SemEval-2014 Task 6: Supervised Semantic Parsing of Robotic Spatial Commands

Kais Dukes
School of Computing, University of Leeds
Leeds LS2 9JT, United Kingdom
sckd@leeds.ac.uk

Abstract

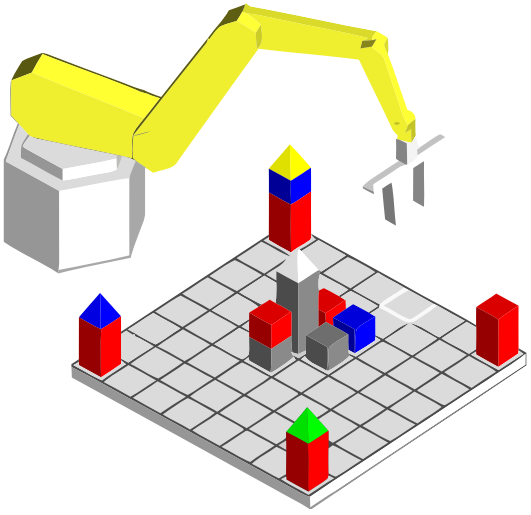
SemEval-2014 Task 6 aims to advance semantic parsing research by providing a high-quality annotated dataset to compare and evaluate approaches. The task focuses on contextual parsing of robotic commands, in which the additional context of spatial scenes can be used to guide a parser to control a robot arm. Six teams submitted systems using both rule-based and statistical methods. The best performing (hybrid) system scored 92.5% and 90.5% for parsing with and without spatial context. However, the best performing statistical system scored 87.35% and 60.84% respectively, indicating that generalized understanding of commands given to a robot remains challenging, despite the fixed domain used for the task.

1 Introduction

Semantic parsers analyze sentences to produce formal meaning representations that are used for the computational understanding of natural language. Recently, state-of-the-art semantic parsing methods have used for a variety of applications, including question answering (Kwiatkowski et al., 2013; Krishnamurthy and Mitchell, 2012), dialog systems (Artzi and Zettlemoyer, 2011), entity relation extraction (Kate and Mooney, 2010) and robotic control (Tellex, 2011; Kim and Mooney, 2012).

Different parsers can be distinguished by the level of supervision they require during training. Fully supervised training typically requires an annotated dataset that maps natural language (NL) to a formal meaning representation such as logical form. However, because annotated data is often not available, a recent trend in semantic parsing research has been to eschew supervised training in favour of either unsupervised or weakly-supervised methods that utilize additional information. For example, Berant and Liang (2014) use a dataset of 5,810 question-answer pairs without annotated logical forms to induce a parser for a question-answering system. In comparison, Poon (2013) converts NL questions into formal queries via indirect supervision through database interaction.

In contrast to previous work, the shared task described in this paper uses the Robot Commands Treebank (Dukes, 2013a), a new dataset made available for supervised semantic parsing. The chosen domain is robotic control, in which NL commands are given to a robot arm used to manipulate shapes on an 8 x 8 game board. Despite the fixed domain, the task is challenging as correctly parsing commands requires understanding spatial context. For example, the command in Figure 1 may have several plausible interpretations, given different board configurations.



'*Move the pyramid on the blue cube on the gray one.*'

Figure 1: Example scene with a contextual spatial command from the Robot Commands Treebank.

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 45–53,
Dublin, Ireland, August 23-24, 2014.

The task is inspired by the classic AI system SHRLDU, which responded to NL commands to control a robot for a similar game board (Winograd, 1972), although that system is reported to not have generalized well (Dreyfus, 2009; Mitkov, 1999). More recent research in command understanding has focused on parsing jointly with *grounding*, the process of mapping NL descriptions of entities within an environment to a semantic representation. Previous work includes Tellex et al. (2011), who develop a small corpus of commands for a simulated fork lift robot, with grounding performed using a factor graph. Similarly, Kim and Mooney (2012) perform joint parsing and grounding using a corpus of navigation commands. In contrast, this paper focuses on parsing using additional situational context for disambiguation and by using a larger NL dataset, in comparison to previous robotics research.

In the remainder of this paper, we describe the task, the dataset and the metrics used for evaluation. We then compare the approaches used by participant systems and conclude with suggested improvements for future work.

