Parsing Natural Language Conversations using Contextual Cues

Shashank Srivastava
Carnegie Mellon University
ssrivastava@cmu.edu

Amos Azaria
Ariel University
amos.azaria@ariel.ac.il

Tom Mitchell
Carnegie Mellon University
tom.mitchell@cmu.edu

Abstract

In this work, we focus on semantic parsing of natural language conversations. Most existing methods for semantic parsing are based on understanding the semantics of a single sentence at a time. However, understanding conversations also requires an understanding of conversational context and discourse structure across sentences. We formulate semantic parsing of conversations as a structured prediction task, incorporating structural features that model the 'flow of discourse' across sequences of utterances. We create a dataset for semantic parsing of conversations, consisting of 113 real-life sequences of interactions of human users with an automated email assistant. The data contains 4759 natural language statements paired with annotated logical forms. Our approach yields significant gains in performance over traditional semantic parsing.

1 Introduction

The ability to interact with computers using natural language is a fundamental challenge towards building intelligent cognitive agents. The problem of language understanding is usually approached using semantic parsing, a growing area within the field of natural language processing (NLP). Semantic parsing is the conversion of natural language utterances to formal semantic representations called *logical forms* that machines can execute. For example, a natural language sentence like *'Set the subject of the mail as hello'* may be mapped to a logical form such as `setFieldValue( subject, stringVal('hello'))`, which can then be executed to yield the desired behavior or output. Similarly, in the domain of mathematics, *'What is the product of three and five?'* may be mapped to a logical form such as `multiply( 3, 5)`[1]. Semantic parsing has previously been explored in an eclectic variety of settings such as querying from databases [Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005; Berant *et al.*, 2013], robot navigation [Kate *et al.*, 2005] and spreadsheet manipulation [Gulwani and Marron, 2014].

[1]Predicates used in *logical forms* (such as `stringVal` and `multiply`) come from domain-specific meaning representations

```
[U1] User:  Compose an email
[A1] Agent:  Composing new email.  Its fields
are:  recipient, subject and body
[U2] User:  Use mom's email
[A2] Agent:  I set the recipient field to
'momthebest@email.com'
[U3] User:  Set the subject to hello, please
[A3] Agent:  I set the subject field to 'hello,
please'
[U4] User:  Set the subject to hello, please
[A4] Agent:  I set the subject field to 'hello'
[U5] User:  Set the body to I'll be late
[A5] Agent:  I set the body field to 'I'll be
late'
[U6] User:  Send the email
[A6] Agent:  I sent the email ...
```

Figure 1: Example of a real-world interaction between a human (User) and an automated email assistant (Agent).

However, most methods for semantic parsing have focused on parsing single natural language sentences in isolation. While this is useful for environments consisting of one-shot interactions of users with a system (e.g., running QA queries on a database), many settings of human-computer interactions require extended interactions between a user and an automated assistant (e.g. making a flight booking). This makes the one-shot parsing model inadequate for many scenarios.

Figure 1 shows a snippet of a conversation between a human user and a digital email assistant. We note that pragmatics and conversational context offer essential cues in understanding several individual utterances from the user. In particular, observe that utterance U2 (*'Set to mom's email'*) cannot be correctly parsed based on its content alone, but requires an understanding of the discourse. Based on the previous statement (composing a new email), setting the recipient field is the user's likely intent. Similarly, utterances U3 and U4 show an example of a repetition, where the agent first misinterprets (U3), and then correctly parses a statement (U4). While parsing U4, the agent needs to implicitly understand that it should interpret the current utterance to a different logical form than before (even though the textual content is identical). This would not be possible in the one-shot parsing setting, which cannot incorporate such implicit feedback. Instead, correctly interpreting the sentence requires modeling of the discourse structure of the conversation.

We address the problem of semantic parsing of natural language conversations. The underlying thesis is that modeling discourse structure and conversational context should assist interpretation of language [Van Dijk, 1980]. Here, we focus on incorporating structural cues for modeling these discourse structures and context. However, we do not address issues such as anaphora and discourse referents [Kamp and Reyle, 1993] that are prevalent in conversations, but lie beyond the scope of the current work. Our main contributions are:

- We address semantic parsing in the context of conversations; and provide an annotated dataset of conversations, comprising of 4759 natural language statements with their associated logical forms.
- We formulate the problem as a structured prediction task, and introduce a latent variable model that incorporates both text-based cues within sentences, and structural inferences across sentences. Using this model, we empirically demonstrate significant improvements in parsing conversations over the state-of-the-art.
- We also present effective heuristic strategies for expanding the search space of allowable meaning representations to improve semantic parsing of conversations.
- We show that latent categories learned by the model are semantically meaningful, and can be interpreted in terms of discourse states in the conversations.

