A New Corpus and Imitation Learning Framework for Context-Dependent Semantic Parsing

Andreas Vlachos
Computer Science Department
University College London
a.vlachos@cs.ucl.ac.uk

Stephen Clark
Computer Laboratory
University of Cambridge
sc609@cam.ac.uk

Abstract

Semantic parsing is the task of translating natural language utterances into a machine-interpretable meaning representation. Most approaches to this task have been evaluated on a small number of existing corpora which assume that all utterances must be interpreted according to a database and typically ignore context. In this paper we present a new, publicly available corpus for context-dependent semantic parsing. The MRL used for the annotation was designed to support a portable, interactive tourist information system. We develop a semantic parser for this corpus by adapting the imitation learning algorithm DAGGER without requiring alignment information during training. DAGGER improves upon independently trained classifiers by 9.0 and 4.8 points in F-score on the development and test sets respectively.

1 Introduction

Semantic parsing is the task of translating natural language utterances into a machine-interpretable meaning representation (MR). Progress in semantic parsing has been facilitated by the existence of corpora containing utterances annotated with MRs, the most commonly used being ATIS (Dahl et al., 1994) and GeoQuery (Zelle, 1995). As these corpora cover rather narrow application domains, recent work has developed corpora to support natural language interfaces to the Freebase database (Cai and Yates, 2013), as well as the development of MT systems (Banarescu et al., 2013).

However, these existing corpora have some important limitations. The MRs accompanying the utterances are typically restricted to some form of database query. Furthermore, in most cases each utterance is interpreted in isolation; thus utterances that use coreference or whose semantics are context-dependent are typically ignored. In this paper we present a new corpus for context-dependent semantic parsing to support the development of an interactive navigation and exploration system for tourism-related activities. The new corpus was annotated with MRs that can handle dialog context such as coreference and can accommodate utterances that are not interpretable according to a database, e.g. repetition requests. The utterances were collected in experiments with human subjects, and contain phenomena such as ellipsis and disfluency. We developed guidelines and annotated 17 dialogs containing 2,374 utterances, with 82.9% exact match agreement between two annotators.

We also develop a semantic parser for this corpus. As the output MRs are rather complex, instead of adopting an approach that searches the output space exhaustively, we use the imitation learning algorithm DAGGER (Ross et al., 2011) that converts learning a structured prediction model into learning a set of classification models. We take advantage of its ability to learn with non-decomposable loss functions and extend it to handle the absence of alignment information during training by developing a randomized expert policy. Our approach improves upon independently trained classifiers by 9.0 and 4.8 F-score on the development and test sets.

2 Meaning Representation Language

Our proposed MR language (MRL) was designed in the context of the portable, interactive navigation and exploration system of Janarthanam et al.

(2013), through which users can obtain information about places and objects of interest, such as monuments and restaurants, as well as directions (see dialog in Fig. 1). The system is aware of the position of the user (through the use of GPS technology) and is designed to be interactive; hence it can initiate the dialog by offering information on nearby points of interest and correcting the route taken by the user if needed. The MRs returned by the semantic parser must represent the user utterances adequately so that the system can generate the appropriate response. The system was developed in the context of the SpaceBook project.[1]

The MRL uses a flat syntax composed of elementary predications, based loosely on minimal recursion semantics (Copestake et al., 2005), but without an explicit treatment of scope. Each MR consists of a dialog act representing the overall function of the utterance, followed for some dialog acts by an unordered set of predicates. All predicates are implicitly conjoined and the names of their arguments specified to improve readability and to allow for some of the arguments to be optional. The argument values can be either constants from the controlled vocabulary, verbatim string extracts from the utterance (enclosed in quotes) or variables (`Xno`). Negation is denoted by a tilde (˜) in front of predicates. The variables are used to bind together the arguments of different predicates within an utterance, as well as to denote coreference across utterances.

The goals in designing the MRL were to remain close to existing semantic formalisms, whilst at the same time producing an MRL that is particularly suited to the application at hand (Janarthanam et al., 2013). We also wanted an MRL that could be computed with efficiently and accurately, given the nature of the NL input. Hence we developed an MRL that is able to express the relevant semantics for the majority of the utterances in our data, without moving to the full expressive power of, e.g., DRT.

Dialog acts The dialog acts are utterance-level labels which capture the overall function of the utterance in the dialog, for example whether an utterance is a question seeking a list as an answer, a statement of information, an acknowledgement, an instruction or a repetition request (`set_question`, `inform`,

USER what's the nearest italian, em, for a meal?

```
dialogAct(set_question)
⋆isA(id:X1, type:restaurant)
def(id:X1)
hasProperty(id:X1, property:cuisine,
            value:"italian")
distance(location:@USER,
         location:X1, value:X2)
argmin(argument:X1, value:X2)
```

WIZARD vapiano's.

```
dialogAct(inform)
isA(id:X4, type:restaurant)
⋆isNamed(id:X4, name:"vapiano's")
equivalent(id:X1, id:X4)
```

USER take me to vapiano!

```
dialogAct(set_question)
⋆route(from_location:@USER,
       to_location:X4)
isA(id:X4, type:restaurant)
isNamed(id:X4, name:"vapiano")
```

WIZARD certainly.

```
dialogAct(acknowledge)
```

WIZARD keep walking straight down clerk street.

```
dialogAct(instruct)
⋆walk(agent:@USER, along_location:X1,
      direction:forward)
isA(id:X1, type:street)
isNamed(id:X1, name:"clerk street")
```

USER yes.

```
dialogAct(acknowledge)
```

USER what is this church?

```
dialogAct(set_question)
⋆isA(id:X2, type:church)
index(id:X2)
```

WIZARD sorry, can you say this again?

```
dialogAct(repeat)
```

USER i said what is this church on my left!

```
dialogAct(set_question)
⋆isA(id:X2, type:church)
index(id:X2)
position(id:X2, ref:@USER,
         location:left)
```

WIZARD it is saint john's.

```
dialogAct(inform)
isA(id:X3, type:church)
⋆isNamed(id:X3, name:"saint john's")
equivalent(id:X2, id:X3)
```

USER A sign here says it is saint mark's.

```
dialogAct(inform)
isA(id:X4, type:church)
⋆isNamed(id:X4, name:"saint mark's")
equivalent(id:X2, id:X4)
```

Figure 1: Sample dialog annotated with MRs

[1] www.spacebook-project.eu

acknowledge, instruct and repeat in Figure 1). The acts defined in the proposed MRL follow the guidelines proposed by Allen and Core (1997), Stolcke et al. (2000) and Bunt et al. (2012).