2 Task Description

The long term research goal encouraged by the task is to develop a system that will robustly execute NL robotic commands. In general, this is a highly complex problem involving computational processing of language, spatial reasoning, contextual awareness and knowledge representation. To simplify the problem, participants were provided with additional tools and resources, allowing them to focus on developing a semantic parser for a fixed domain that would fit into an existing component architecture. Figure 2 shows how these components interact.

Semantic parser: Systems submitted by participants are *semantic parsers* that accept an NL command as input, mapping this to a formal Robot Control Language (RCL), described further in section 3.3. The Robot Commands Treebank used for the both training and evaluation is an annotated corpus that pairs NL commands with contextual RCL statements.

Spatial planner: A *spatial planner* is provided as an open Java API[1]. Commands in the treebank are specified in the context of spatial scenes. By interfacing with the planner, participant systems

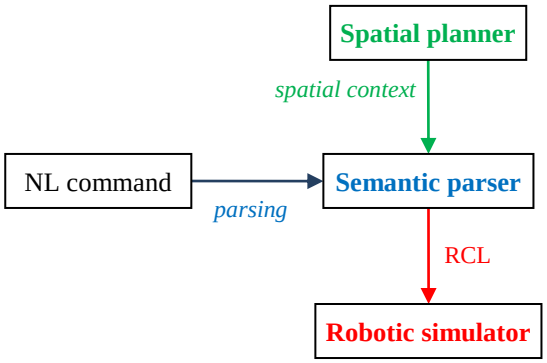


Figure 2: Integrated command understanding system.

have access to this additional information. For example, given an RCL fragment for the expression '*the red cube on the blue block*', the planner will ground the entity, returning a list of zero or more board coordinates corresponding to possible matches. The planner also validates commands to determine if they are compatible with spatial context. It can therefore be used to constrain the search space of possible parses, as well as enabling early resolution of attachment ambiguity during parsing.

Robotic simulator: The simulated environment consists of an 8 x 8 board that can hold prisms and cubes which occur in eight different colors. The robot's gripper can move to any discrete position within an 8 x 8 x 8 space above the board. The planner uses the simulator to enforce physical laws within the game. For example, a block cannot remain unsupported in empty space due to gravity. Similarly, prisms cannot lie below other block types. In the integrated system, the parser uses the planner for context, then provides the final RCL statement to the simulator which executes the command by moving the robot arm to update the board.

3 Data

3.1 Data Collection

For the shared task, 3,409 sentences were selected from the treebank. This data size compares with related corpora used for semantic parsing such as the ATIS (Zettlemoyer and Collins, 2007), GeoQuery (Kate et al., 2005), Jobs (Tang and Mooney, 2001) and RoboCup (Kuhlmann et al., 2004) datasets, consisting of 4,978; 880; 640 and 300 sentences respectively.

The treebank was developed via a game with a purpose (`www.TrainRobots.com`), in which players were shown *before* and *after* configurations

[1] `https://github.com/kaisdukes/train-robots`

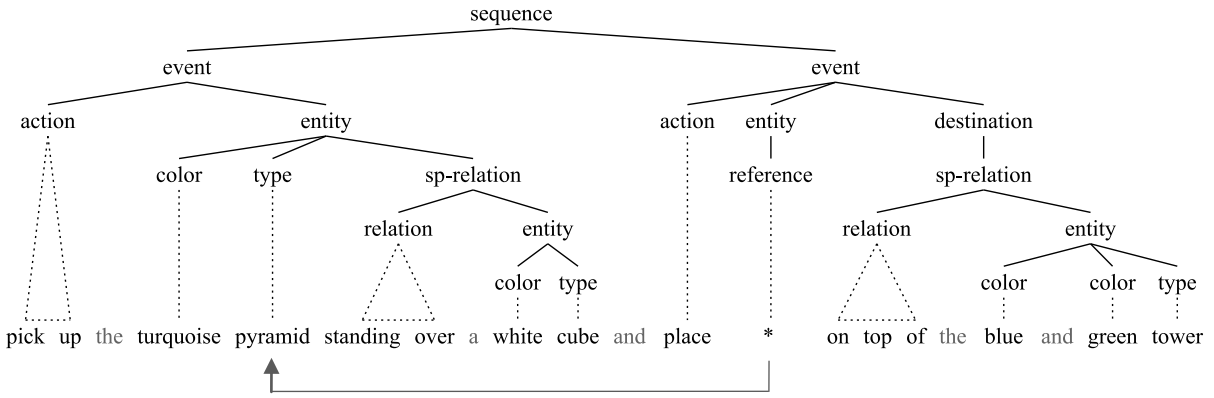


Figure 3: Semantic tree from the treebank with an elliptical anaphoric node and its annotated antecedent.

and asked to give a corresponding command to a hypothetical robot arm. To make the game more competitive and to promote data quality, players rated each other's sentences and were rewarded with points for accurate entries (Dukes, 2013b).