2 Related Work

Supervised semantic parsing has been studied in a wide range of settings [Zettlemoyer and Collins, 2005; Wong and Mooney, 2007; Kwiatkowski *et al.*, 2010]. Recent approaches have focused on various strategies for using weaker forms of supervision [Clarke *et al.*, 2010; Krishnamurthy and Mitchell, 2012; Berant *et al.*, 2013] and rapid prototyping of semantic parsers for new domains [Wang *et al.*, 2015; Pasupat and Liang, 2015]. Other works have explored semantic parsing in a grounded contexts, and using perceptual context to assist semantic parsing [Matuszek *et al.*, 2012; Krishnamurthy and Kollar, 2013]. However, none of these approaches incorporate conversational context to jointly interpret a conversational sequence. For instance, poor interpretations made while parsing at a point in a conversation cannot be re-evaluated in light of more incoming information. A notable work that incorporates conversational data in relation to semantic parsing is [Artzi and Zettlemoyer, 2011]. However, rather that incorporating contextual cues, their goal is very different: using rephrasings in conversation logs as weak supervision for inducing a semantic parser. Closer to our work is previous work by [Zettlemoyer and Collins, 2009], who also learn context-sensitive interpretations of sentences using a two-step model. However, their formulation is specific to CCG grammars and focuses on modeling discourse referents.

The role of context in assigning meaning to language has been emphasized from abstract perspectives in computational semantics [Bates, 1976; Van Dijk, 1980], as well as in systems for task-specific applications [Larsson and Traum, 2000]. Examples of the former include analyzing language from perspectives of speech acts [Searle, 1969] and semantic scripts [Schank and Abelson, 1977; Chambers and Jurafsky, 2008; Pichotta and Mooney, 2015]. These works induce typical trajectories of event sequences from unlabeled text to infer what might happen next.

On the other hand, a notable application area that has explored conversational context within highly specific settings is state tracking in dialog systems. Here, the focus is on inferring the state of a conversation given all previous dialog history [Higashinaka *et al.*, 2003; Williams *et al.*, 2013] in context of specific task trajectories, rather than interpreting the semantic meanings of individual utterances.

In terms of approach, while our formulation is largely agnostic to the choice of semantic parsing framework, for this work our method is based on CCG semantic parsing, which is a popular semantic parsing approach [Zettlemoyer and Collins, 2007; Kwiatkowski *et al.*, 2013; Artzi *et al.*, 2015]. The CCG grammar formalism [Steedman and Baldridge, 2011] has been widely used for its explicit pairing of syntax with semantics, and allows expression of long range dependencies extending beyond context-free-grammars. We also allow our semantic parser to output logical forms that may not be entailed from an utterance using a strict grammar formalism, expanding on similar ideas in [Wang *et al.*, 2015; Goldwasser and Roth, 2014]. Finally, some of the heuristics proposed in this paper for improving parsing performance are motivated by previous work on paraphrasing for semantic parsing [Berant and Liang, 2014].

3 Semantic Parsing with Conversational Context

We present an approach for semantic parsing of conversations by posing conversations as sequences of utterances to model 'flow of discourse'. We consider the problem as a structured prediction task, where we jointly learn preferences for collective assignments of logical forms for sentences in a sequence.

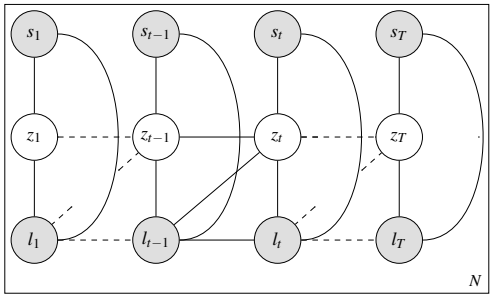

Figure 2: Model diagram for semantic parsing of conversational sequences. Traditional semantic parsing features depend on utterances s_t and associated logical forms l_t only. Our model additionally allows structured features that can depend on previous logical forms l_{t-1}, latent variables z_t representing the discourse state of the conversation at any step, and the previous utterances $s_1 \dots s_t$.

Let $\mathbf{s}$ denote a conversation sequence of T utterances by a user, with individual utterances denoted as $\{s_1 \dots s_T\}$. Let $\mathbf{l} := \{l_1 \dots l_T\}$ be the intended logical forms for cor-

responding utterances. We assume a supervised learning setting where we have labeled training sequences $\mathcal{T} := \{(\mathbf{s}^{(1)}, \mathbf{l}^{(1)}) \dots (\mathbf{s}^{(N)}, \mathbf{l}^{(N)})\}$ consisting of utterances and their associated logical forms. In comparison, the traditional supervised setting for learning semantic parsers consists of pairs of training utterances and associated logical forms (s_i, l_i), but doesn't have a sequential structure. Our model utilizes this sequential structure to incorporate information about discourse and pragmatics.

