Reference Resolution in Situated Dialogue with Learned Semantics

Xiaolong Li
Computer & Information Science
& Engineering
University of Florida
xiaolongl@ufl.edu

Kristy Elizabeth Boyer
Computer & Information Science
& Engineering
University of Florida
keboyer@ufl.edu

Abstract

Understanding situated dialogue requires identifying referents in the environment to which the dialogue participants refer. This reference resolution problem, often in a complex environment with high ambiguity, is very challenging. We propose an approach that addresses those challenges by combining learned semantic structure of referring expressions with dialogue history into a ranking-based model. We evaluate the new technique on a corpus of human-human tutorial dialogues for computer programming. The experimental results show a substantial performance improvement over two recent state-of-the-art approaches. The proposed work makes a stride toward automated dialogue in complex problem-solving environments.

1 Introduction

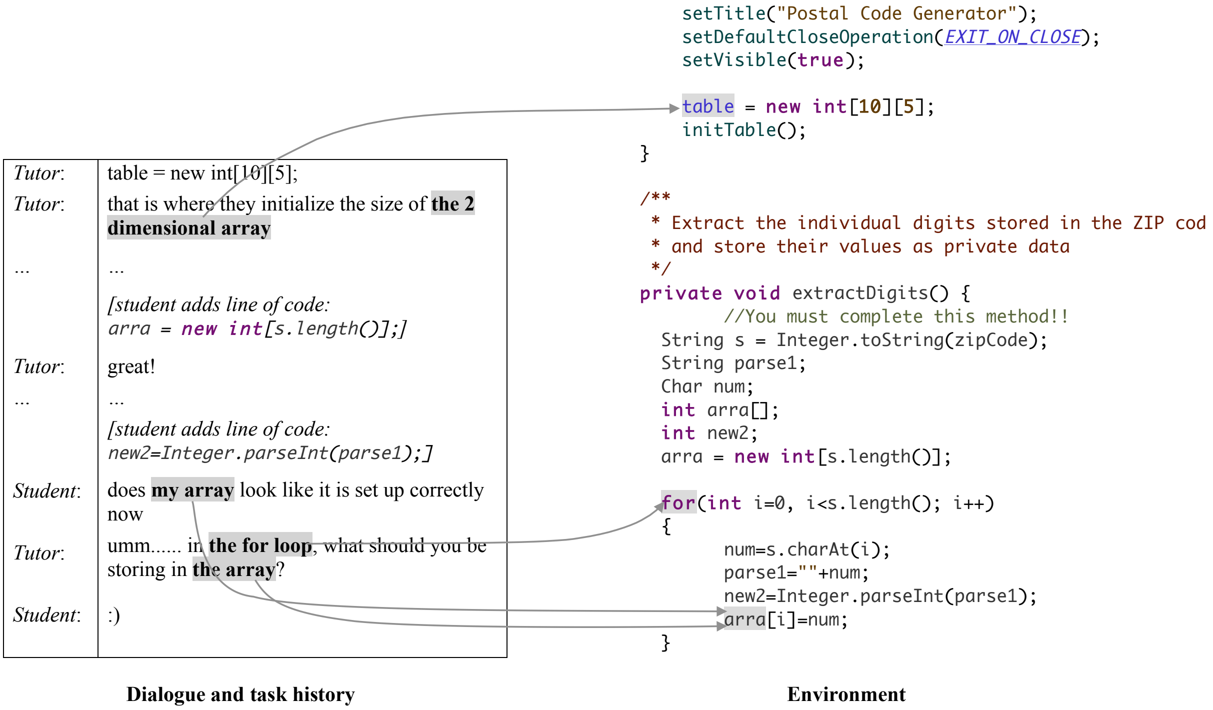
The content of a situated dialogue is very closely related to the environment in which it happens (Grosz and Sidner, 1986). As dialogue systems move toward assisting users in increasingly complex tasks, these systems must understand users' language within the environment of the tasks. To achieve this goal, dialogue systems must perform reference resolution, which involves identifying the referents in the environment that the user refers to (Iida et al., 2010; Liu et al., 2014; Liu and Chai, 2015; Chai et al., 2004). Imagine a dialogue system that assists a novice student in solving a programming problem. To understand a question or statement the student poses, such as, "Should I use the 2 dimensional array?", the system must link the *referring expression* "the 2 dimensional array" to an *object*[1] in the *environment*.

This process is illustrated in Figure 1, which shows an excerpt from a corpus of tutorial dialogue situated in an introductory computer programming task in the Java programming language. The arrows link referring expressions in the situated dialogue to their referents in the environment. To identify the referent of each referring expression, it is essential to capture the semantic structure of the referring expression of the object it refers to, such as *"the 2 dimensional array"* contains two attributes, *"2 dimensional"* and *"array"*. At the same time, the dialogue history and the history of user task actions (such as editing the code) play a key role. To disambiguate the referent of *"my array"*, temporal information is needed: in this case, the referent is a variable named `arra`, which is an array that the student has just created.