3.2 Annotation

In total, over 10,000 commands were collected through the game. During an offline annotation phase, sentences were manually mapped to RCL. However, due to the nature of the game, players were free to enter arbitrarily complex sentences to describe moves, not all of which could be represented by RCL. In addition, some commands were syntactically well-formed, but not compatible with the corresponding scenes. The 3,409 commands selected for the task had RCL statements that were both understood by the planner and when given to the robotic simulator resulted in the expected move being made between *before* and *after* board configurations. Due to this extra validation step, all RCL statements provided for the task were contextually well-formed.

```
(sequence:
  (event:
    (action: take)
    (entity:
      (id: 1)
      (color: cyan)
      (type: prism)
      (spatial-relation:
        (relation: above)
        (entity:
          (color: white)
          (type: cube)))))
  (event:
    (action: drop)
    (entity:
      (type: reference)
      (reference-id: 1))
    (destination:
      (spatial-relation:
        (relation: above)
        (entity:
          (color: blue)
          (color: green)
          (type: stack))))))
```

Figure 4: RCL representation with co-referencing.

3.3 Robot Control Language

RCL is a novel linguistically-oriented semantic representation. An RCL statement is a semantic tree (Figure 3) where leaf nodes generally align to words in the corresponding sentence, and non-leaves are tagged using a pre-defined set of categories. RCL is designed to annotate rich linguistic structure, including ellipsis (such as 'place [it] on'), anaphoric references ('it' and 'one'), multi-word spatial expressions ('on top of') and lexical disambiguation ('one' and 'place'). Due to ellipsis, unaligned words and multi-word expressions, a leaf node may align to zero, one or more words in a sentence. Figure 4 shows the RCL syntax for the tree in Figure 3, as accepted by the spatial planner and the simulator. As these components do not require NL word alignment data, this additional information was made available to task participants for training via a separate Java API.

The tagset used to annotate RCL nodes can be divided into general tags (that are arguably applicable to other domains) and specific tags that were customized for the domain in the task (Tables 1 and 2 overleaf, respectively). The general elements are typed *entities* (labelled with semantic features) that are connected using *relations* and *events*. This universal formalism is not domain-specific, and is inspired by semantic frames (Fillmore and Baker, 2001), a practical representation used for NL understanding systems (Dzikovska, 2004; UzZaman and Allen, 2010; Coyne et al., 2010; Dukes, 2009).

In the remainder of this section we summarize aspects of RCL that are relevant to the task; a

more detailed description is provided by Dukes (2013a; 2014). In an RCL statement such as Figure 4, a preterminal node together with its child leaf node correspond to a feature-value pair (such as the feature *color* and the constant *blue*). Two special features which are distinguished by the planner are *id* and *reference-id*, which are used for co-referencing such as for annotating anaphora and their antecedents. The remaining features model the simulated robotic domain. For example, the values of the *action* feature are the moves used to control the robotic arm, while values of the *type* and *relation* features are the entity and relation types understood by the spatial planner (Table 2). As well as qualitative relations (such as 'below' or 'above'), the planner also accepts spatial relations that include quantitative measurements, such as in 'two squares left of the red prism' (Figure 5).

RCL Element	Description
action	Aligned to a verbal group in NL, e.g. 'drop' or 'pick up'.
cardinal	Number (e.g. 2 or 'three').
color	Colored attribute of an entity.
destination	A spatial destination.
entity	Entity within the domain.
event	Specification of a command.
id	Id for anaphoric references.
indicator	Spatial attribute of an entity.
measure	Used for distance metrics.
reference-id	A resolved reference.
relation	Relation type (e.g. 'above').
sequence	Used to specify a sequence of events or statements.
spatial-relation	Used to specify a spatial relation between two entities or to describe a location.
type	Used to specify an entity type.