In addition, we also associate a latent categorical variable denoted as z_t with each user utterance s_t to reinforce the modeling of the flow of discourse (see Figure 2). The latent states can take one of K possible discrete values, and abstractly represent distinct discourse states. The value of K is a predefined parameter for the model[2]. In Section 6, we show that these latent states learn distinct interpretable discourse states that are prevalent in conversations, and can support dynamic modeling of context with the progress of a conversation.

For a given utterance sequence $\mathbf{s} = \{s_1 \dots s_T\}$, our model predicts logical assignments, $\hat{\mathbf{l}} = \{\hat{l}_1 \dots \hat{l}_T\}$, and latent discourse states, $\hat{\mathbf{z}} = \{\hat{z}_1 \dots \hat{z}_T\}$ by solving the following inference problem, i.e. finding the highest scoring assignment of logical forms, $\mathbf{l}$, and discourse states, $\mathbf{z}$, under a given model:

$$(\hat{\mathbf{l}}, \hat{\mathbf{z}}) = \underset{\boldsymbol{l} \in \mathcal{L}(\mathbf{s}), \mathbf{z}}{\operatorname{argmax}} \quad S_w(\mathbf{s}, \mathbf{l}, \mathbf{z}) \tag{1}$$

Here, $\mathcal{L}(\mathbf{s})$ is the search space associated with sequence $\mathbf{s}$, consisting of possible joint assignments of logical forms to the various utterances in the sequence[3], and the hat symbol ($\hat{}$) represents predicted variables. $S_w(\mathbf{s}, \mathbf{l}, \mathbf{z})$ represents a linear score denoting the goodness of an assignment of logical forms, $\mathbf{l}$, and latent discourse states, $\mathbf{z}$, to utterances in conversation sequence $\mathbf{s}$. This score is defined as:

$$S_w(\mathbf{s}, \mathbf{l}, \mathbf{z}) = w^T \phi(\mathbf{s}, \mathbf{l}, \mathbf{z})$$

where, ϕ is a feature function that produces a real-valued feature vector for the tuple $(\mathbf{s}, \mathbf{l}, \mathbf{z})$. As we shall see in Section 4, these features consist of two categories $\phi = [\phi_{text}\ \phi_{context}]$: (a) ϕ_{text}: features for individual parses that model how well individual logical forms, l_t, match corresponding natural language utterances, s_t, in isolation. This subset subsumes all features from traditional semantic parsing models (b) $\phi_{context}$: features that model conversational context and discourse across the chain structure of the conversation. The model parameters, w, consist of a real-valued weight for each kind of feature, and are learned during training.

Learning: The model parameters w can be trained via the latent variable Structured Perceptron algorithm [Collins, 2002; Zettlemoyer and Collins, 2007], which performs subgradient updates minimizing the following structured hinge loss:

$$L(w, s, l, z) := \Big| \max_{\hat{\mathbf{l}} \in \mathcal{L}(\mathbf{s}), \hat{\mathbf{z}}} S_w(\mathbf{s}, \hat{\mathbf{l}}, \hat{\mathbf{z}}) - \max_{\mathbf{z}^\star} S_w(\mathbf{s}, \mathbf{l}, \mathbf{z}^\star) \Big|_+ \tag{2}$$

The objective consists of a difference of two terms: the first is the score of predicted assignment $(\hat{\mathbf{l}}, \hat{\mathbf{z}})$ for sequence $\mathbf{s}$ under the current model, while the second is the score of the highest-scoring latent discourse states for $\mathbf{s}$ with the ground truth logical form $\mathbf{l}$. These correspond to solving the following inference problems: (1) finding the best combination of logical forms and discourse states for a sequence (Equation 1), and (2) finding the best combination of discourse states for a sequence and given logical forms. We describe the procedure to solve Equation 1 below. The second inference problem is a simpler case of the same equation, since one of the two variables to be inferred ($\mathbf{l}$) is already known. Finally, in our experiments, we also use an $l2$-regularizer (with ridge parameter 0.01) for weight-vector w.

We note that our formulation does not pre-suppose a specific semantic parsing framework, grammar or feature-definition. In this work, we use a CCG-based semantic parsing approach. However, our framework can seamlessly extend to other formalisms such as DCS [Liang *et al.*, 2013] that are trained with gradient updates.

