct state.

4 Learning pragmatic interpretation

We capture the fate of each interpretation $i_{t,j}$ in a discrete variable F whose value is *correct*, *eliminated*, or *dropped*. We also represent each intention $i_{t,j}$, observation o, and state s_k in terms of features. We seek to learn a function

$$\begin{aligned} P(F = \textit{correct} \quad | & \quad \text{features}(i_{t,j}), \\ & \quad \text{features}(o), \\ & \quad \text{features}(s_k)) \end{aligned}$$

from a set of training examples $E = \{e_1, ..., e_n\}$ where, for $l = 1..n$, we have:

$$\begin{aligned} e_l = (\quad & F = \text{fate}(i_{t,j}), \text{features}(i_{t,j}), \\ & \text{features}(o), \text{features}(s_k)). \end{aligned}$$

We chose to train maximum entropy models (Berger et al., 1996). Our learning framework is described in Section 4.1; the results in Section 4.2.

4.1 Learning setup

We defined a range of potentially useful features, which we list in Figures 5, 6, and 7. These features formalize pragmatic distinctions that plausibly provide evidence of the correct interpretation for a user utterance or action. You might annotate any of these features by hand, but computing them automatically lets us easily explore a much larger range of possibilities. To allow these various kinds of features (integer-valued, binary-valued, and string-valued) to interface to the maximum entropy model, these features were converted into a much broader class of indicator features taking on a value of either 0.0 or 1.0.

feature set	description
NumTacitActions	The number of tacit actions in $i_{t,j}$.
TaskActions	These features represent the action type (function symbol) of each action a_k in $i_{t,j} = \langle A_1 : a_1, A_2 : a_2, ..., A_n : a_n \rangle$, as a string.
ActorDoesTaskAction	For each $A_k : a_k$ in $i_{t,j} = \langle A_1 : a_1, A_2 : a_2, ..., A_n : a_n \rangle$, a feature indicates that A_k (represented as string "Agent" or "User") has performed action a_k (represented as a string action type, as in the TaskActions features).
Presuppositions	If o is an utterance, we include a string representation of each presupposition assigned to o by $i_{t,j}$. The predicate/argument structure is captured in the string, but any gensym identifiers within the string (e.g. `target12`) are replaced with exemplars for that identifier type (e.g. `target`).
Assertions	If o is an utterance, we include a string representation of each dialogue act assigned to o by $i_{t,j}$. Gensym identifiers are filtered as in the Presuppositions features.
Syntax	If o is an utterance, we include a string representation of the bracketed phrase structure of the syntactic analysis assigned to o by $i_{t,j}$. This includes the categories of all non-terminals in the structure.
FlexiTaskIntentionActors	Given $i_{t,j} = \langle A_1 : a_1, A_2 : a_2, ..., A_n : a_n \rangle$, we include a single string feature capturing the actor sequence $\langle A_1, A_2, ..., A_n \rangle$ in $i_{t,j}$ (e.g. "User, Agent, Agent").

Figure 5: The interpretation features, $\text{features}(i_{t,j})$, available for selection in our learned model.

feature set	description
Words	If o is an utterance, we include features that indicate the presence of each word that occurs in the utterance.

Figure 6: The observation features, $\text{features}(o)$, available for selection in our learned model.

feature set	description
NumTasksUnderway	The number of tasks underway in s_k.
TasksUnderway	The name, stack depth, and current task state for each task underway in s_k.
NumRemainingReferents	The number of objects yet to be identified in s_k.
TabulatedFacts	String features representing each proposition in the conversational record in s_k (with filtered gensym identifiers).
CurrentTargetConstraints	String features for each positive and negative constraint on the current target in s_k (with filtered gensym identifiers). E.g. "`positive: squareFigureObject(target)`" or "`negative: solidFigureObject(target)`".
UsefulProperties	String features for each property instantiated in the experiment interface in s_k. E.g. "`squareFigureObject`", "`solidFigureObject`", etc.

Figure 7: The dialogue state features, $\text{features}(s_k)$, available for selection in our learned model.

We used the MALLET maximum entropy classifier (McCallum, 2002) as an off-the-shelf, trainable maximum entropy model. Each run involved two steps. First, we applied MALLET's feature selection algorithm, which incrementally selects features (as well as conjunctions of features) that maximize an exponential gain function which represents the value of the feature in predicting interpretation fates. Based on manual experimentation, we chose to have MALLET select about 300 features for each learned model. In the second step, the selected features were used to train the model to estimate probabilities. We used MALLET's implementation of Limited-Memory BFGS (Nocedal, 1980).

4.2 Evaluation

We are generally interested in whether COREF's experience with previous subjects can be leveraged to improve its interactions with new subjects. Therefore, to evaluate our approach, while making maximal use of our available data set, we performed a hold-one-subject-out cross-validation using our 20 human subjects $H = \{h_1, ..., h_{20}\}$. That is, for each subject h_i, we trained a model on the training examples associated with subjects $H \setminus \{h_i\}$, and then tested the model on the examples associated with subject h_i.