The dialog acts are divided into two categories. The first category contains those that are accompanied by a set of predicates to represent the semantics of the sentence, such as set_question and inform. For these acts we denote their focal points — for example the piece of information requested in a set_question — with an asterisk ($\star$) in front of the relevant predicate. The second category contains dialog acts that are not accompanied by predicates, such as acknowledge and repeat.

Predicates The MRL contains predicates to denote entities, properties and their relations:

- Predicates introducing entities and their properties: isA, isNamed and hasProperty.
- Predicates describing user actions, such as walk and turn, with arguments such as direction and along_location.
- Predicates describing geographic relations, such as distance, route and position, using ref to denote relative positioning.
- Predicates denoting whether an entity is introduced using a definite article (def), an indefinite (indef) or an indexical (index).
- Predicates expressing numerical relations such as argmin and argmax.

Coreference In order to model coreference we adopt the notion of discourse referents (DRs) and discourse entities (DEs) from Discourse Representation Theory (DRT) (Webber, 1978; Kamp and Reyle, 1993). DRs are referential expressions appearing in utterances which denote DEs, which are mental entities in the speaker's model of discourse. Multiple DEs can refer to the same real-world entity; for example, in Fig. 1 "vapiano's" refers to a different DE from the restaurant in the previous sentence ("the nearest italian"), even though they are likely to be the same real-world entity. We considered DEs instead of actual entities in the MRL because they allow us to capture the semantics of interactions such as the last exchange between the wizard and user. The MRL represents multiple DEs referring to the same real-world entity through the predicate equivalent.

Coreference is indicated by using identical variables across predicate arguments within an utterance or across utterances. The main principle in determining whether DRs corefer is that it must be possible to infer this from the dialog context alone, without using world knowledge.

3 Data Collection and Annotation

The NL utterances were collected using Wizard-of-Oz experiments (Kelley, 1983) with pairs of human subjects. In each experiment, one human pretended to be a tourist visiting Edinburgh (by physically walking around the city), while the other performed the role of the system responding through a suitable interface using a text-to-speech system.

Each user-wizard pair was given one of two scenarios involving requests for directions to different points of interest. The first scenario involves seeking directions to the national museum of Scotland, then going to a nearby coffee shop, followed by a pub via a cash machine and finally looking for a park. The second scenario involves looking for a Japanese restaurant and the university gym, requesting information about the Flodden Wall monument, visiting the Scottish parliament and the Dynamic Earth science centre, and going to the Royal Mile and the Surgeon's Hall museum. Each experiment formed one dialog which was manually transcribed from recorded audio files. 17 dialogs were collected in total, 7 from the first scenario and 10 from the second. More details are reported in Hill et al. (2013).

Given the varied nature of the dialogs, some of the user requests were not within the scope of the system. Furthermore, the proposed MRL has its own limitations; for example it does not have predicates to express temporal relationships. Thus, it was necessary to filter the utterances collected and decide which ones to annotate with MRs.[2] In particular, we did not annotate utterances falling into one or more of the following categories:

- Utterances that are not human-interpretable, e.g. utterances that were interrupted too early to be

[2] A similar filtering process was used for GeoQuery (Section 7.5.1 in Zelle (1995)) and ATIS (principles of interpretation document (/atis3/doc/pofi.doc) in the NIST CDs).

vocabulary type	number of terms
dialog acts	15
predicates	19
arguments	41
constants	9
entity types	26
properties	4

Table 1: MRL vocabulary used in the annotation

interpretable. In such cases, the system is likely to respond with a repetition request.

- Utterances that are human-interpretable but outside the scope of the system, e.g. questions about historical events which are not included in the database of the application considered.
- Utterances that are within the scope of the system but too complex to be represented by the proposed MRL, e.g. an utterance requiring representation of time to be interpreted.

Note that we still annotate an utterance if the core of its semantics can be captured by the MRL. For example, "take me to vapiano now!" would be annotated, even though the MRL cannot represent the meaning of "now". Broad information requests such as "tell me more about this church" are also annotated using the predicate `extraInfo(id:Xno)`. We argue that determining which utterances should be translated into MRs, and which should be ignored, is an important subtask for real-world applications of semantic parsing.

The annotation was performed by one of the authors and a freelance linguist with no experience in semantic parsing. As well as annotating the user utterances, we also annotated the wizard utterances with dialog acts and the entities mentioned, as they provide the necessary context to perform context-dependent interpretation. In practice, though, we expect this information to be used by a natural language generation system to produce the system's response and thus be available to the semantic parser.

The total number of user utterances annotated was 2374, out of which 1906 were annotated with MRs, the remaining not translated due to the reasons discussed earlier in this section. The number and types of the MRL vocabulary terms used appear in Tbl. 1.[3]

In order to assess the quality of the guidelines and the annotation, we conducted an inter-annotator agreement study. For this purpose, the two annotators annotated one dialog consisting of 510 utterances. Exact match agreement at the utterance level, which requires that the MRs by the annotators agree on dialog act, predicates and within-utterance variable assignment, was 0.829, which is a strong result given the complexity of the annotation task, and which suggests that the proposed guidelines can be applied consistently. We also assessed the agreement on predicates using F-score, which was 0.914.