Table 1: Universal semantic elements in RCL.

Category	Values
Actions	move, take, drop
Relations	left, right, above, below, forward, backward, adjacent, within, between, nearest, near, furthest, far, part
Indicators	left, leftmost, right, rightmost, top, highest, bottom, lowest, front, back, individual, furthest, nearest, center
entity types	cube, prism, corner, board stack, row, column, edge, tile, robot, region, reference, type-reference
Colors	blue, cyan, red, yellow, green, magenta, gray, white

Table 2: Semantic categories customized for the task.

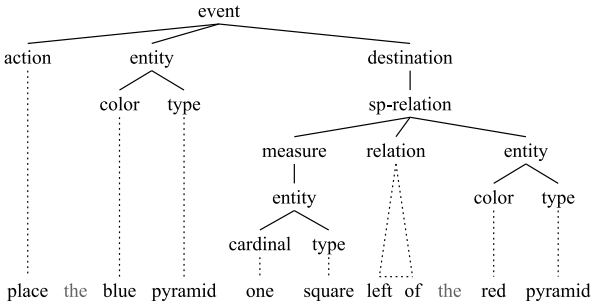


Fig.ure 5: A quantitative relation with a landmark.

RCL distinguishes between *relations* which relate entities and *indicators*, which are attributes of entities (such as 'left' in 'the left cube'). For the task, participants are asked to map NL sentences to well-formed RCL by identifying spatial relations and indicators, then parsing higher-level entities and events. Finally, a well-formed RCL tree with an event (or sequence of events) at top-level is given the simulator for execution.

4 Evaluation Metrics

Out of the 3,400 sentences annotated for the task, 2,500 sentences were provided to participants for system training. During evaluation, trained systems were presented with 909 previously unseen sentences and asked to generate corresponding RCL statements, with access to the spatial planner for additional context. To keep the evaluation process as simple as possible, each parser's output for a sentence was scored as correct if it exactly matched the expected RCL statement in the treebank. Participants were asked to calculate two metrics, *P* and *NP*, which are the proportion of exact matches with and without using the spatial planner respectively:

$$P = \frac{\#\ matches\ with\ planning}{\#\ sentences}$$

$$NP = \frac{\#\ matches\ without\ planning}{\#\ sentences}$$

System	Authors	Statistical?	Strategy	P	NP	NP - P
UW-MRS	Packard	Hybrid	Rule-based ERG + Berkeley parser	92.50	90.50	-2.00
AT&T Labs	Stoyanchev et al.	Statistical	Statistical maximum entropy parser	87.35	60.84	-26.51
RoBox	Evang and Bos	Statistical	CCG parser + structured perceptron	86.80	79.21	-7.59
Shrdlite	Ljunglöf	Rule-based	Hand crafted domain-specific grammar	86.10	51.50	-34.60
KUL-Eval	Mattelaer et al.	Statistical	CCG parser	71.29	57.76	-13.53
UWM	Kate	Statistical	KRISP parser	N/A	45.98	N/A

Table 3: System results for supervised semantic parsing of the Robot Commands Treebank (P = parsing with integrated spatial planning, NP = parsing without integrated spatial planning, NP - P = drop in performance without integrated spatial planning, N/A = performance not available).

These metrics contrast with measures for partially correct parsed structures, such as Parseval (Black et al., 1991) or the leaf-ancestor metric (Sampson and Babarczy, 2003). The rationale for using a strict match is that in the integrated system, a command will only be executed if it is completely understood, as both the spatial planner and the simulator require well-formed RCL.

5 Systems and Results

Six teams participated in the shared task using a variety of strategies (Table 3). The last measure in the table gives the performance drop without spatial context. The value *NP* - *P* = -2 for the best performing system suggests this as an upper bound for the task. The different values of this measure indicate the sensitivity to (or possibly reliance on) context to guide the parsing process. In the remainder of this section we compare the approaches and results of the six systems.

UW-MRS: Packard (2014) achieved the best score for parsing both with and without spatial context, at 92.5% and 90.5%, respectively, using a hybrid system that combines a rule-based grammar with the Berkeley parser (Petrov et al., 2006). The rule-based component uses the English Resource Grammar, a broad coverage hand-written HPSG grammar for English. The ERG produces a ranked list of Minimal Recursion Semantics (MRS) structures that encode predicate argument relations (Copestake et al., 2005). Approximately 80 rules were then used to convert MRS to RCL. The highest ranked result that is validated by the spatial planner was selected as the output of the rule-based system. Using this approach, Packard reports scores of *P* = 82.4% and *NP* = 80.3% for parsing the evaluation data.