Inference: Both training and prediction for the model depend on efficiently solving the inference problem in Equation 1. In general, the problem can be tractably solved if components of feature function ϕ decompose into smaller factors. In our case, ϕ_{text} features decompose according to the structure of individual parse trees. Similarly, $\phi_{context}$ features factorize according to chain structure of the discourse due to Markov properties (details in Section 4). Our inference procedure consists of a hierarchical two step process: (1) we find a candidate set of possible logical forms for each utterance, s_t, in a sequence, and (2) we find the best joint assignment among these by incorporating information from contextual and structural cues. We now briefly describe the two steps.

In the first step, we obtain a set of candidate logical forms, $\mathbb{L}(s_t)$, for individual utterances, s_t, while viewing them in isolation. For this, we score a potential logical form using only the text-based features (ϕ_{text}). This is identical to traditional semantic parsing, and the highest scoring logical forms for an utterance can be found using the k-best CYK algorithm. In practice, considerations such as large grammars make exact inference prohibitive for this setting. Following previous works in semantic parsing [Kwiatkowski *et al.*, 2013; Berant *et al.*, 2013], we employ beam search to find an approximate set of best candidate parses for a sentence.

In the next step, we combine the various $\mathbb{L}(s_t)$ to infer the best joint semantic parse $\mathbf{l}$ (and discourse states $\mathbf{z}$) for the complete sequence, $\mathbf{s}$. This involves obtaining a sequence of l_t's that incorporates scores from the contextual and discourse features ($\phi_{context}$). Since these features decompose according to the chain structure of the sequence, the highest scoring assignments of a sequence of discourse states $\mathbf{z}$ and logical

[2]Setting $K = 1$ effectively reduces the model to not using latent variables at all. This model still incorporates structural context and discourse by learning preferences for joint assignments of logical forms to utterances. However, the latent variables z_t afford additional flexibility to the model. i.e. for the same context, the model can behave differently based on the current state

[3]For any utterance s_t, the grammar of the meaning representation formalism can specify the set $\mathbb{L}(s_t)$ of its candidate logical forms. The associated search space for the sequence $\mathbf{s}$ is then simply given by the cross-product $\mathcal{L}(\mathbf{s}) = \otimes_t \mathbb{L}(s_t)$

forms $\mathbf{l}$ can be efficiently computed using the Viterbi algorithm (where *hidden* states of the Viterbi chart correspond to pairs of logical forms in the candidate set and discrete discourse states).

Expansion strategies: The approach described above uses the traditional semantic parsing setting (using the score from ϕ_{text} only) to define the set of possible logical forms for an utterance, $\mathbb{L}(s_t)$. However, one may define strategies to expand the candidate set $\mathbb{L}(s_t)$ to also include logical forms that are not included in the beam from the text-only features. We consider the following simple heuristics to expand the candidate set, $\mathbb{L}(s_t)$, for each utterance s_t:

1. **Highest PMI (PMI)**: Add to $\mathbb{L}(s_t)$ the logical forms that have the n-highest PMI with the best scoring logical form from the text-only model for the previous utterance (s_{t-1}) in the training set.
2. **Highest conditional probability (Prob)**: Add to $\mathbb{L}(s_t)$ the n most frequent logical forms that followed the best scoring logical form from the text-only model for the previous utterance (s_{t-1}) in the training set.
3. **Paraphrase (PP)**: Add to $\mathbb{L}(s_t)$ the candidate sets for k utterances in the training set that are semantically most similar to s_t. For computing similarity, we use Vector Tree Kernels [Srivastava *et al.*, 2013] that provide a semantic similarity score between two sentences using syntactic information and distributional embeddings.
4. **Most frequent (MF)**: Add the n-most frequent logical forms observed in the training data to the candidate set $\mathbb{L}(s_t)$ for each utterance.

Prediction: Our model is trained to simultaneously consider all utterances of a sequence. At prediction time however, in most scenarios, the agent would need to continually infer the parse of the latest sentence during the conversation, in a real-time setting. In particular, we need to ensure not to use future utterances while predicting the logical form for the current utterance. Hence, at prediction time, we simply use the model to predict logical forms for the sequence of conversation ending at the current utterance.

4 Features

In this section, we outline the text-based and context-based features used by our model. The text-based features are based on lexical and grammar rules, typically employed in traditional semantic parsers, whereas the context-based features are based on simple counts of specific configurations of logical predicates and discourse state assignments for a sequence. Thus, both kinds of features can be computed efficiently.