To quantify the performance of the learned model in comparison to our baseline, we adapt the mean reciprocal rank statistic commonly used for evaluation in information retrieval (Vorhees, 1999). We expect that a system will use the probabilities calculated by a disambiguation model to decide which interpretations to pursue and how to follow them up through the most efficient interaction. What matters is not the absolute probability of the correct interpretation but its rank with respect to competing interpretations. Thus, we consider each utterance as a query; the disambiguation model produces a ranked list of responses for this query (candidate interpretations), ordered by probability. We find the rank r of the correct interpretation in this list and measure the outcome of the query as $\frac{1}{r}$. Because of its weak assumptions, our baseline disambiguation model actually leaves many ties. So in fact we must compute an *expected* reciprocal rank (ERR) statistic that averages $\frac{1}{r}$ over all ways of ordering the correct interpretation against competitors of equal probability.

ERR range	Hand-built model	Learned models
1	20.75%	81.76%
$[\frac{1}{2}, 1)$	74.21%	16.35%
$[\frac{1}{3}, \frac{1}{2})$	3.46%	1.26%
$[0, \frac{1}{3})$	1.57%	0.63%
mean(ERR)	0.77	0.92
var(ERR)	0.02	0.03

Figure 8: For the 318 ambiguous sensory events, the distribution of the expected reciprocal of rank of the correct interpretation, for the initial, hand-built model and the learned models in aggregate.

Figure 8 shows a histogram of ERR across the ambiguous utterances from the corpus. The learned models correctly resolve almost 82%, while the baseline model correctly resolves about 21%. In fact, the learned models get much of this improvement by learning weights to break the ties in our baseline model. The overall performance measure for a disambiguation model is the mean expected reciprocal rank across all examples in the corpus. The learned model improves this metric to 0.92 from a baseline of 0.77. The difference is unambiguously significant (Wilcoxon rank sum test $W = 23743.5, p < 10^{-15}$).

4.3 Selected features

Feature selection during training identified a variety of syntactic, semantic, and pragmatic features as useful in disambiguating correct interpretations. Selections were made from every feature set in Figures 5, 6, and 7. It was often possible to identify relevant features as playing a role in successful disambiguation by the learned models. For example, the learned model trained on $H \setminus \{\texttt{c4}\}$ delivered the following probabilities for the two interpretations COREF found for c4's utterance (1b):

$$P(I = i_{2,1} | o, S_2 = s_{8923}) = \quad 0.665$$
$$P(I = i_{2,2} | o, S_2 = s_{8923}) = \quad 0.335$$

The correct interpretation, $i_{2,1}$, hypothesizes that the user means saddlebrown, the darker of the two shades of brown in the display. Among the features selected in this model is a Presuppositions feature (see Figure 5) which is present just in case the word 'brown' is interpreted as meaning saddlebrown rather than some other shade. This feature allows the learned model to prefer to interpret c4's use of 'brown' as meaning this darker shade of brown, based on the observed linguistic behavior of other users.

5 Results in context

Our work adds to a body of research learning deep models of language from evidence implicit in an agent's interactions with its environment. It shares much of its motivation with co-training (Blum and Mitchell, 1998) in improving initial models by leveraging additional data that is easy to obtain. However, as the examples of Section 2.3 illustrate, COREF's interactions with its users offer substantially more information about interpretation than the raw text generally used for co-training. Closer in spirit is AI research on learning vocabulary items by connecting user vocabulary to the agent's perceptual representations at the time of utterance (Oates et al., 2000; Roy and Pentland, 2002; Cohen et al., 2002; Yu and Ballard, 2004; Steels and Belpaeme, 2005). Our framework augments this information about utterance context with additional evidence about meaning from linguistic interaction. In general, dialogue coherence is an important source of evidence for all aspects of language, for both human language learning (Saxton et al., 2005) as well as machine models. For example, Bohus et al. (2008) use users' confirmations of their spoken requests in a multi-modal interface to tune the system's ASR rankings for recognizing subsequent utterances.

Our work to date has a number of limitations. First, although 318 ambiguous interpretations did occur, this user study provided a relatively small number of ambiguous interpretations, in machine learning terms; and most (80.2%) of those that did occur were 2-way ambiguities. A richer domain would require both more data and a generative approach to model-building and search.

Second, this learning experiment has been performed after the fact, and we have not yet investigated the performance of the learned model in a follow-up experiment in which COREF uses the learned model in interactions with its users.

A third limitation lies in the detection of 'correct' interpretations. Our scheme sometimes conflates the user's actual intentions with COREF's subsequent assumptions about them. If COREF decides to strategically drop the user's *actual intended interpretation*, our scheme may mark another interpretation as 'correct'. Alternative approaches may do better at harvesting meaningful examples of correct and incorrect interpretations from an agent's dialogue experience. Our approach also depends on having clear evidence about what an interlocutor has said and whether the system has interpreted it correctly—evidence that is often unavailable with spoken input or information-seeking tasks. Thus, even when spoken language interfaces use probabilistic inference for dialogue management (Williams and Young, 2007), new techniques may be needed to mine their experience for correct interpretations.