Reference resolution in situated dialogue is challenging because of the ambiguity inherent within dialogue utterances and the complexity of the environment. Prior work has leveraged dialogue history and task history information to improve the accuracy of reference resolution (Iida et al., 2010; Iida et al., 2011; Funakoshi et al., 2012). However, these prior approaches have employed relatively simple semantic information from the referring expressions, such as a manually created lexicon, or have operated within an environment with a limited set of pre-defined objects. Besides reference resolution in situated dialogue, there is also a research direction in which machine learning models are used to learn the semantics of noun phrases in order to map noun phrases to objects in a related environment (Kennington and Schlangen, 2015; Liang et al., 2009; Naim et al., 2014; Kushman et al., 2014). However, these prior approaches operated at the granularity of single

[1] The word "object" has a technical meaning within the domain of object-oriented programming, which is the domain of the corpus utilized in this work. However, we follow the standard usage of "object" in situated dialogue (Iida et al., 2010), which for programming is any portion of code in the environment.

Proceedings of the SIGDIAL 2016 Conference, pages 329–338,
Los Angeles, USA, 13-15 September 2016. ©2016 Association for Computational Linguistics



Tutor:	table = new int[10][5];
Tutor:	that is where they initialize the size of **the 2 dimensional array**
...	...
	[student adds line of code: *arra = new int[s.length()];]*
Tutor:	great!
...	...
	[student adds line of code: *new2=Integer.parseInt(parse1);]*
Student:	does **my array** look like it is set up correctly now
Tutor:	umm...... in **the for loop**, what should you be storing in **the array**?
Student:	:)

```
    setTitle("Postal Code Generator");
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    setVisible(true);

    table = new int[10][5];
    initTable();
}

/**
 * Extract the individual digits stored in the ZIP cod
 * and store their values as private data
 */
private void extractDigits() {
        //You must complete this method!!
  String s = Integer.toString(zipCode);
  String parse1;
  Char num;
  int arra[];
  int new2;
  arra = new int[s.length()];

  for(int i=0, i<s.length(); i++)
  {
        num=s.charAt(i);
        parse1=""+num;
        new2=Integer.parseInt(parse1);
        arra[i]=num;
  }
```

Dialogue and task history **Environment**

Figure 1 Excerpt of tutorial dialogue illustrating reference resolution. Referring expressions are shown in bold italics.[2]

spoken utterances not contextualized within a dialogue history, and they too focus on environments with a limited number (and a pre-defined set) of objects. As this paper demonstrates, these prior approaches do not perform well in situated dialogues for complex problem solving, in which the user creates, modifies, and removes objects from the environment in unpredictable ways.

To tackle the problem of reference resolution in this type of situated dialogue, we propose an approach that combines semantics from a conditional-random-field-based semantic parser along with salient features from dialogue history and task history. We evaluate this approach on the JavaTutor corpus, a corpus of textual tutorial dialogue collected within an online environment for computer programming. The results show that our approach achieves substantial improvement over two existing state-of-the-art approaches, with existing approaches achieving 55.2% accuracy at best, and the new approach achieving 68.5% accuracy.

2 Related Work

The work in this paper is informed by research in coreference resolution for text as well as reference resolution in situated dialogue and multi-modal environments. This section describes related work in those areas.

The classic reference resolution problem for discourse aims to resolve coreference relationships within a given text (Martschat and Strube, 2015; McCarthy and Lehnert, 1995; Soon et al., 2001). Effective approaches for discourse cannot be directly applied to the problem of linking referring expressions to their referents in a rich situated dialogue environment, because the information embedded within the environment plays an important role in understanding the referring relationships in the situated dialogue. Our approach combines referring expressions' semantic information along with dialogue history, task history, and a representation of the environment in which the dialogue is situated.

Reference resolution in dialogue has been investigated in recent years. Some of the previous work focuses on reference resolution in a multimodal setting (Chai et al., 2004; Liu et al., 2014; Liu et al., 2013; Krishnamurthy and Kollar, 2013; Matuszek et al., 2012). For this problem re-

[2] Typos and syntactic errors are shown as they appear in the original corpus.

searchers have used multimodal information, including vision, gestures, speech, and eye gaze, to contribute to the problem of reference resolution. Given that the focus of these works is on employing rich multimodal information, the research is usually conducted on a limited number of objects, and typically uses spatial relationship between objects as constraints to solve the reference resolution problem. We conduct reference resolution in an environment with a dynamic number of referents and there is no obvious spatial relationship between the objects.