Simple text-based features ϕ_{text}: These depend on a single utterance s_t and a candidate logical form l_t. As such, they lie in the ambit of traditional semantic parsing features, and are standard features for CCG semantic parsers [Zettlemoyer and Collins, 2007; Azaria *et al.*, 2016; Artzi and Zettlemoyer, 2013]. We use the following text-based features:

T1: Lexicon features: Indicator features for each lexicon entry that fires for the given utterance, and indicator features for each syntactic category (POS) in the utterance combined with the logical form.

T2: Rule application features: Indicator features for both unary and binary rules in the parse of the given utterance to the associated logical form.

T3: String-based features: Number of words in the utterance, indicator features denoting whether string spans occur at the beginning or end of the utterance.

Structural context-based features $\phi_{context}$: These features model the flow of discourse and context-specific regularities by learning preferences for correlations between logical predicates, discourse state assignments and features based on the text of conversational history.

C1: Transition features: Indicator features denoting combinations of logical predicates in successive utterances (e.g., $\{L_i$:`setBodyToString`, L_{i-1}:`createEmail`$\}$)[4], combinations of discourse variable assignments in successive utterances (e.g., $\{Z_i$=`State1`, Z_{i-1}=`State2`$\}$), combinations of logical predicates and discourse variable in successive steps.

C2: Emission features: Indicator features denoting presence of a logical predicate in the current utterance combined with the current discourse state (e.g. $\{Z_i$=`State0`, L_i:`greeting`$\}$).

C3: Lexical trigger features: Indicator features denoting presence of trigger words (from CCG lexicon) in the current utterance, paired with logical predicates in the current logical form and the current discourse state.

C4: Repetition features: Indicator features denoting whether the current utterance is a repeat, Indicator features denoting if the current utterance is a repeat and the current logical form is the same as the previous step, etc.[5]

C5: Domain-specific features: Indicator feature that looks at long term history to denote whether the user is currently teaching the system a new procedure (e.g., `inProcedure=true`). See Section 5 for explanation.

C6: Positional features: Feature denoting the position of the current utterance in the conversation.

C7: Provenance features: Indicator feature denoting whether the current logical form was derived from the CCG grammar, or added through an expansion strategy.

5 Dataset

Most existing datasets for semantic parsing focus on understanding single utterances at a time, rather than conversations. We created a dataset of real-life user conversations in an email assistant environment. For this, we annotated raw transcripts of interactions between human subjects and an email assistant agent in a text-dialog environment provided in previous work by [Azaria *et al.*, 2016].

Each interaction session consists of the user trying to accomplish a set of email-based tasks by interacting with the agent (refer to Figure 1 for a simple example). The system also allows users to teach new procedures (e.g., forwarding an email), concepts (e.g., concept of a contact with

[4] $\{L_i$:`a`, L_j:`b`$\}$ denotes that the logical form L_i includes the logical predicate `a`, and L_j contains `b`.

[5] While Figure 2 indicates possible edge features between latent variables (z_t or l_t) and s_t , the model also allows features that could depend on the entire history of utterances observed till t ($s_1 \ldots s_t$).

Number of user utterances	4759
User sessions	113
Avg length of session (utterances)	42
Word types (all utterances)	704

Table 1: Corpus statistics for the Email Assistant dataset

fields name, phone number, etc.), and instances (e.g., instantiating a contact) on-the-fly. Because of this feature, and since the users are unaware of the capabilities of the agent, linguistic usage in the experiments is complex and diverse, compared to many existing datasets. The data consists of sequences of user utterances and system responses. In order to make the data usable for research, we pruned conversation sequences and annotated user utterances with their associated logical forms. e.g., the utterance: *'What is Mom's email?'* is annotated with the logical form `(evalField (getFieldByInstanceName mom email))`, following the logical language in the original paper [Azaria *et al.*, 2016]. Utterances that could not be reasonably expected to be interpreted by the email agent were marked as `unknownCommand` (7% of utterances). However, if the user later taught the system what she meant, future instances of the utterance were marked with the intended logical form (e.g., users often taught the command *'Next'* to read and move to the next email in the inbox). Sequences devolving into non-meaningful interactions were removed, e.g., if the annotator deemed that the user did not intend to complete a task. Superfluous segments of the original conversation (e.g., utterances re-phrasing a previous utterances that the system didn't process) were also manually pruned.

Annotating every command by manually specifying its logical form would require experience with the underlying logical language of the system. Instead, we developed a software that allowed faster annotation using an alternate procedure. The software allows annotators to load a conversation sequence, and execute each utterance against a live version of the email agent. If the response indicates that agent has correctly interpreted the command, the annotator may save the associated logical form for the utterance. However, if the response indicates that the system did not interpret the command correctly (judged by the annotator's belief of the user's intent), the annotator may provide a command which (i) reflects the intention of the utterance, and that (ii) the email agent can interpret correctly. In effect, the strategy uses annotators to paraphrase the original command into simpler commands (still in natural language) that the agent would parse to the correct logical form, without exposing them to the underlying meaning representation formalism.