4 Comparison to Existing Corpora

The most closely related corpus to the one presented in this paper (herein SPACEBOOK) is the airline travel information system (ATIS) corpus (Dahl et al., 1994) which consists of dialogs between a user and a flight booking system collected in Wizard-of-Oz experiments. Each utterance is annotated with the SQL statement that would return the requested piece of information from the flights database. The utterance interpretation is context-dependent. For example, when the user follows up an initial flight request — e.g. "find me flights to Boston" — with utterances containing additional preferences — e.g. "on Monday" — the interpretation of the additional preferences extends the MR for the initial request.

Compared to ATIS, the dialogs in the SPACEBOOK corpus are substantially longer (8.8 vs. 139.7 utterances on average respectively) and cover a broader domain due to the longer scenarios used in data collection. Furthermore, allowing the wizards to answer in natural language instead of restricting them to responding via database queries as in ATIS led to more varied dialogs. Finally, our approach to annotating coreference avoids repeating the MR of previous utterances, thus resulting in shorter expressions that are closer to the semantics of the NL utterances.

The datasets developed in the recent dialog state tracking challenge (Henderson et al., 2014) also consist of dialogs between a user and a tourism information system. However the task is easier since only three entity types are considered (restaurant, coffeeshop and pub), a slot-filling MRL is used and the

[3] The annotated dialogs, the guidelines and the lists of the vocabulary terms are available from `https://sites.google.com/site/andreasvlachos/resources`.

argument slots take values from fixed lists.

The abstract meaning representation (AMR) described by Banarescu et al. (2013) was developed to provide a semantic interpretation layer to improve machine translation (MT) systems. It has similar predicate argument structure to the MRL proposed here, including a lack of cover for temporal relations and scoping. However, due to the different application domains (MT vs. tourism-related activities), there are some differences. Since MT systems operate at the sentence-level, each sentence is interpreted in isolation in AMR, whilst our proposed MRL takes context into account. Also, AMR tries to account for all the words in a sentence, whilst our MRL only tries to capture the semantics of those words that are relevant to the application at hand.

Other popular semantic parsing corpora include GeoQuery (Zelle, 1995) and Free-917 (Cai and Yates, 2013). Both consist exclusively of questions to be answered with a database query, the former considering a small American geography database and the latter the much wider Freebase database (Bollacker et al., 2008). Unlike SpaceBook and ATIS, there is no notion of context in either of these corpora. Furthermore, the NL utterances in these corpora are compiled to be interpreted as database queries, which is equivalent to only one of the dialog acts (`set_question`) in the SpaceBook corpus. Thus the latter allows the exploration of the application of dialog act tagging as a first step in semantic parsing. Finally, MacMahon et al. (2006) developed a corpus of natural language instructions paired with sequences of actions; however the domain is limited to simple navigation instructions and there is no notion of dialog in this corpus.

5 Semantic Parsing for the New Corpus

The MRL in Fig. 1 is readable and easy to annotate with. However, it is not ideal for experiments, as it is difficult to compare MR expressions beyond exact match. For these reasons, we converted the MR expressions into a node-argument form. In particular, all predicates introducing entities (`isA`) and most predicates introducing relations among entities (e.g. `distance`) become nodes, while all other predicates (e.g. `isNamed`, `def`) are converted into arguments. For example, the MR for the first utterance in Fig. 1 is converted into the form in Fig. 2g. Entities appearing in MR expressions without a type (e.g. X2 in the last utterance of Fig. 1) are denoted with a node of type `empty`. Each node has a unique id (e.g. X1) and each argument can take as value a constant (e.g. *det*), a node id, or a verbatim string extract from the utterance. Arguments that are absent (e.g. the `name` of `restaurant`) are set to the constant *null*. This conversion results in 16 utterance-level labels (15 dialog acts plus one for the non-interpretable utterances), 35 node types and 32 arguments.