More closely related work to our own involves reference resolution in dialogue situated within a collaborative game (Iida et al., 2010; Iida et al., 2011; Funakoshi et al., 2012). To link referring expressions to one of the seven gamepiece objects, they encoded dialogue history and task history, and our proposed approach leverages these features as well. However, in contrast to our complex problem-solving domain of computer programming, their domain has a small number of possible referents, so they used a manually created lexicon to extract semantic information from referring expressions. Funakoshi et al. (2012) went further, using Bayesian networks to model the relationship between referents and words used in referring expressions. That model is based on a hand-crafted concept dictionary and distribution over different referents. This approach cannot be directly applied to a dialogue with a dynamic environment because it is not possible to manually define the distribution over all possible referents beforehand, since objects in the environment are not known before they are created. So we chose Iida et al.'s work (2010) as one of the two most recent approaches to compare with.

Another closely related research direction involves reference resolution in physical environments (Kennington and Schlangen, 2015; Kushman et al., 2014; Naim et al., 2014; Liang et al., 2009). Although not within situated dialogue per se (because only one participant speaks), these lines of investigation have produced approaches that link natural language noun phrases to objects in an environment, such as a set of objects of different type and color on a table (Kennington and Schlangen, 2015) or a variable in a mathematical formula (Kushman et al., 2014). Some of these learn the mapping relationship by learning the semantics of words in the referring expressions (Kennington and Schlangen, 2015; Liang et al., 2009) with *referring expression-referent* pairs as input. Most recently, Kennington and Schlangen (2015) used a word-as-classifier approach to learn word semantics to map referring expressions to a set of 36 Pentomino puzzle pieces on a table. We implement their word-as-classifier approach and compare it with our novel approach.

3 Reference Resolution Approach

This section describes a new approach to reference resolution in situated dialogue. It links each referring expression from the dialogue to a most likely referent object in the environment. Our approach involves three main steps. First, referring expressions from the situated dialogue are segmented and labeled according to their semantic structure. Using a semantic segmentation and labeling approach we have previously developed (Li and Boyer, 2015), we use a conditional random field (CRF) for this joint segmentation and labeling task, and the values of the labeled attributes are then extracted (Section 3.1). The result of this step is *learned semantics*, which are attributes of objects expressed within each referring expression. Then, these learned semantics are utilized within the novel approach reported in this paper. As Section 3.2 describes, dialogue and task history are used to filter the objects in the environment to build a candidate list of referents, and then as Section 3.3 describes, a ranking-based classification approach is used to select the best matching referent.

For situated dialogue we define E_t as the state of the environment at time t. E_t consists of all objects present in the environment. Importantly, the objects in the environment vary along with the dialogue: at each moment, new objects could be created ($|E_t| > |E_{t-1}|$), and existing objects could be removed ($|E_t| < |E_{t-1}|$) because of the task performed by the user.

$$E_t = \{o_i | o_i \textit{ is an object in the environment at time } t\}$$

We assume that all of the objects o_i are observable in the environment. For example, in situated dialogues about programming, we can find all of the objects and extract their attributes using a source code parser. Then, reference resolution is defined as finding a best-matching o_i in E_t for referring expression RE.

3.1 Referring Expression Semantic Interpretation

In situated dialogues, a referring expression may contain rich semantic information about the referent, especially when the context of the situated dialogue is complex. Approaches such as domain-specific lexicons are limited in their ability to address this complexity, so we utilize a linear-chain CRF to parse the semantic structure of the referring expression. This more automated approach can also potentially avoid the manual labor required in creating and maintaining a lexicon.

In this approach, every object within the environment must be represented according to its attributes. We treat the set of all possible attributes of objects as a vector, and for each object o_i in the environment we instantiate and populate an attribute vector Att_Vec_i. For example, the attribute vector for a two-dimensional array in a computer program could be *[CATEGORY = ‘array, DIMENSION = ‘2, LINE = ‘30, NAME = ‘table, ...].* We ultimately represent $E_t = \{o_i\}$ as the set of all attribute vectors Att_Vec_i, and for a referring expression we aim to identify Att_Vec_j, the actual referent.

Since a referring expression describes its referents either implicitly or explicitly, the attributes expressed in it should match the attributes of its referent. We segment referring expressions and label the semantics of each segment using the CRF and the result is a set of segments, each of which represents some attribute of its referent. This process is illustrated in (Figure 2 (a)). After segmenting and labeling attributes in the referring expressions, the attribute *values* are extracted from each semantic segment using regular expressions (Figure 2 (b)), e.g., value *2* is extracted from *2 dimensional* to fill in the *ARRAY_DIM* element in an empty Att_Vec. The result is an attribute vector that represents the referring expression.

3.2 Generating a List of Candidate Referents

Once the referring expression is represented as an object attribute vector as described above, we wish to link that vector to the closest-matching object in the environment. Each object is represented by its own attribute vector, and there may be a large number of objects in E_t. Given a referring expression R_k, we would like to trim the list to keep only those objects that are likely to be referent for R_k.