To further boost performance, the Berkeley parser was used for back-off. To train the parser, the RCL treebank was converted to phrase structure by removing non-aligned nodes and inserting additional nodes to ensure one-to-one alignment with words in NL sentences. Performance of the Berkeley parser alone was *NP* = 81.5% (no *P*-measure was available as spatial planning was not integrated).

To combine components, the ERG was used initially, with fall back to the Berkeley parser when no contextually compatible RCL statement was produced. The hybrid approach improved accuracy considerably, with *P* = 92.5% and *NP* = 90.5%. Interestingly, Packard also performs precision and recall analysis, and reports that the rule-based component had higher precision, while the statistical component had higher recall, with the combined system outperforming each separate component in both precision and recall.

AT&T Labs Research: The system by Stoyanchev et al. (2014) scored second best for contextual parsing and third best for parsing without using the spatial planner (*P* = 87.35% and *NP* = 60.84%). In contrast to Packard's UW-MRS submission, the AT&T system is a combination of three statistical models for tagging, parsing and reference resolution. During the tagging phase, a two-stage sequence tagger first assigns a part-of-speech tag to each word in a sentence, followed by an RCL feature-value pair such as *(type: cube)* or *(color: blue)*, with unaligned words tagged as 'O'. For parsing, a constituency parser was trained using non-lexical RCL trees. Finally, anaphoric references were resolved using a maximum entropy feature model. When combined, the three components generate a list of weighted RCL trees, which are filtered by the spatial planner. Without integrated planning, the most-probable parse tree is selected.

In their evaluation, Stoyanchev et al. report accuracy scores for the separate phases as well as for the combined system. For the tagger, they report an accuracy score of 95.2%, using the

standard split of 2,500 sentences for training and 909 for evaluation. To separately measure the joint accuracy of the parser together with reference resolution, gold-standard tags were used resulting in a performance of *P* = 94.83% and *NP* = 67.55%. However, using predicted tags, the system's final performance dropped to *P* = 87.35% and *NP* = 60.84%. To measure the effect of less supervision, the models were additionally trained on only 500 sentences. In this scenario, the tagging model degraded significantly, while the parsing and reference resolution models performed nearly as well.

RoBox: Using Combinatory Categorial Grammar (CCG) as a semantic parsing framework has been previously shown to be suitable for translating NL into logical form. Inspired by previous work using a CCG parser in combination with a structured perceptron (Zettlemoyer and Collins, 2007), RoBox (Evang and Bos, 2014) was the best performing CCG system in the shared task scoring *P* = 86.8% and *NP* = 79.21%.

Using a similar approach to UW-MRS for its statistical component, RCL trees were interpreted as phrase-structure and converted to CCG derivations for training. During decoding, RCL statements were generated directly by the CCG parser. However, in contrast to the approach used by the AT&T system, RoBox interfaces with the planner during parsing instead of performing spatial validation a post-processing step. This enables early resolution of attachment ambiguity and helps constrain the search space. However, the planner is only used to validate *entity* elements, so that *event* and *sequence* elements were not validated. As a further difference to the AT&T system, anaphora resolution was not performed using a statistical model. Instead, multiple RCL trees were generated with different candidate anaphoric references, which were filtered out contextually using the spatial planner.

RoBox suffered only a 7.59% absolute drop in performance without using spatial planning, second only to UW-MRS at 2%. Evang and Bos perform error analysis on RoBox and report that most errors relate to ellipsis, the ambiguous word *one*, anaphora or attachment ambiguity. They suggest that the system could be improved with better feature selection or by integrating the CCG parser more closely with the spatial planner.

Shrdlite: The Shrdlite system by Ljunglöf (2014), inspired by the Classic SHRDLU system by Winograd (1972), is a purely rule-based system that was shown to be effective for the task. Scoring *P* = 86.1% and *NP* = 51.5%, Shrdlite ranked fourth for parsing with integrated planning, and fifth without using spatial context. However, it suffered the largest absolute drop in performance without planning (34.6 points), indicating that integration with the planner is essential for the system's reported accuracy.