Figure 1 summarizes the statistics for the curated dataset. For evaluation and future comparisons, we split the data into a training fold (93 conversation sequences) and a test fold (20 conversation sequences).

6 Evaluation and Analysis

In this section, we discuss quantitative and qualitative evaluation of our method. We first make a comparative evaluation of our method in recovering gold-standard annotation parses on the held-out test set. Next, we make an ablation study to assess the contributions of different families of structural features described in Section 4. We then briefly analyze the characteristics of the latent states learned by the model, and qualitatively discuss the performance of the model.

Parsing performance: For training our models, we tune parameters, i.e. number of training epochs (5), and the number of clusters ($K = 3$) through 10-fold cross-validation on the training data. For the CCG grammar, we use the PAL lexicon induction algorithm [Krishnamurthy, 2016] to expand the base lexicon provided by [Azaria *et al.*, 2016]. Our baselines include the following semantic parsing models:

- *Unstructured CCG:* CCG parser from [Azaria *et al.*, 2016], which uses the same lexicon and text-based features from Section 4, but does not incorporate structural features
- *Seq2Seq:* Deep neural network based on sequence-to-sequence RNN model from [Bahdanau *et al.*, 2015] to directly maps utterances to logical forms.
- *LEX:* Alignment-based model that chooses best parse using lexical trigger scores only, with no syntactic rules (does not use provided lexicon).

Table 2 compares the performance of variations of our method for semantic parsing with conversational context (SPCon) with baselines on the held-out test set of conversational sequences. Parses are evaluated for exact match in logical forms, and reported results are averages over 5 runs. We observe that the *Seq2Seq* and *Unstructured CCG* models perform comparably, whereas *LEX* doesn't perform as well.

We find that our structured models (*SPCon* and its variations) consistently outperform the baseline models. Further, the expansion strategies suggested in Section 3 lead to consistent gains in performance. The improvement of *SPCon* over *Unstructured CCG*, and further improvements of expanded models over *SPCon* are statistically significant ($\alpha = 0.1$, McNemar's test). In particular, the expansion strategies that afford highest coverage (MF and PP) prove to be most effective. This suggests that the structural features are helpful for disambiguating between a large number of candidate logical forms, even when text-only features don't yield the correct logical form as a candidate. This also indicates that the performance of the unstructured CCG parser is restricted by issues of recall (the correct parse is not

	Accuracy
Previous methods	
Unstructured CCG	51.9
Seq2Seq	52.3
LEX	46.4
Proposed models	
SPCon	54.2
SPCon + PMI	56.2
SPCon + Prob	56.9
SPCon + MF	**62.3**
SPCon + PP	59.8

Table 2: Test accuracies on Email Assistant dataset

among the candidates from the beam search for the majority of error cases) due to the open-ended linguistic usage in the dataset. The expansion strategies partially alleviate this issue.

Feature ablation: Next, we perform an ablation study to analyze the contributions of various kinds of features to the model performance. Figure 3 shows the effects of successively removing different categories of structural features from the best performing model described above.

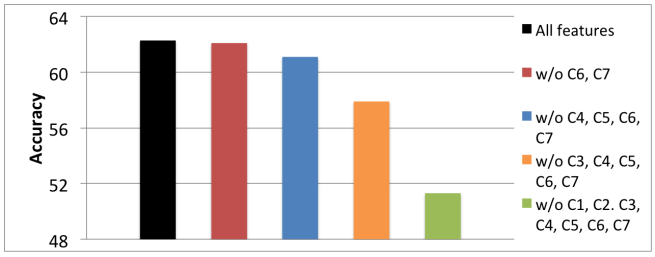

Figure 3: Comparison of parsing accuracy by successive removal of structured feature families described in Section 4. Removing structural transition and emission features (C1 and C2) leads to the most significant drop in performance.

We note that removing positional and provenance features (C6 and C7) has minimal effect on model performance. Removing features identifying repetition and domain-specific features (C4 and C5) leads to a 1% drop. Further removing lexical trigger features (C3) that associate certain words with specific logical predicates and discourse states leads to a more significant drop. However, the biggest effect is seen by removing transition and emission features (C1 and C2). This is expected since these are fundamental for modeling associations across steps in the sequential structure of conversations. Ablating these features in the final step reduces the model to text-based features only, and the model performance is then understandably close to the performance of the unstructured CCG model (the marginal difference is due to batch updates for sentences in a conversation sequence vs online updates in the unstructured case). This also validates our thesis that incorporating contextual information leads to better models for understanding conversations.