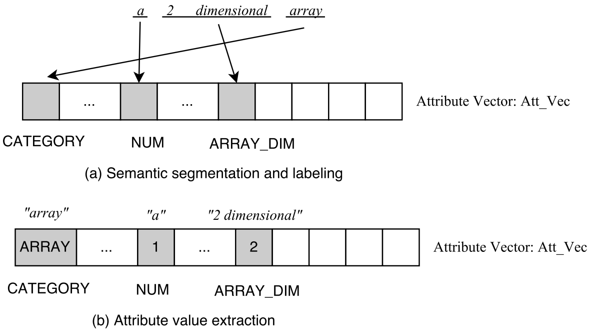

a 2 dimensional array
...
...
Attribute Vector: Att_Vec
CATEGORY
NUM
ARRAY_DIM
(a) Semantic segmentation and labeling
"array"
"a"
"2 dimensional"
ARRAY
...
1
...
2
Attribute Vector: Att_Vec
CATEGORY
NUM
ARRAY_DIM
(b) Attribute value extraction

Figure 2 Semantic interpretation of referring expressions.

There are two desired criteria for generating the list of candidate referents. First, the actual referent must be in the candidate list. At the same time, the candidate list should be as short as possible. We can pare down the set of all objects in E_t by considering focus of attention in dialogue. Early approaches performed reference resolution by estimating each dialogue participant's focus of attention (Lappin and Leass, 1994; Grosz et al., 1995). According to Ariel's accessibility theory (Ariel, 1988), people tend to use more precise descriptions such as proper names in referring expressions for referents in long term memory, and use less precise descriptions such as pronouns for referents in short term memory. In a precise description, there is more semantic information, while in a more vague description like a pronoun, there is less semantic information. Thus, these two sources of information, semantics and focus of attention, work together in identifying a referent.

Our approach employs this idea in the process of candidate referent selection by tracking the focus of attention of the dialogue participants from the beginning of the dialogue through dialogue history and task history, as has been done in prior work we use for comparison within our experiments (Iida et al., 2010). We also use the learned semantics of the referring expression (represented as the referring expression's attribute vector) as filtering conditions to select candidates.

The candidate generation process consists of three steps.

1. Candidate generation from dialogue history DH.

$$DH =< O_d, T_d >$$

Here, $O_d =< o_d^1, o_d^2, ..., o_d^m >$ is a sequence of objects that were mentioned since the beginning of the dialogue. $T_d =< t_d^1, t_d^2, ..., t_d^m >$ is a sequence of timestamps when corresponding objects were mentioned. All of the objects in E_t that were ever mentioned in the dialogue history, $\{o_i | o_i \in DH \ \& \ o_i \in E_t\}$, will also be added into the candidate list.

2. Candidate generation from task history TH. Similarly, $TH =< O_b, T_b >$, which is all of the objects in E_t that were ever manipulated by the user, will be added into the candidate list.

3. Candidate generation using learned semantics, which are the referent's attributes. Given a set of attributes extracted from a referring expression, all objects in E_t with one of the same attribute values will be added into the candidate list. The attributes are considered separately to avoid the case in which a single incorrectly extracted attribute could rule out the correct referent. Table 1 shows the algorithm used in this step.

```
Given a referring expression $R_k$, whose attribute vector $Att\_Vec_k$ has been extracted.
for each element $att_i$ of $Att\_Vec_k$
    if $att_i$ is not null
        for each $o$ in $E_t$
            if $att_i$ == $o.att_i$
                add $o$ into candidate list $C_k$
```

Table 1 Algorithm to select candidates using learned semantics

3.3 Ranking-based classification

With the list of candidate referents in hand, we employ a ranking-based classification model to identify the most likely referent. Ranking-based models have been shown to perform well for reference resolution problems in prior work (Denis and Baldridge, 2008; Iida et al., 2010). For a given referring expression R_k and its candidate referent list $C_k = \{o_1, o_2, ..., o_{N_k}\}$, in which each o_i is an object identified as a candidate referent, we compute the probability of each candidate o_i being the true referent of R_k, $p(R_k, o_i) = f(R_k, o_i)$, where f is the classification function. (Note that our approach is classifier-agnostic. As we describe in Section 5, we experimented with several different models.) Then, the candidates are ranked by $p(R_k, o_i)$, and the object with the highest probability is taken as the referent of R_k.

4 Corpus

Human problem solving represents a highly complex domain that poses great challenges for reference resolution. We evaluate our new reference resolution approach on a corpus of human-human textual dialogue in the domain of computer programming (Boyer et al., 2011). In each dialogue, a human tutor assisted a student remotely using typed dialogue as the student completed given programming tasks in the Java programming language. The programming tasks involved array manipulation and control flow, which are challenging for students with little programming experience. Students' and tutors' view of the task were synchronized in real time. At the beginning of each problem-solving session students were provided a framework of code to fill in, which is around 200 lines initially. The corpus contains 45 tutoring sessions, 4857 utterances in total, 108 utterances for each session on average. We manually annotated the referring expressions in the dialogue and their referents in the corresponding Java code for six dialogues from the corpus (346 referring expressions). These six sessions contain 758 utterances. The dialogues focus on the details of solving the programming problems, with very little social or off-task talk. Figure 1 shows an excerpt of this dialogue.