Shrdlite uses a hand-written compact unification grammar for the fragment of English appearing in the training data. The grammar is small, consisting of only 25 grammatical rules and 60 lexical rules implemented as a recursive-descent parser in Prolog. The lexicon consists of 150 words (and multi-word expressions) divided into 23 lexical categories, based on the RCL preterminal nodes found in the treebank. In a post-processing phase, the resulting parse trees are normalized to ensure that they are well-formed by using a small set of supplementary rules.

However, the grammar is highly ambiguous resulting in multiple parses for a given input sentence. These are filtered by the spatial planner. If multiple parse trees were found to be compatible with spatial context (or when not using the planner), the tree with the smallest number of nodes was selected as the parser's final output. Additionally, because both the training and evaluation data were collected via crowdsourcing, sentences occasionally contain spelling errors, which were intentionally included in the task. To handle misspelt words, Shrdlite uses Levenshtein edit distance with a penalty to reparse sentences when the parser initially fails to produce any analysis.

KUL-Eval: The CCG system by Mattelaer et al. (2014) uses a different approach to the RoBox system described previously. KUL-Eval scored *P* = 71.29% and *NP* = 57.76% in comparison to the RoBox scores of *P* = 86.8% and *NP* = 79.21%.

During training, the RCL treebank was converted to λ-expressions. This process is fully reversible, so that no information in an RCL tree is lost during conversion. In contrast to RoBox, but in common with the AT&T parser, KUL-Eval performs spatial validation as a post-processing step and does not integrate the planner directly into the parsing process. A probabilistic CCG is used for parsing, so that multiple λ-expressions are returned (each with an associated confidence measure) that are translated into RCL. Finally, in the validation step, the spatial planner is used to discard RCL statements that are incompatible with spatial context and the remaining most-probable parse is returned as the system's output.

Mattelaer et al. note that in several cases the parser produced partially correct statements but that these outputs did not contribute to the final score, given the strictly matching measures used for the *P* and *NP* metrics. However, well-formed RCL statements are required by the spatial planner and robotic simulator for the integrated system to robustly execute the specified NL command. Partially correct structures included statements which almost matched the expected RCL tree with the exception of incorrect feature-values, or the addition or deletion of nodes. The most common errors were feature-values with incorrect entity types (such as 'edge' and 'region') and mismatched spatial relations (such as confusing 'above' and 'within' and confusing 'right', 'left' and 'front').

UWM: The UWM system submitted by Kate (2014) uses an existing semantic parser, KRISP, for the shared task. KRISP (Kernel-based Robust Interpretation for Semantic Parsing) is a trainable semantic parser (Kate and Mooney, 2006) that uses Support Vector Machines (SVMs) as the machine learning method with a string subsequence kernel. As well as training data consisting of RCL paired with NL commands, KRISP required a context-free grammar for RCL, which was hand-written for UWM. During training, *id* nodes were removed from the RCL trees. These were recovered after parsing in a post-processing phase to resolve anaphora by matching to the nearest preceding antecedent.

In contrast to other systems submitted for the task, UWM does not interface with the spatial planner and parses purely non-contextually. Because the planner was not used, the system's accuracy was negatively impacted by simple issues that may have been easily resolved using spatial context. For example, in RCL, the verb 'place' can map to either *drop* or *move* actions, depending on whether or not a block is held in the gripper in the corresponding spatial scene. Without using spatial context, it is hard to distinguish between these cases during parsing.

The system scored a non-contextual measure of *NP* = 45.98%, with Kate reporting a 51.18% best F-measure (at 72.67% precision and 39.49% recall). No *P*-measure was reported as the spatial planner was not used. Due to memory constraints when training the SVM classifiers, only 1,500 out of 2,500 possible sentences were used from the treebank to build the parsing model. However, it may be possible to increasing the size of training data in future work through sampling.

6 Discussion

The six systems evaluated for the task employed a variety of semantic parsing strategies. With the exception of one submission, all systems interfaced with the spatial planner, either in a post-processing phase, or directly during parsing to enable early disambiguation and to help constrain the search space. An open question that remains following the task is how applicable these methods would be to other domains. Systems that relied heavily on the planner to guide the parsing process could only be adapted to domains for a which a planner could conceivably exist. For example, nearly all robotic tasks such as such as navigation, object manipulation and task execution involve aspects of planning. NL question-answering interfaces to databases or knowledge stores are also good candidates for this approach, since parsing NL questions into a semantic representation within the context of a database schema or an ontology could be guided by a query planner.