6 Conclusion

We have implemented a system COREF that makes productive use of its dialogue experience by learning to rank new interpretations based on features it has historically associated with correct utterance interpretations. We present these results as a proof-of-concept that contribution tracking provides a source of information that an agent can use to improve its statistical interpretation process. Further work is required to scale these techniques to richer dialogue systems, and to understand the best architecture for extracting evidence from an agent's interpretive experience and modeling that evidence for future language use. Nevertheless, we believe that these results showcase how judicious system-building efforts can lead to dialogue capabilities that defuse some of the bottlenecks to learning rich pragmatic interpretation. In particular, a focus on improving our agents' basic abilities to tolerate and resolve ambiguities as a dialogue proceeds may prove to be a valuable technique for improving the overall dialogue competence of the agents we build.

Acknowledgments

This work was sponsored in part by NSF CCF-0541185 and HSD-0624191, and by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed do not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. Thanks to our reviewers, Rich Thomason, David Traum and Jason Williams.

References

Adam L. Berger, Stephen Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100.

Dan Bohus, Xiao Li, Patrick Nguyen, and Geoffrey Zweig. 2008. Learning n-best correction models from implicit user feedback in a multi-modal local search application. In *The 9th SIGdial Workshop on Discourse and Dialogue*.

Susan E. Brennan and Herbert H. Clark. 1996. Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology*, 22(6):1482–1493.

Herbert H. Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. In Philip R. Cohen, Jerry Morgan, and Martha E. Pollack, editors, *Intentions in Communication*, pages 463–493. MIT Press, Cambridge, Massachusetts, 1990.

Paul R. Cohen, Tim Oates, Carole R. Beal, and Niall Adams. 2002. Contentful mental states for robot baby. In *Eighteenth national conference on Artificial intelligence*, pages 126–131, Menlo Park, CA, USA. American Association for Artificial Intelligence.

David DeVault and Matthew Stone. 2007. Managing ambiguities across utterances in dialogue. In *Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue (Decalog 2007)*, pages 49–56.

David DeVault. 2008. *Contribution Tracking: Participating in Task-Oriented Dialogue under Uncertainty*. Ph.D. thesis, Department of Computer Science, Rutgers, The State University of New Jersey, New Brunswick, NJ.

Barbara Di Eugenio, Pamela W. Jordan, Richmond H. Thomason, and Johanna D. Moore. 2000. The agreement process: An empirical investigation of human-human computer-mediated collaborative dialogue. *International Journal of Human-Computer Studies*, 53:1017–1076.

Darren Gergle, Carolyn P. Rosé, and Robert E. Kraut. 2007. Modeling the impact of shared visual information on collaborative reference. In *CHI 2007 Proceedings*, pages 1543–1552.

Patrick G. T. Healey and Greg J. Mills. 2006. Participation, precedence and co-ordination in dialogue. In *Proceedings of Cognitive Science*, pages 1470–1475.

Jerry R. Hobbs, Mark Stickel, Douglas Appelt, and Paul Martin. 1993. Interpretation as abduction. *Artificial Intelligence*, 63:69–142.

Pamela W. Jordan and Marilyn A. Walker. 2005. Learning content selection rules for generating object descriptions in dialogue. *JAIR*, 24:157–194.

Andrew McCallum. 2002. MALLET: A MAchine learning for LanguagE toolkit. http://mallet.cs.umass.edu.

Jorge Nocedal. 1980. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782.

Tim Oates, Zachary Eyler-Walker, and Paul R. Cohen. 2000. Toward natural language interfaces for robotic agents. In *Proc. Agents*, pages 227–228.

Massimo Poesio and Ron Artstein. 2005. Annotating (anaphoric) ambiguity. In *Proceedings of the Corpus Linguistics Conference*.

Massimo Poesio and Renata Vieira. 1998. A corpus-based investigation of definite description use. *Computational Linguistics*, 24(2):183–216.

Deb Roy and Alex Pentland. 2002. Learning words from sights and sounds: A computational model. *Cognitive Science*, 26(1):113–146.

Matthew Saxton, Carmel Houston-Price, and Natasha Dawson. 2005. The prompt hypothesis: clarification requests as corrective input for grammatical errors. *Applied Psycholinguistics*, 26(3):393–414.

David Schlangen and Raquel Fernández. 2007. Speaking through a noisy channel: Experiments on inducing clarification behaviour in human–human dialogue. In *Proceedings of Interspeech 2007*.

Luc Steels and Tony Belpaeme. 2005. Coordinating perceptually grounded categories through language. a case study for colour. *Behavioral and Brain Sciences*, 28(4):469–529.

Richmond H. Thomason, Matthew Stone, and David DeVault. 2006. Enlightened update: A computational architecture for presupposition and other pragmatic phenomena. For the Ohio State Pragmatics Initiative, 2006, available at http://www.research.rutgers.edu/˜ddevault/.

Ellen M. Vorhees. 1999. The TREC-8 question answering track report. In *Proceedings of the 8th Text Retrieval Conference*, pages 77–82.

Jason Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2):393–422.

Chen Yu and Dana H. Ballard. 2004. A multimodal learning interface for grounding spoken language in sensory perceptions. *ACM Transactions on Applied Perception*, 1:57–80.