5 Experiments & Result

To evaluate the new approach, we performed a set of experiments that compare our approach with two state-of-the-art approaches.

5.1 Semantic Parsing

The referring expressions were extracted from the tutorial dialogues and their semantic segments and labels were manually annotated. A linear-chain CRF was trained on that data and used to perform referring expression segmentation and labeling (Li and Boyer, 2015). The current paper reports the first use of that learned semantics approach for reference resolution.

Next, we proceeded to extract the attribute values, a step that our previous work did not address. For the example shown in Figure 2 (b), from the learned semantic structure, we may know that *2 dimensional* refers to the dimension of the array, the attribute *ARRAY_DIM*. (In the current domain there are 14 attributes that comprise the generic attribute vector V, such as ARRAY_DIM, NUM, and CATEGORY.) To actually extract the attribute values, we use regular expressions that capture our three types of attribute values: categorical, numeric, and strings. For example, the value type of *CATEGORY* is categorical, like *method* or *variable*. Its values are taken from a closed set. *NAME* has values that are strings. *LINE_NUMBER*'s value is numeric. For categorical attributes, we add the categorical attribute values into the semantic tag set of the CRF used for segmentation. In this way, the attribute values of categorical attributes will be generated by the CRF. For attributes with text string values, we take the whole surface string of the semantic segment as its attribute value. The accuracy of the entire semantic parsing pipeline is 93.2% using 10-fold cross-validation. The accuracy is defined as the percentage of manually labeled attribute values that were successfully extracted from referring expressions.

5.2 Candidate Referent Generation

We applied the approach described in Section 3.2 on each session to generate a list of candidate referents for each referring expression. In a program, there could be more than one appearance of the same object. We take all of the appearances of the same object to be the same, since they all refer to the same artifact in the program. The average number of generated candidates for each referring expression was 44.8. The percentage of referring expressions whose actual referents were in the generated candidate list, or '"hit rate" is 90.5%, based on manual tagging. This performance indicates that the candidate referent list generation performs well.

A referring expression could be a pronoun, such as "it" or "that", which does not contain attribute information. In previous reference resolution research, it was shown that training separate models for different kinds of referring expressions could improve performance (Denis and Baldridge, 2008). We follow this idea and split the dataset into two groups: referring expressions containing attributes, RE_{att}, (270 referring expressions), and referring expressions that do not contain attributes, RE_{non} (76 referring expressions).

The candidate generation approach performed better for the referring expressions without attributes (hit rate 94.7%), compared to referring expressions with attributes (hit rate 89.3%). Since the candidate list for referring expressions without attributes relies solely on dialogue and task history, 94.7% of those referents had been mentioned in the dialogue or manipulated by the user previously. For referring expressions with attribute information, the generation of the candidate list also used learned semantic information. Only 70.0% of those referents had been mentioned in the dialogue or manipulated by the user before.

5.3 Identifying Most Likely Referent

We applied the approach described in section 3.3 to perform reference resolution on the corpus of tutorial dialogue. The data from the six manually labeled Java tutoring sessions were split into a training set and a test set. We used leave-one-dialogue-out cross validation (which leads to six folds) for the reference resolution experiments. In each fold, annotated referring expressions from one of the tutoring sessions were taken as the test set, and data from the other five sessions were the training set. We tested logistic regression, decision tree, naive Bayes, and neural networks as classifiers to compute the $p(R_k, o_i)$ for each *(referring expression, candidate)* pair for the ranking-based model. The features provided to each classifier are shown in Table 2.

To evaluate the performance of the new approach, we compare against two other recent approaches. First, we compare against a ranking-based model that uses dialogue history and task history features (Iida et al., 2010). This model uses semantics from a domain-specific lexicon instead of a semantic parser. (As described in Section 2, their work was extended by Funakoshi et al. (2012), but that work relies upon a handcrafted probability distribution of referents to concepts, which is not feasible in our domain since it has no fixed set of possible referents.) Therefore, we compare against their 2010 approach, implementing it in a way that creates the strongest possible baseline: we built a lexicon directly from our manually labeled semantic segments. First, we split all of the semantic segments into groups by their tags. Then, for each group of segments, any token that appeared twice or more was added into the lexi-