Latent states: We investigate the effect of latent states on model performance. We varied the number of latent states (K) and found model performance deteriorated for more than $K = 3$ states (see Figure 4). We qualitatively explored contents of individual latent states to see if learned latent states reflect distinct discourse states in conversations. Table 3 characterizes some of the highest weighted features

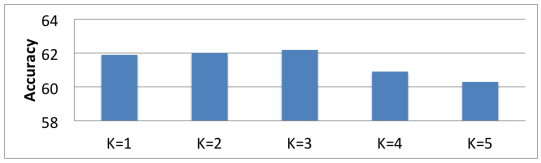

Figure 4: Parsing accuracy for different values of K

associated with each state for a run of the model. State 1 appears to be associated with confusion as it has highest weights for features that indicate presence of logical predicates `unknownCommand` and `cancel`. Similarly, State 2 is associated with teaching new procedures (the feature `inProcedure=true` has a high weight, and another high-weighted feature indicates presence of the logical predicate `doSeq` which is strongly associated with teaching procedures). On the other hand, State 3 has a more generic character, consisting of the most common logical predicates, and is the predicted state for the majority of utterances.

We also observe that latent states enable our approach to model interesting context-specific linguistic usage. For example, an analysis of state-specific weights learned by the model showed that it learns two distinct interpretations for the trigger word *cancel* in different contexts: within a learning procedure *cancel* is strongly associated with a logical predicate to quit the procedure, outside this context it is strongly associated with undoing the action taken in the previous step.

State 1	has(`unknownCommand`), has(`cancel`)
State 2	`inProcedure=true`, has(`doSeq`)
State 3	has(`createInstanceByName`), has(`readInstance`)

Table 3: High-weight features for each latent state ($K = 3$). has(`a`) denotes a feature associated with the presence of logical predicate `a`.

Errors: We found that in many examples, structured features partially address many of the issues highlighted in Figure 1. A manual inspection of errors revealed that a significant number (about 20%) of them are due to user-specific procedures that were taught to the agent by human users during the original study. These errors are too hard to resolve with a batch training approach, and would require incorporating user-specific behavior in the semantic parsing.

7 Conclusion

In this paper, we introduce the problem of semantic parsing of conversations. We present a conceptually simple structured prediction formulation that incorporates conversational context by leveraging structural regularities in conversation sequences. This enables joint learning of text-based features traditionally used by semantic parsers, as well as structural features to model the flow of discourse. The current work uses a simple model of discourse as statistical regularities (with Markov properties) in sequential structure of conversations. This can be refined by incorporating models of discourse entities and discourse referents from discourse representation theory. Understanding of conversations can also be enhanced by models incorporating background world knowledge. Finally, the idea of using conversational context can be generalized to incorporate other modes of contextual information, such as from the agent's execution environment.

Acknowledgments

This work was supported in part by the Yahoo! InMind project and by the Samsung GRO program.

References

[Artzi and Zettlemoyer, 2011] Yoav Artzi and Luke Zettlemoyer. Bootstrapping semantic parsers from conversations. In *EMNLP*, pages 421–432, 2011.

[Artzi and Zettlemoyer, 2013] Yoav Artzi and Luke Zettlemoyer. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, 1:49–62, 2013.

[Artzi *et al.*, 2015] Yoav Artzi, Kenton Lee, and Luke Zettlemoyer. Broad-coverage ccg semantic parsing with AMR. In *EMNLP*, pages 1699–1710, 2015.

[Azaria *et al.*, 2016] Amos Azaria, Jayant Krishnamurthy, and Tom M Mitchell. Instructable intelligent personal agent. In *AAAI*, volume 4, 2016.

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[Bates, 1976] Elizabeth Bates. *Language and context : the acquisition of pragmatics*. Academic Press New York, 1976.

[Berant and Liang, 2014] Jonathan Berant and Percy Liang. Semantic parsing via paraphrasing. In *ACL*, 2014.

[Berant *et al.*, 2013] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *EMNLP*, volume 2, page 6, 2013.

[Chambers and Jurafsky, 2008] Nathanael Chambers and Daniel Jurafsky. Unsupervised learning of narrative event chains. In *ACL*, volume 94305, pages 789–797, 2008.

[Clarke *et al.*, 2010] James Clarke, Dan Goldwasser, Ming-Wei Chang, and Dan Roth. Driving semantic parsing from the world's response. In *CoNLL*, pages 18–27. Association for Computational Linguistics, 2010.

[Collins, 2002] Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *EMNLP*, pages 1–8, 2002.