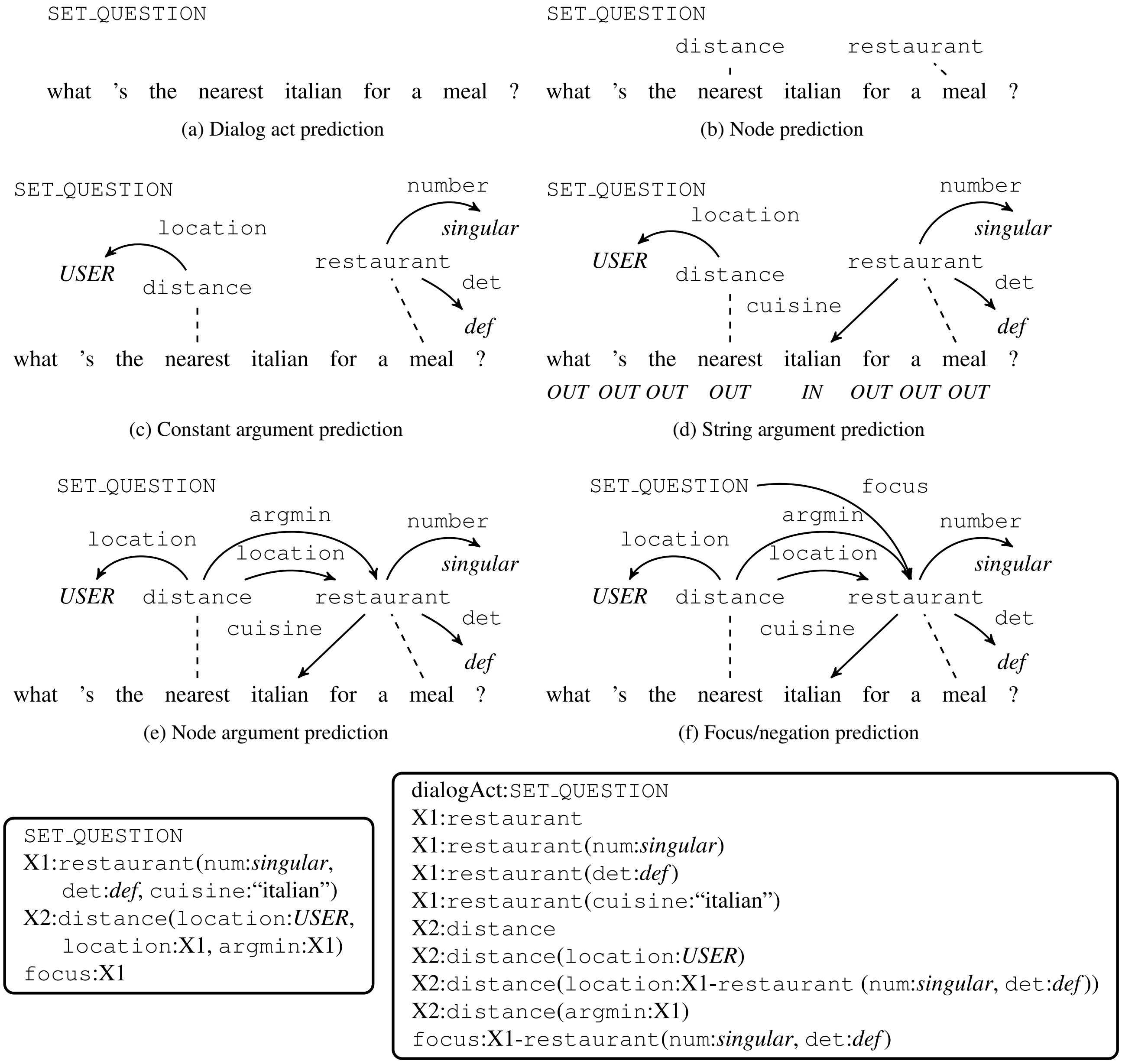
The comparison between a predicted and a gold standard node-argument form is performed in three stages. First we map the ids of the predicted nodes to those of the gold standard. While ids do not carry any semantics, they are needed to differentiate between multiple nodes of the same type; e.g. if a second `restaurant` had been predicted in Fig. 2h then it would have a different id and would not be matched to a gold standard node. Second, we decompose the node-argument forms into a set of atomic predictions (Fig. 2h). This decomposition allows the awarding of partial credit, e.g. when the node type is correct but some of the arguments are not. Using these atomic predictions we calculate precision, recall and F-score.

The mapping between predicted and gold standard ids is performed by evaluating all mappings (with mappings between nodes of different types not allowed), and choosing the one resulting in the lowest sum of false positives and negatives.

5.1 Task decomposition

Fig. 2 shows the decomposition of the semantic parsing task in stages, which are described below.

Dialog act prediction We first assign an utterance-level label using a classifier that exploits features based on the textual content of the utterance and on the utterance preceding it. The features extracted from the utterance are all unigrams, bigrams and trigrams and the final punctuation mark. Unlike in typical text classification tasks, content words are not always helpful in dialog act tagging; e.g. the token “meal” in Fig. 2a is not indicative of `set_question`, while n-grams of words typically considered as stopwords, such as “what ’s the”, can be more helpful. If the dialog act



(a) Dialog act prediction

(b) Node prediction

(c) Constant argument prediction

(d) String argument prediction

(e) Node argument prediction

(f) Focus/negation prediction

SET_QUESTION
X1:restaurant(num:*singular*,
det:*def*, cuisine:“italian”)
X2:distance(location:*USER*,
location:X1, argmin:X1)
focus:X1

dialogAct:SET_QUESTION
X1:restaurant
X1:restaurant(num:*singular*)
X1:restaurant(det:*def*)
X1:restaurant(cuisine:“italian”)
X2:distance
X2:distance(location:*USER*)
X2:distance(location:X1-restaurant (num:*singular*, det:*def*))
X2:distance(argmin:X1)
focus:X1-restaurant(num:*singular*, det:*def*)

(g) Node-argument form

(h) Atomic predictions

Figure 2: Semantic parsing decomposition.

predicted is to be accompanied by other predicates according to the guidelines (Sec. 2) we proceed to the following stages, otherwise stop.

The features based on the preceding utterance indicate whether it was by the user or the wizard and, in the latter case, its dialog act. Such features are useful in determining the act of short, ambiguous utterances such as “yes”, which is tagged as yes when following a prop_question utterance, but as acknowledge otherwise.

Node prediction In node prediction we use a classifier to predict whether each of the tokens in the utterance denotes a node of a particular type or empty (Fig. 2b). The features used include the target token and its lemma, which are conjoined with the PoS tag, the previous and following tokens, as well as the lemmas of the tokens with which it has syntactic dependencies. Further features represent the dialog act (e.g. route is more likely to appear in a set_question utterance), and the number and types of the nodes already predicted. Since the evaluation ignores the alignment between nodes and tokens, it would have been correct to predict the correct nodes from any token; e.g. `restaurant` could be predicted from "italian" instead. However, alignment does affect argument prediction, since it determines its feature extraction.

Constant argument prediction In this stage (Fig. 2c) we predict, for each argument of each node, whether its value is an MRL vocabulary term, a verbatim string extract, a node, or absent (special values *STRING*, *NODE*, *null* respectively). If the value predicted is *STRING* or *NODE* it is replaced by the predictions in subsequent stages. For each argument different values are possible; thus we use separate classifiers for each, resulting in 32 classifiers. The features used include the node type, the token that predicted the node, and the syntactic dependency paths from that token to all other tokens in the utterance. We also include as features the values predicted for other arguments of the node, the dialog act, and the other node types predicted.