Learned Semantic Features (SF)
SF1: whether RE has CATEGORY attribute
SF2: whether RE.CATEGORY == o.CATEGORY
SF3: whether RE has RE.NAME
SF4: whether RE.NAME == o.NAME
SF5: RE.NAME ≈ o.NAME
SF6: RE.VAR_TYPE exist
SF7: RE.VAR_TYPE == o.VAR_TYPE
SF8: RE.LINE_NUMBER exist
SF9: RE.LINE_NUMBER == o.LINE_NUMBER
SF10: RE.ARRAY_DIMENSION exist
SF11: RE.ARRAY_DIMENSION == o.ARRAY_DIMENSION
SF12: CATEGORY of o
Dialogue History (DH) Features
DH1: whether o is the latest mentioned object
DH2: whether o was mentioned in the last 30 seconds
DH3: whether o was mentioned in the last [30, 60] seconds
DH4: whether o was mentioned in the last [60, 180] seconds
DH5: whether o was mentioned in the last [180, 300] seconds
DH6: whether o was mentioned in the last [300, 600] seconds
DH7: whether o was mentioned in the last [600, infinite] seconds
DH8: whether o was never mentioned from the beginning
DH9: String matching between o and RE
Task History (TH) Features
TH1: whether o is the most recent object manipulated
TH2: whether o was manipulated in the last 30 seconds
TH3: whether o was manipulated in the last [30, 60] seconds
TH4: whether o was manipulated in the last [60, 180] seconds
TH5: whether o was manipulated in the last [180, 300] seconds
TH6: whether o was manipulated in the last [300, 600] seconds
TH7: whether o was manipulated in the last [600, infinite] seconds
TH8: whether o was never manipulated from the beginning
TH9: whether o is in the current working window

Table 2 Features used for segmentation and labeling.

con. Although the necessary data to do this would not be available in a real application of the technique, it ensures that the lexicon for the baseline condition has good coverage and creates a high baseline for our new approach to compare against. Additionally, for fairness of comparison, for each semantic feature used in our model, we extracted the same feature using the lexicon. There were three kinds of attribute values in the domain: categorical, string, and numeric (as described in Section 5.1). We extracted categorical attribute values using the appearance of tokens in the lexicon. We used regular expressions to determine whether a referring expression contains the name of a candidate referent. We also used regular expressions to extract attribute values from referring expressions, such as line number. We also provided the *Iida* baseline model (2010) with a feature to indicate string matching between referring expressions and candidate referents, since this feature was captured in our model as an attribute.

We also compared our approach against a very recent technique that leveraged a word-as-classifier approach to learn semantic compatibility between referring expressions and candidate referents (Kennington and Schlangen, 2015). To create this comparison model we used a word-as-classifier to learn the semantics of referring expressions instead of CRF. This weakly supervised approach relies on co-appearance between words and object's attributes. We then used the resulting semantic compatibility in a ranking-based model to select the most likely referent.

The three conditions for our experiment are as follows.

- *Iida Baseline Condition*: Features including dialogue history, task history, and **semantics from a handcrafted lexicon** (Iida et al., 2010).

- *Kennington Baseline Condition*: Features including dialogue history, task history, and **learned semantics from a word-as-classifier model** (Kennington and Schlangen, 2015).

- *Proposed approach*: Features including dialogue history, task history, and **learned semantics from CRF**.

Within each of these experimental conditions, we varied the classifier used to compute $p(R_k, o_i)$, testing four classifiers: logistic regression (LR), decision tree (DT), naive Bayes (NB), and neural network (NN). The neural network has one hidden layer and the best-performing number of perceptrons was 100 (we experimented between 50 and 120).

To measure the performance of the reference resolution approaches, we analyzed accuracy, defined to be the percent of referring expressions that were successfully linked to their referents. We chose accuracy for our metric following standard practice (Iida et al., 2010; Kennington and Schlangen, 2015) because it provides an overall measure of the number of (R_k, o_i) pairs that were correctly identified. For the rare cases in which one referring expression referred to multiple referents, the output referent of the algorithm was taken as correct if it selected any of the multiple referents.

The results are shown in Table 3. We focus on comparing the results on referring expressions that contain attribute information, shown in the table as REF_{ATT}. REF_{ATT} accounts for 78% of all of the cases (270 out of 346). Among the three approaches, our proposed approach outperformed both prior approaches. Compared to the Iida 2010 approach which achieved a maximum of 55.2% accuracy, our approach achieved 68.5% accuracy using a neural net classifier, and this difference is statistically significant based on the results of a Wilcoxon signed-rank test ($n = 6$; $p = 0.046$). Our approach outperformed the Kennington 2015 approach even more substantially, as its best performance was 46.3% accuracy ($p = 0.028$). Intuitively, the better performance of our model compared to the Iida approach is due to its ability to more accurately model referring expressions' semantics. Compared to a lexicon, semantic parsing finds optimal segmentation for a referring expression, while a lexicon approach extracts different attribute information from referring expressions separately. Note that our approach and the Iida 2010 approach achieved the same performance on REF_{NON} referring expressions. Since these referring expressions do not contain attribute information, these two approaches used the same set of features.