However, approaches with a more attractive *NP* - *P* measure (such as UW-MRS and RoBox) are arguably more easily generalized to other domains, as they are less reliant on a planner. Additionally, the usual arguments for rule-based systems verses supervised statistical systems apply to any discussion on domain adaptation: rule-based systems require human manual effort, while supervised statistical systems required annotated data for the new domain.

In comparing the best two statistical systems (AT&T and RoBox) it is interesting to note that these performed similarly with integrated planning (*P* = 87.35% and 86.80%, respectively), but differed considerably without planning (*NP* = 60.84% and 79.21%). As these two systems employed different parsers (a constituency parser and a CCG parser), it is difficult to perform a direct comparison to understand why the AT&T system is more reliant on spatial context. It would also be interesting to understand, in further work, why the two CCG-based systems differed considerably in their *P* and *NP* scores.

It is also surprising that the best performing system, UW-MRS, suffered only a 2% drop in performance without using the planner, demonstrating clearly that in the majority of sentences in the evaluation data, spatial context is not actually required to perform semantic parsing. Although as shown by the *NP* - *P* scores, spatial context can dramatically boost performance of certain approaches for the task when used.

7 Conclusion and Future Work

This paper described a new task for SemEval: Supervised Semantic Parsing of Robotic Spatial Commands. Despite its novel nature, the task attracted high-quality submissions from six teams, using a variety of semantic parsing strategies.

It is hoped that this task will reappear at SemEval. Several lessons were learnt from this first version of the shared task which can be used to improve the task in future. One issue which several participants noted was the way in which the treebank was split into training and evaluation datasets. Out of the 3,409 sentences in the treebank, the first 2,500 sequential sentences were chosen for training. Because this data was not randomized, certain syntactic structures were only found during evaluation and were not present in the training data. Although this may have affected results, all participants evaluated their systems against the same datasets. Based on participant feedback, in addition to reporting *P* and *NP*-measures, it would also be illuminating to include a metric such as Parseval F1-scores to measure partial accuracy. An improved version of the task could also feature a better dataset by expanding the treebank, not only in terms of size but also in terms of linguistic structure. Many commands captured in the annotation game are not yet represented in RCL due to linguistic phenomena such as negation and conditional statements.

Looking forward, a more promising approach to improving the spatial planner could be probabilistic planning, so that semantic parsers could interface with probabilistic facts with confidence measures. This approach is particularly suitable for robotics, where sensors often supply noisy signals about the robot's environment.

Acknowledgements

The author would like to thank the numerous volunteer annotators who helped develop the dataset used for the task using crowdsourcing, by participating in the online game-with-a-purpose.

References

Yoav Artzi and Luke Zettlemoyer. 2011. Bootstrapping Semantic Parsers from Conversations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP (pp. 421–432).

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the Conference of the Association for Computational Linguistics*, ACL (pp. 1415–1425).

Ezra Black, Steven Abney, Dan Flickinger, Claudia Gdaniec, et al. 1991. A Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars. In *Proceedings of the DARPA Speech and Natural Language Workshop* (pp. 306-311). San Mateo, California.

Ann Copestake, et al. 2005. Minimal Recursion Semantics: An Introduction. *Research on Language and Computation*, 3(2) (pp. 281-332).

Bob Coyne, Owen Rambow, et al. 2010. Frame Semantics in Text-to-Scene Generation. *Knowledge-Based and Intelligent Information and Engineering Systems* (pp. 375-384). Springer, Berlin.

Hubert Dreyfus and Stuart Dreyfus. 2009. Why Computers May Never Think Like *People. Readings in the Philosophy of Technology*.

Kais Dukes. 2009. LOGICON: A System for Extracting Semantic Structure using Partial Parsing. In *International Conference on Recent Advances in Natural Language Processing*, RANLP (pp. 18-22). Borovets, Bulgaria.

Kais Dukes. 2013a. Semantic Annotation of Robotic Spatial Commands. In *Proceedings of the Language and Technology Conference*, LTC.

Kais Dukes. 2013b. Train Robots: A Dataset for Natural Language Human-Robot Spatial Interaction through Verbal Commands. In *International Conference on Social Robotics. Embodied Communication of Goals and Intentions Workshop*.