String argument prediction For each argument predicted to be *STRING* (e.g. `cuisine` in Figure 2d), we predict for each token in left-to-right order whether it should be part of the value for this argument or not (*IN* or *OUT*). Since the strings that are appropriate for each argument differ (e.g. the strings for `cuisine` are unlikely to be appropriate for `name`), we use separate classifiers for each of them, resulting in five classifiers. The features used include the target token and its lemma, its conjunction with the PoS tag, the previous and following tokens, and the lemmas of the tokens with which it has syntactic dependencies. We also added the label assigned to the previous token and the syntactic dependency path to the token that predicted the node.

Node argument prediction For each argument predicted to have *NODE* as its value, we predict for every other node whether it should be the value or not (e.g. `argmin` in Fig. 2e). As with the string argument prediction, we use separate binary classifiers for each argument, resulting in 18 classifiers. The features extracted are similar to that stage, but we now consider the tokens that predicted each candidate argument node (e.g. "meal" for `restaurant`) instead of the tokens in the utterance.

Focus/Negation prediction We predict whether each node should be focused or negated as two separate binary tasks. The features used include the token that predicted the target node, its lemma and PoS tag and the syntactic dependency paths to all other tokens in the utterance. Further features include the type of the node and its arguments.

6 Imitation Learning

In order to learn the classifiers for the task decomposition described, two challenges must be addressed. The first is the complexity of the structure to be predicted. The task involves many interdependent predictions made by a variety of classifiers, and thus cannot be tackled by approaches that assume a particular type of graph structure, or restrict structure feature extraction in order to perform efficient dynamic programming. The second challenge is the lack of alignment information during training. Imitation learning algorithms such as SEARN (Daumé III et al., 2009) and DAGGER (Ross et al., 2011) have been applied successfully to a variety of structured prediction tasks including summarization, biomedical event extraction and dynamic feature selection (Daumé III et al., 2009; Vlachos, 2012; He et al., 2013) thanks to their ability to handle complex output spaces without exhaustive search and their flexibility in incorporating features based on the structured output. In this work we focus on DAGGER and extend it to handle the missing alignments.

6.1 Structured prediction with DAGGER

The dataset aggregation (DAGGER) algorithm (Ross et al., 2011) forms the prediction of an instance s as a sequence of T actions $\hat{y}_{1:T}$ predicted by a learned policy which consists of one or more classifiers. These actions are taken in a greedy fashion, i.e. once an action has been taken it cannot be changed. During training, it converts the problem of learning how to predict these sequences of actions into cost sensitive classification (CSC) learning. In CSC learning each training example has a vector of misclassification costs associated with it, thus rendering some mistakes on some examples to be more expensive than others (Domingos, 1999).

Algorithm 1: Imitation learning with DAGGER

```
   Input: training instances $\mathcal{S}$, expert policy $\pi^\star$,
   loss function $\ell$, learning rate $\beta$, CSC learner $CSCL$
   Output: Learned policy $H_N$
1  CSC Examples $E = \emptyset$
2  for $i = 1$ to $N$ do
3      $p = (1 - \beta)^{i-1}$
4      current policy $\pi = p\pi^\star + (1 - p)H_{i-1}$
5      for $s$ in $S$ do
6          Predict $\pi(s) = \hat{y}_{1:T}$
7          for $\hat{y}_t$ in $\pi(s)$ do
8              Extract features $\Phi_t = f(s, \hat{y}_{1:t-1})$
9              foreach possible action $y_t^j$ do
10                 Predict $y\prime_{t+1:T} = \pi(s; \hat{y}_{1:t-1}, y_t^j)$
11                 Assess $c_t^j = \ell(\hat{y}_{1:t-1}, y_t^j, y\prime_{t+1:T})$
12             Add $(\Phi_t, c_t)$ to $E$
13     Learn $H_i = CSCL(E)$
```

Algorithm 1 presents the training procedure. DAGGER requires a set of labeled training instances $\mathcal{S}$ and a loss function ℓ that compares complete outputs for instances in $\mathcal{S}$ against the gold standard. In addition, an expert policy $\pi^\star$ must be specified which is an oracle that returns the optimal action for the instances in $\mathcal{S}$, akin to an expert demonstrating the task. $\pi^\star$ is typically derived from the gold standard; e.g. in part of speech tagging $\pi^\star$ would return the correct tag for each token. In addition, the learning rate β and a CSC learner (*CSCL*) must be provided. The algorithm outputs a learned policy H_N that, unlike $\pi^\star$, can generalize to unseen data.

Each training iteration begins by setting the probability p (line 3) of using $\pi^\star$ in the current policy π. In the first iteration, only $\pi^\star$ is used but, in later iterations, π becomes stochastic and, for each action, $\pi^\star$ is used with probability p, and the learned policy from the previous iteration H_{i-1} with probability $1 - p$ (line 4). Then π is used to predict each training instance s (line 6). For each action $\hat{y}_t$, a CSC example is generated (lines 7-12). The features Φ_t are extracted from s and all previous actions $\hat{y}_{1:t-1}$ (line 8). The cost for each possible action y_t^j is estimated by predicting the remaining actions $y\prime_{t+1:T}$ for s using π (line 10) and calculating the loss incurred given y_t^j w.r.t. the gold standard for s using ℓ (line 11). As π is stochastic, it is common to use multiple samples of $y\prime_{t+1:T}$ to assess the cost of each action y_t^j by repeating lines 10-11. The features, together with the costs for each possible action, form a CSC example (Φ_t, c_t) (line 12). At the end of each iteration, the CSC examples obtained from all iterations are used by the CSC learning algorithm to learn the classifier(s) for H_i (line 13).