Interestingly, the model using a word-as-classifier approach to learn the semantic compatibility between referring expressions and referent's attributes performs the worst. We believe that the reason for this poor performance is mainly from the way it performs semantic compositions. It cannot learn structures in referring expressions, such as that *2 dimensional* is a segment, *dimensional* represents the type of the attribute, and *2* is the value of the attribute. The word-as-classifier model cannot deal with this complex semantic composition.

The results reported above relied on learned semantics. We also performed experiments using manually labeled, gold-standard semantics of referring expressions. The result in Table 4 shows that ranking-based models have the potential to achieve a considerably better result, 73.6%, with more accurate semantic information. Given the 85.3% agreement between two human annotators, the model performs very well, since the semantics of whole utterances in situated dialogue also play a very important role in identifying a given referring expression's referent.

experimental condition	$f(R_k, o_i)$ classifier	**accuracy**	
		REF_{ATT}	REF_{NON}
Iida 2010	LR	0.500	0.440
	DT	0.537	0.453
	NB	0.466	0.413
	NN	**0.552**	0.373
Kennington 2015	**LR**	**0.4627**	0.3867
	DT	0.3769	0.3333
	NB	0.3209	0.4000
	NN	0.4216	0.4000
Our approach	LR	0.631	0.440
	DT	0.631	0.453
	NB	0.493	0.413
	NN	**0.685**	0.373

Table 3 Reference resolution results.

models	**accuracy**	
	REF_{ATT}	REF_{NON}
LR + SEM_gold	0.684	0.429
DT + SEM_gold	0.643	0.429
NB + SEM_gold	0.511	0.377
NN + SEM_gold	**0.736**	0.325

Table 4 Reference resolution results with gold semantic labels.

6 Conclusion

Dialogue systems need to move toward supporting users in increasingly complex tasks. To do this effectively, accurate reference resolution is crucial. We have presented a new approach that applies learned semantics to reference resolution in situated dialogue for collaborative tasks. The experiments with human-human dialogue on a collaborative programming task showed a tremendous improvement using semantic information that was learned with a CRF-based semantic parsing approach compared to the previous state-of-art approaches. The accuracy was improved substantially, from 55.2% to 68.5%.

There are several important future research directions in reference resolution for situated dialogues. First, models should incorporate more semantic information from discourse structure and utterance understanding besides semantics from referring expressions. This is illustrated by the observation that the reference resolution accuracy using gold-standard semantic information from referring expressions is still substantially lower than the agreement rate between human annotators. Another research direction that holds promise is to use an unsupervised approach to extract semantic information from referring expressions. It is hoped that this line of investigation will enable rich natural language dialogue interactions to support users in a wide variety of complex situated tasks.

Acknowledgments

The authors wish to thank the members of the LearnDialogue group at the University of Florida and North Carolina State University, particularly Joseph Wiggins, along with Wookhee Min and Adam Dalton for their helpful input. This work is supported in part by Google through a Capacity Research Award and by the National Science Foundation through grant CNS-1622438. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the authors, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

References

Ariel, Mira. 1988. Referring and Accessibility. *Journal of Linguistics*, 24, 65-87.

Björkelund, Anders and Kuhn, Jonas. 2014. Learning Structured Perceptrons for Coreference Resolution with Latent Antecedents and Non-local Features. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 47-57.

Boyer, Kristy Elizabeth and Ha, Eun Young and Phillips, Robert and Lester, James C.. 2011. The Impact of Task-Oriented Feature Sets on HMMs for Dialogue Modeling. *Proceedings of the 12th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 49–58.

Chai, Joyce and Hong, Pengyu and Zhou, Michelle. 2004. A Probabilistic Approach to Reference Resolution in Multimodal User Interfaces. *Proceedings of the 9th International Conference on Intelligent User Interfaces SE - IUI '04*, 70-77.

Denis, Pascal and Baldridge, Jason. 2008. Specialized Models and Reranking for Coreference Resolution. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 660-669.

Devault, David and Kariaeva, Natalia and Kothari, Anubha and Oved, Iris and Stone, Matthew. 2005. An Information-State Approach to Collaborative Reference. *Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions*, 1-4.

Funakoshi, Kotaro and Nakano, Mikio and Tokunaga, Takenobu and Iida, Ryu. 2012. A Unified Probabilistic Approach to Referring Expressions. *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 237-246.

Graesser, A C and Lu, S and Jackson, G T and Mitchell, H H and Ventura, M and Olney, A and Louwerse, M M. 2004. AutoTutor: A Tutor with Dialogue in Natural Language. *Behavior Research Methods, Instruments, & Computers*, 36, 180-192.

Grosz, B J and Weinstein, S and Joshi, A K. 1995. Centering - a Framework for Modeling the Local Coherence of Discourse. *Computational Linguistics*, 21(2), 203-225.