Kais Dukes. 2014. Contextual Semantic Parsing using Crowdsourced Spatial Descriptions. *Computation and Language*, arXiv:1405.0145 [cs.CL]

Myroslava Dzikovska 2004. A Practical Semantic Representation For Natural Language Parsing. *PhD Thesis*. University of Rochester.

Kilian Evang and Johan Bos. 2014. RoBox: CCG with Structured Perceptron for Supervised Semantic Parsing of Robotic Spatial Commands. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval.

Charles Fillmore and Collin Baker. 2001. Frame semantics for Text Understanding. In *Proceedings of WordNet and Other Lexical Resources Workshop*.

Rohit Kate and Ray Mooney. 2006. Using String Kernels for Learning Semantic Parsers. In *Proceedings of the International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics*, COLING-ACL (pp. 913–920).

Rohit Kate and Raymond Mooney. 2010. Joint Entity and Relation Extraction using Card-Pyramid Parsing. In *Proceedings of the Conference on Computational Natural Language Learning*, CoNLL (pp. 203-212).

Rohit Kate, Yuk Wah Wong and Raymond Mooney. 2005. Learning to Transform Natural to Formal Languages. In *Proceedings of the National Conference on Artificial Intelligence* (pp. 1062-1068).

Rohit Kate. 2014. UWM: Applying an Existing Trainable Semantic Parser to Parse Robotic Spatial Commands. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval.

Joohyun Kim and Raymond Mooney. 2012. Unsupervised PCFG Induction for Grounded Language Learning with Highly Ambiguous Supervision. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL (pp. 433-444).

Jayant Krishnamurthy and Tom Mitchell. 2012. Weakly Supervised Training of Semantic Parsers. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL.

Gregory Kuhlmann et al. 2004. Guiding a Reinforcement Learner with Natural Language Advice: Initial Results in RoboCup Soccer. In *Proceedings of the AAAI Workshop on Supervisory Control of Learning and Adaptive Systems*.

Tom Kwiatkowski, Eunsol Choi, Yoav Artzi and Luke Zettlemoyer. 2013. Scaling Semantic Parsers with On-the-fly Ontology Matching. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP.

Peter Ljunglöf. 2014. Shrdlite: Semantic Parsing using a Handmade Grammar. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval.

Willem Mattelaer, Mathias Verbeke and Davide Nitti. 2014. KUL-Eval: A Combinatory Categorial Grammar Approach for Improving Semantic Parsing of Robot Commands using Spatial Context. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval.

Ruslan Mitkov. 1999. Anaphora Resolution: The State of the Art. *Technical Report*. University of Wolverhampton.

Woodley Packard. 2014. UW-MRS: Leveraging a Deep Grammar for Robotic Spatial Commands. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval.

Slav Petrov, et al. 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. *In Proceedings of the International Conference on Computational Linguistics and the Annual Meeting of the Association for Computational Linguistics*, COLING-ACL (pp. 433-440).

Hoifung Poon. 2013. Grounded Unsupervised Semantic Parsing. In *Proceedings of the Conference of the Association for Computational Linguistics*, ACL (pp. 466-477).

Geoffrey Sampson and Anna Babarczy. 2003. A Test of the Leaf-Ancestor Metric for Parse Accuracy. *Natural Language Engineering*, 9.4 (pp. 365-380).

Svetlana Stoyanchev, et al. 2014. AT&T Labs Research: Tag&Parse Approach to Semantic Parsing of Robot Spatial Commands. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval.

Lappoon Tang and Raymond Mooney. 2001. Using Multiple Clause Constructors in Inductive Logic Programming for Semantic Parsing. *Machine Learning*, ECML.

Stefanie Tellax, et al. 2011. Approaching the Symbol Grounding Problem with Probabilistic Graphical Models. *AI Magazine*, 32:4 (pp. 64-76).

Naushad UzZaman and James Allen. 2010. TRIPS and TRIOS System for TempEval-2. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval (pp. 276-283).

Terry Winograd. 1972. Understanding Natural Language. *Cognitive Psychology*, 3:1 (pp. 1-191).

Luke Zettlemoyer and Michael Collins. 2007. Online Learning of Relaxed CCG Grammars for Parsing to Logical Form. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL (pp. 878-887).