[Goldwasser and Roth, 2014] Dan Goldwasser and Dan Roth. Learning from natural instructions. *Machine learning*, 94(2):205–232, 2014.

[Gulwani and Marron, 2014] Sumit Gulwani and Mark Marron. Nlyze: Interactive programming by natural language for spreadsheet data analysis and manipulation. In *SIGMOD*, 2014.

[Higashinaka *et al.*, 2003] Ryuichiro Higashinaka, Mikio Nakano, and Kiyoaki Aikawa. Corpus-based discourse understanding in spoken dialogue systems. In *ACL*, pages 240–247, 2003.

[Kamp and Reyle, 1993] Hans Kamp and Uwe Reyle. *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Studies in Linguistics and Philosophy. Springer, 1993.

[Kate *et al.*, 2005] Rohit J Kate, Yuk Wah Wong, and Raymond J Mooney. Learning to transform natural to formal languages. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 1062, 2005.

[Krishnamurthy and Kollar, 2013] Jayant Krishnamurthy and Thomas Kollar. Jointly learning to parse and perceive: Connecting natural language to the physical world. *Transactions of Association for Computational Linguistics*, 2013.

[Krishnamurthy and Mitchell, 2012] Jayant Krishnamurthy and Tom M Mitchell. Weakly supervised training of semantic parsers. In *EMNLP-CoNLL*, pages 754–765, 2012.

[Krishnamurthy, 2016] Jayant Krishnamurthy. Probabilistic models for learning a semantic parser lexicon. In *NAACL-HLT*, pages 606–616, 2016.

[Kwiatkowski *et al.*, 2010] Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. Inducing probabilistic ccg grammars from logical form with higher-order unification. In *EMNLP*, pages 1223–1233, 2010.

[Kwiatkowski *et al.*, 2013] Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke S Zettlemoyer. Scaling semantic parsers with on-the-fly ontology matching. In *ACL*, 2013.

[Larsson and Traum, 2000] Staffan Larsson and David R. Traum. Information state and dialogue management in the trindi dialogue move engine toolkit. *Nat. Lang. Eng.*, pages 323–340, 2000.

[Liang *et al.*, 2013] Percy Liang, Michael I Jordan, and Dan Klein. Learning dependency-based compositional semantics. *Computational Linguistics*, 39:389–446, 2013.

[Matuszek *et al.*, 2012] Cynthia Matuszek, Nicholas FitzGerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. A joint model of language and perception for grounded attribute learning. *arXiv preprint arXiv:1206.6423*, 2012.

[Pasupat and Liang, 2015] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. *arXiv preprint arXiv:1508.00305*, 2015.

[Pichotta and Mooney, 2015] Karl Pichotta and Raymond J Mooney. Learning statistical scripts with LSTM recurrent neural networks. In *AAAI*, 2015.

[Schank and Abelson, 1977] Roger C Schank and Robert P Abelson. Scripts, plans, goals and understanding: an inquiry into human knowledge structures. *Erlbaum*, 1977.

[Searle, 1969] John R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, 1969.

[Srivastava *et al.*, 2013] Shashank Srivastava, Dirk Hovy, and Eduard H Hovy. A walk-based semantically enriched tree kernel over distributed word representations. In *EMNLP*, 2013.

[Steedman and Baldridge, 2011] Mark Steedman and Jason Baldridge. Combinatory categorial grammar, 2011.

[Van Dijk, 1980] Teun Adrianus Van Dijk. Text and context: Explorations in the semantics and pragmatics of discourse. *Nordic Journal of Linguistics*, 2, 1980.

[Wang *et al.*, 2015] Yushi Wang, Jonathan Berant, and Percy Liang. Building a semantic parser overnight. In *ACL*, 2015.

[Williams *et al.*, 2013] Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. The dialog state tracking challenge. In *SIGDIAL*, pages 404–413, 2013.

[Wong and Mooney, 2007] Yuk Wah Wong and Raymond J Mooney. Learning synchronous grammars for semantic parsing with lambda calculus. In *ACL*, volume 45, page 960, 2007.

[Zelle and Mooney, 1996] John M Zelle and Raymond J Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence*, pages 1050–1055, 1996.

[Zettlemoyer and Collins, 2005] Luke S. Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI '05*, pages 658–666, 2005.

[Zettlemoyer and Collins, 2007] Luke S Zettlemoyer and Michael Collins. Online learning of relaxed ccg grammars for parsing to logical form. In *EMNLP-CoNLL*, pages 678–687, 2007.

[Zettlemoyer and Collins, 2009] Luke S. Zettlemoyer and Michael Collins. Learning context-dependent mappings from sentences to logical form. In *EMNLP-AFNLP*, ACL '09, pages 976–984, 2009.