Grosz, Barbara J and Sidner, Candace L. 1986. Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, 175-204.

Harabagiu, Sanda M. and Räzvan C. Bunescu and Maiorano, Steven J.. 2001. Text and Knowledge Minding for Coreference Resolution. *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies. (NAACL-HLT)*, 1-8.

Heeman, Peter a. and Hirst, Graeme. 1995. Collaborating on Referring Expressions. *Computational Linguistics*, 21, 351-382.

Huwel, Sonja and Wrede, Britta. 2006. Spontaneous Speech Understanding for Robust Multimodal Human-robot Communication. *Proceedings of the COLING/ACL*, 391-398.

Iida, Ryu and Kobayashi, Shumpei and Tokunaga, Takenobu. 2010. Incorporating Extra-linguistic Information into Reference Resolution in Collaborative Task Dialogue. *Proceedings of the 48th Annual*

Meeting of the Association for Computational Linguistics, 1259-1267.

Iida, Ryu and Yasuhara, Masaaki and Tokunaga, Takenobu. 2011. Multi-modal Reference Resolution in Situated Dialogue by Integrating Linguistic and Extra-Linguistic Clues. *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*, 84-92.

Kennington, Casey and Schlangen, David. 2015. Simple Learning and Compositional Application of Perceptually Grounded Word Meanings for Incremental Reference Resolution. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 292-301.

Krishnamurthy, Jayant and Kollar, Thomas. 2013. Jointly Learning to Parse and Perceive: Connecting Natural Language to the Physical World. *Association for Computational Linguistics*, 193-206.

Kruijff, Geert-Jan M and Lison, Pierre and Benjamin, Trevor. 2010. Situated dialogue processing for human-robot interaction. *Cognitive Systems Monographs*, 8, 311-364.

Kushman, Nate and Artzi, Yoav and Zettlemoyer, Luke and Barzilay, Regina. 2014. Learning to Automatically Solve Algebra Word Problems. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 271-281.

Lappin, Shalom and Leass, Herbert J.. 1994. An Algorithm for Pronominal Anaphora Resolution. *Computational Linguistics*, 535-561.

Li, Xiaolong and Boyer, Kristy Elizabeth. 2015. Semantic Grounding in Dialogue for Complex Problem Solving. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics and Human Language Technology (NAACL HLT)*, 841-850.

Liang, Percy and Jordan, Michael I and Klein, Dan. 2009. Learning Semantic Correspondences with Less Supervision. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 91–99.

Liu, Changsong and Chai, Joyce Y. 2015. Learning to Mediate Perceptual Differences in Situated Human-Robot Dialogue. *Proceedings of AAAI 2015*, 2288-2294.

Liu, Changsong and She, Lanbo and Fang, Rui and Chai, Joyce Y. 2014. Probabilistic Labeling for Efficient Referential Grounding Based On Collaborative Discourse. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 13-18.

Liu, Changsong and Fang, Rui and She, Lanbo and Chai, Joyce. 2013. Modeling Collaborative Referring for Situated Referential Grounding. *Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 78–86.

Manning, Christopher D and Bauer, John and Finkel, Jenny and Bethard, Steven J. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 55-60.

Martschat, Sebastian and Strube, Michael. 2015. Latent Structures for Coreference Resolution. *Transactions of the Association for Computational Linguistics*, 3, 405-418.

Matuszek, Cynthia and FitzGerald, Nicholas and Zettlemoyer, Luke and Liefeng, Bo and Fox, Dieter. 2012. A Joint Model of Language and Perception for Grounded Attribute Learning. *Proceedings of the 29th International Conference on Machine Learning*, 1671-1678.

McCarthy, Joseph F. and Lehnert, Wendy G.. 1995. Using Decision Trees for Coreference Resolution. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 1-5.

Naim, I and Song, Yc and Liu, Q and Kautz, H and Luo, J and Gildea, D. 2014. Unsupervised Alignment of Natural Language Instructions with Video Segments. *Proceedings of AAAI 2014*, 1558-1564.

Ponzetto, Simone Paolo and Strube, Michael. 2006. Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution. *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 192-199.

Soon, Wee Meng and Ng, Hwee Tou and Lim, Daniel Chung Yong. 2001. Incorporating Extra-linguistic Information into Reference Resolution in Collaborative Task Dialogue. *Computational Linguistics*, 27, 521-544.

VanLehn, Kurt and Jordan, P W and Rosé, C P and Bhembe, D and Bottner, M and Gaydos, A and Makatchev, M and Pappuswamy, U and Ringenberg, M and Roque, A. 2002. The Architecture of Why2-Atlas: A Coach for Qualitative Physics Essay Writing. *Proceedings of the Sixth International Conference on Intelligent Tutoring System*, 2363, 158-167.