When predicting the training instances (line 6), and when estimating the costs for each possible action (lines 10-11), the policy learned in the previous iteration H_{i-1} is used as part of π after the first iteration. Thus the CSC examples generated to learn H_i depend on the predictions of H_{i-1} and, by gradually increasing the use of H_{i-1} and ignoring $\pi^\star$ in π, the learned policies are adjusted to their own predictions, thus learning the dependencies among the actions and how to predict them in order to minimize the loss. The learning rate β determines how fast π moves away from $\pi^\star$. The use of H_{i-1} in predicting the training instances (line 6) also has the effect of exploring sub-optimal actions so that the learned policies are adjusted to recover from their mistakes. Finally, note that if only one training iteration is performed, the learned policy is equivalent to a set of independently trained classifiers since no training against the predictions of the previously learned policy takes place.

6.2 Training with missing alignments

The loss function ℓ in DAGGER is only used to compare complete outputs against the gold standard. Therefore, when generating a CSC training example in DAGGER (lines 7-12), we do not need to know whether an action y_t^j is correct or not, we only evaluate what the effect of y_t^j is on the loss incurred by the complete action sequence. Thus, it does not need to decompose over the actions taken to evaluate them. The ability to train against non-decomposable loss functions is useful when the training data has missing labels, as is the case with semantic parsing. Following Sec. 5, ℓ is defined as the sum of the false positive and false negative atomic predictions used to calculate precision and recall and, since it ignores the alignment between tokens and nodes, it cannot assess node prediction actions. However, we can use it under DAGGER to learn a node prediction classifier together with the classifiers of the other stages.

The only component of DAGGER which assumes knowledge of the correct actions for training is the expert policy $\pi^\star$. Since these are not available for the node prediction stage, we replace $\pi^\star$ with a randomized expert policy π^{rand}, in which actions that are not specified by the annotation are chosen randomly from a set of equally optimal ones. For example, in Fig. 2b when predicting the action for each token, π^{rand} chooses randomly among `null`, `distance`, and `restaurant`, so that by the end of the stage the correct nodes have been predicted. Randomizing this choice helps explore the actions available. In our experiments we placed a uniform distribution over the available actions, i.e. all optimal actions are equally likely to be chosen. The actions returned by π^{rand} will often result in alignments that do not incur any loss but are nonsensical, e.g. predicting `restaurant` from "what". However, since π^{rand} is progressively ignored, the effect of such actions is reduced.

While being able to learn a semantic parser without alignment information is useful, it would help to use some supervision, e.g. that "street" commonly predicts the node `street`. We incorporate such an alignment dictionary in π^{rand} as follows: if the target token is mapped to a node type in the dictionary, and if a node of this type needs to be predicted for the utterance, then this type is returned. Otherwise, the prediction is made with π^{rand}. Finally, like π^{rand} itself, the dictionary is progressively ignored and neither constrains the training process, nor is used during testing.

7 Experiments

We split the annotated dialogs into training and test sets. The former consists of four dialogs from the first scenario and seven from the second, and the latter of three dialogs from each scenario. All development and feature engineering was conducted using cross-validation on the training set, at the dialog level rather than the utterance level (therefore resulting in as many folds as dialogs in the training set), to ensure that each fold contains utterances from all parts of the scenario from which the dialog is taken.

To perform cost-sensitive classification learning we used the adaptive regularization of weight vectors (AROW) algorithm (Crammer et al., 2009). AROW is an online algorithm for linear predictors that adjusts the per-feature learning rates so that popular features do not overshadow rare but useful ones. Given the task decomposition, each learned hypothesis consists of 59 classifiers. We restricted the prediction of nodes to content words since function words are unlikely to provide useful alignments. All preprocessing was performed using the Stanford CoreNLP toolkit (Manning et al., 2014).[4] The DAGGER parameters were set to 12 training iterations, $\beta = 0.3$ and 3 samples for action cost assessment. We compared our DAGGER-based imitation learning approach (henceforth *Imit*) against independently trained classifiers using the same classification learner and features (henceforth *Indep*).

For both systems we incorporated an alignment dictionary (*+align* versions) as described in Sec. 6.2, in order to improve node prediction performance. The dictionary was extracted from the training data and contains 96 tokens that commonly predict a particular node type.

The results from the cross-validation experiments are reported in Tbl. 2. Overall performance evaluated as described in Sec. 5 was 53.6 points in F-score for *Imit*, 5.7 points higher than *Indep* and the difference is greater for the *+align* versions. These results demonstrate the advantages of training classifiers using imitation learning versus independently trained classifiers. Isolating the performance for node and argument prediction stages, we observe that the main bottleneck is the former, which in the case of *Imit* is 60.9 points in F-score compared to 78.8 for the latter. Accuracy for dialog acts is 78.9%.

As shown in Tbl. 2, the alignment dictionary improved not only node prediction performance by 6 points in F-score, but also argument prediction by 2.5 points, thus demonstrating the benefits of learning the alignments together with the other components of the semantic parser. The overall performance improved by 5.5 points in F-score.

Finally, we ran an experiment with oracle node prediction and found that the overall performance using cross-validation on the training data improved to 88.2 and 79.9 points in F-score for the *Imit+align* *Indep+align* systems. This is in agreement with the results presented by Flanigan et al. (2014) on developing a semantic parsing parser for the AMR for-

[4] The implementation of the semantic parser is available from `https://sites.google.com/site/andreasvlachos/resources`.

	Imit			*Imit+align*			*Indep*			*Indep+align*		
exact match (accuracy)	58.4%			59.1%			56%			55.9%		
dialog act (accuracy)	78.9%			79.3%			78.8%			79%		
nodes (Rec/Prec/F)	72.3	52.6	60.9	76.1	59.8	66.9	44.4	61.6	51.6	53.3	64	58.1
arguments (Rec/Prec/F)	77.6	80	78.8	79.6	83	81.3	74.1	67.2	70.1	78.2	66.3	71.8
focus (Rec/Prec/F)	81.8	87.2	84.4	84.4	86.7	85.5	85.9	87	86.5	86.8	8.3	84.7
overall (Rec/Prec/F)	59.3	48.9	53.6	62.2	54.4	59.1	45.3	50.8	47.9	50	50.1	50.1

Table 2: Performances using 11-fold cross-validation on the training set.

malism who also argue that node prediction is the main performance bottleneck.

Tbl. 3 gives results on the test set. The overall performance for *Imit* is 48.4 F-score and 47.9% for exact match. As in the cross-validation results on the training data, training with imitation learning improved upon independently trained classifiers. The performance was improved further using the alignment dictionary, reaching 53.5 points in F-score and 49.1% exact match accuracy.

In the experimental setup above, dialogs from the same scenarios appear in both training and testing. While this is a reasonable evaluation approach also followed in ATIS evaluations, it is likely to be relatively forgiving; in practice, semantic parsers are likely to encounter entities, activities, etc. unseen in training. Hence we conducted a second evaluation in which dialogs from one scenario are used to train a parser evaluated on the other (still respecting the train/test split from before). When testing on the dialogs from the first scenario and training on the dialogs from the second, the overall performance using *Imit+align* was 36.9 points in F-score, while in the reverse experiment it was 41.7. Note that direct comparisons against the performances in Tbl. 3 are not meaningful since fewer dialogs are being used for training and testing in the cross-scenario setup.

8 Comparison with Related Work

Previous work on semantic parsing handled the lack of alignments during training in a variety of ways. Zettlemoyer and Collins (2009) manually engineered a CCG lexicon for the ATIS corpus. Kwiatkowski et al. (2011) used a dedicated algorithm to infer a similar dictionary and used alignments from Giza++ (Och and Ney, 2000) to initialize the relevant features. Most recent work on GeoQuery uses an alignment dictionary that includes for each geographical entity all noun phrases referring to it (Jones et al., 2012). More recently, Flanigan et al. (2014) developed a dedicated alignment model on top of which they learned a semantic parser for the AMR formalism. In our approach, we learn the alignments together with the semantic parser without requiring a dictionary.

In terms of structured prediction frameworks, most previous work uses hidden variable linear (Zettlemoyer and Collins, 2007) or log-linear (Liang et al., 2011) models with beam search. In terms of direct comparisons with existing work, the goal of this paper is to introduce the new corpus and provide a competitive first attempt at the new semantic parsing task. However, we believe it is non-trivial to apply existing approaches to the new task, since, assuming a decomposition similar to that of Sec. 5.1, exhaustive search would be too expensive, and applying vanilla beam search would be difficult since different predictions result in beams of (sometimes radically) different lengths that are not comparable.

We have attempted applying the MT-based semantic parsing approach proposed by Andreas et al. (2013) to our dataset but in initial experiments the performance was poor. The main reason for this is that, unlike GeoQuery, the proposed MRL does not align well with English.

The expert policy in DAGGER is a generalization of the dynamic oracle of Goldberg and Nivre (2013) for shift-reduce dependency parsing to any structured prediction task decomposed into a sequence of actions. The randomized expert policy proposed extends DAGGER to learn not only how to avoid error propagation, but also how to infer latent variables.

The main bottleneck is training data sparsity. Some node types appear only a few times in relatively long utterances, and thus it is difficult to infer appropriate alignments for them. Unlike machine translation between natural languages, it is unreal-

	Imit			*Imit+align*			*Indep*			*Indep+align*		
exact match (accuracy)	47.9%			49.1%			47.6%			46.1%		
dialog act (accuracy)	77%			80.5%			79.8%			79.5%		
nodes (Rec/Prec/F)	68.7	45.7	54.8	75.5	51.7	61.4	41.9	61.1	49.7	54	64.9	58.9
arguments (Rec/Prec/F)	73.9	73.7	73.8	76.8	77.3	77.1	69.5	61.3	65.1	77.3	63.6	69.8
focus (Rec/Prec/F)	87.1	80.7	83.8	86	81.2	83.6	81.6	73.4	77.3	90.6	76.8	83.1
overall (Rec/Prec/F)	56.6	42.3	48.4	63.5	46.2	53.5	41.2	47.8	44.3	50	47.4	48.7

Table 3: Performances on the test set.

istic to expect large quantities of utterances to be annotated with MR expressions. An appealing alternative would be to use response-based learning, i.e. use the response from the system instead of MR expressions as training signal (Liang et al., 2011; Kwiatkowski et al., 2013; Berant and Liang, 2014). However such an approach would not be straightforward to implement in our application, since the response from the system is not always the result of a database query but, e.g., a navigation instruction that is context-dependent and thus difficult to assess its correctness. Furthermore, it would require the development of a user simulator (Keizer et al., 2012), a non-trivial task which is beyond the scope of this work. A different approach is to use dialogs between a system and its users as proposed by Artzi and Zettlemoyer (2011) using the DARPA communicator corpus (Walker et al., 2002). However, in that work utterances were selected to be shorter than 6 words and to include one noun phrase present in the lexicon used during learning while ignoring short but common phrases such as "yes" and "no"; thus it is unclear whether it would be applicable to our dataset.

Finally, dialog context is only taken into account in predicting the dialog act for each utterance. Even though our corpus contains coreference information, we did not attempt this task as it is difficult to evaluate and our performance on node prediction on which it relies is relatively low. We leave coreference resolution on the new corpus as an interesting and challenging task for future work.

9 Conclusions

In this paper we presented a new corpus for context-dependent semantic parsing in the context of a portable, interactive navigation and exploration system for tourism-related activities. The MRL used for the annotation can handle dialog context such as coreference and can accommodate utterances that are not interpretable according to a database. We conducted an inter-annotator agreement study and found 0.829 exact match agreement.

We also developed a semantic parser for the SpaceBook corpus using the imitation learning algorithm DAgger that, unlike previous approaches, can infer the missing alignments in the training data using a randomized expert policy. In experiments using the new corpus we found that training with imitation learning substantially improves performance compared to independently trained classifiers. Finally, we showed how to improve performance further by incorporating an alignment dictionary.

Acknowledgements

The research reported was conducted while the first author was at the University of Cambridge and funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 270019 (SpaceBook project `www.spacebook-project.eu`). The authors would like to acknowledge the work of Diane Nicholls in the annotation; the efforts of Robin Hill in collecting the dialogs from Wizard-of-Oz experiments; and Tim Vieira for helpful comments on an earlier version of this manuscript.

References

James Allen and Mark Core. 1997. Dialogue act markup in several layers. Technical report, University of Rochester.

Jacob Andreas, Andreas Vlachos, and Stephen Clark. 2013. Semantic parsing as machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (short papers)*.

Yoav Artzi and Luke Zettlemoyer. 2011. Bootstrapping semantic parsers from conversations. In *Proceedings* *of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 421–432, Edinburgh, UK.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria, August. Association for Computational Linguistics.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1247–1250.

Harry Bunt, Jan Alexandersson, Jae-Woong Choe, Alex Chengyu Fang, Koiti Hasida, Volha Petukhova, Andrei Popescu-Belis, and David Traum. 2012. Iso 24617-2: A semantically-based standard for dialogue annotation. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*, Istanbul, Turkey.

Qingqing Cai and Alexander Yates. 2013. Large-scale Semantic Parsing via Schema Matching and Lexicon Extension. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

Ann Copestake, Dan Flickinger, Ivan Sag, and Carl Pollard. 2005. Minimal recursion semantics: An introduction. *Research in Language and Computation*, 3(2–3):281–332.

Koby Crammer, Alex Kulesza, and Mark Dredze. 2009. Adaptive regularization of weight vectors. In *Advances in Neural Information Processing Systems 22*, pages 414–422.

Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the scope of the ATIS task: the ATIS-3 corpus. In *Proceedings of the Workshop on Human Language Technology*, pages 43–48, Plainsboro, New Jersey.

Hal Daumé III, John Langford, and Daniel Marcu. 2009. Search-based structured prediction. *Machine Learning*, 75:297–325.

Pedro Domingos. 1999. Metacost: a general method for making classifiers cost-sensitive. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining*, pages 155–164. Association for Computing Machinery.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Baltimore, Maryland, June. Association for Computational Linguistics.

Yoav Goldberg and Joakim Nivre. 2013. Training deterministic parsers with non-deterministic oracles. *Transactions of the Association for Computational Linguistics*, 3(1):403–414, October.

He He, Hal Daumé III, and Jason Eisner. 2013. Dynamic feature selection for dependency parsing. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1455–1464, Seattle, October.

Matthew Henderson, Blaise Thomson, and Jason Williams. 2014. The Third Dialog State Tracking Challenge. In *Proceedings of IEEE Spoken Language Technology*.

Robin Hill, Jana Götze, and Bonnie Webber. 2013. SpaceBook Project: Final Data Release, Wizard-of-Oz (WoZ) experiments. Technical report, University of Edinburgh.

Srinivasan Janarthanam, Oliver Lemon, Phil Bartie, Tiphaine Dalmas, Anna Dickinson, Xingkun Liu, William Mackaness, and Bonnie Webber. 2013. Evaluating a city exploration dialogue system with integrated question-answering and pedestrian navigation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1660–1668, Sofia, Bulgaria, August. Association for Computational Linguistics.

Bevan Keeley Jones, Mark Johnson, and Sharon Goldwater. 2012. Semantic parsing with Bayesian tree transducers. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 488–496.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer, Dordrecht.

Simon Keizer, Stphane Rossignol, Senthilkumar Chandramohan, and Olivier Pietquin. 2012. User simulation in the development of statistical spoken dialogue systems. In Oliver Lemon and Olivier Pietquin, editors, *Data-Driven Methods for Adaptive Spoken Dialogue Systems*, pages 39–73. Springer New York.

John F. Kelley. 1983. An empirical methodology for writing user-friendly natural language computer applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 193–196.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2011. Lexical generalization in CCG grammar induction for semantic parsing. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1512–1523, Edinburgh, UK.

Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke Zettlemoyer. 2013. Scaling semantic parsers with on-the-fly ontology matching. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1545–1556, Seattle, WA.

Percy Liang, Michael Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 590–599, Portland, Oregon.

Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. 2006. Walk the talk: connecting language, knowledge, and action in route instructions. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 1475–1482. AAAI Press.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong, China.

Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *14th International Conference on Artificial Intelligence and Statistics*, pages 627–635.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Andreas Vlachos. 2012. An investigation of imitation learning algorithms for structured prediction. *Journal of Machine Learning Research Workshop and Conference Proceedings, Proceedings of the 10th European Workshop on Reinforcement Learning*, 24:143–154.

Marilyn A. Walker, Alexander I. Rudnicky, Rashmi Prasad, John S. Aberdeen, Elizabeth Owen Bratt, John S. Garofolo, Helen Wright Hastie, Audrey N. Le, Bryan L. Pellom, Alexandros Potamianos, Rebecca J. Passonneau, Salim Roukos, Gregory A. Sanders, Stephanie Seneff, and David Stallard. 2002. DARPA communicator: cross-system results for the 2001 evaluation. In *Proceedings of the 7th International Conference on Spoken Language Processing*.

Bonnie Lynn Webber. 1978. *A Formal Approach to Discourse Anaphora*. Ph.D. thesis, Harvard University.

John M. Zelle. 1995. *Using Inductive Logic Programming to Automate the Construction of Natural Language Parsers*. Ph.D. thesis, Department of Computer Sciences, The University of Texas at Austin.

Luke S. Zettlemoyer and Michael Collins. 2007. Online learning of relaxed CCG grammars for parsing to logical form. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 678–687.

Luke S. Zettlemoyer and Michael Collins. 2009. Learning context-dependent mappings from sentences to logical form. In *Proceedings of the Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 976–984, Singapore.