Builder, we have done it: Evaluating & Extending Dialogue-AMR NLU Pipeline for Two Collaborative Domains

Claire Bonial[1], Mitchell Abrams[2], David Traum[3], and Clare R. Voss[1]

[1]U.S. Army Research Laboratory, Adelphi, MD 20783
[2]Institute for Human and Machine Cognition, Pensacola, FL 32502
[3]USC Institute for Creative Technologies, Playa Vista, CA 90094
claire.n.bonial.civ@mail.mil

Abstract

We adopt, evaluate, and improve upon a two-step natural language understanding (NLU) pipeline that incrementally tames the variation of unconstrained natural language input and maps to executable robot behaviors. The pipeline first leverages Abstract Meaning Representation (AMR) parsing to capture the propositional content of the utterance, and second converts this into "Dialogue-AMR," which augments standard AMR with information on tense, aspect, and speech acts. Several alternative approaches and training data sets are evaluated for both steps and corresponding components of the pipeline, some of which outperform the original. We extend the Dialogue-AMR annotation schema to cover a different collaborative instruction domain and evaluate on both domains. With very little training data, we achieve promising performance in the new domain, demonstrating the scalability of this approach.

1 Introduction

We adopt, evaluate, and improve upon the two-step NLU pipeline, described in Bonial et al. (2020), which aims to incrementally tame the variation of incoming natural language that the robot must interpret before responding. For each domain in which it operates, the robot must determine whether or not the commands it receives correspond to one of its executable behaviors, such as MOVEMENT (along a front-back axis) and ROTATION. The NLU pipeline leverages AMR to capture the basic content of the input language, and then a conversion system adds behavior time, completion status and speech act information to the original "Standard-AMR," and updates the main action relation of the input AMR to a relation consistently representing an executable robot behavior (see Fig. 1 for a Standard and Dialogue-AMR example comparison). There are two high-level components of the NLU pipeline: a Standard-AMR parser and a graph-to-graph conversion system to convert the Standard-AMR into Dialogue-AMR. Here, we offer the first evaluation of both the Dialogue-AMR annotation schema itself and the components of the pipeline used to automatically obtain the Dialogue-AMR. We test not only in the human-robot, search-and-navigation dialogue domain for which the schema and pipeline was developed, but also in a somewhat similar, yet challenging domain: human-human communication collaboratively building structures in the virtual gaming environment, "Minecraft." In this way, we address the question of what would happen if we wanted our robot to collaborate on a new and different task. We refer to this challenge as "domain extension," instead of "domain adaptation," as we aim to maintain the coverage of our original domain while extending to a new one.

```
(a)                          (b)
(m / move-01                 (c / command-SA
 :ARG0 (y / you)               :ARG0 (c2 / commander)
 :direction (b / back))        :ARG1 (g / go-02 :completable -
                                       :ARG0 r
                                       :direction (b / back)
                                       :time (a / after
                                                  :op1 (n / now)))
                               :ARG2 (r / robot))
```

Figure 1: *Move back* in (a) Standard-AMR (parser output), (b) Dialogue-AMR (conversion system output).

After providing background on AMR and Dialogue-AMR (§2) and detailing our approach (§3), we report on the human-robot evaluation (§4), followed by the Minecraft evaluation (§5), and domain extension of the conversion system and subsequent evaluation (§6). Our contributions include:
i. Retraining existing **Standard-AMR parsers** (3.1) and evaluating on the human-robot (4.1) and Minecraft domains (5.1);
ii. Evaluating and improving a **conversion system** for automatically obtaining Dialogue-AMR (3.2) in both the robot (4.2) and Minecraft (5.2) domains;
iii. Extending the coverage of the **Dialogue-AMR annotation schema** (2.1) to a new domain (6.1) and evaluation after domain extension (6.3).

Proceedings of the 14th International Conference on Computational Semantics, pages 173–183
June 17–18, 2021.

2 Background

To summarize where this work is situated with respect to the past research on this topic—while Bonial et al. (2020) details the Dialogue-AMR annotation schema and proposes the two-step pipeline as one way of automatically obtaining Dialogue-AMR, the technical details of an implementation of the pipeline itself are not described and no evaluation is given. Subsequent research from Abrams et al. (2020) does provide an initial evaluation of a baseline version of the graph-to-graph conversion component of the proposed two-step pipeline; we adopt and evaluate an updated version of this component (described in greater detail in §3.2), however, our evaluation is not directly comparable to the evaluation given in Abrams et al. (2020), since the earlier version of the component was tested on only a limited subset of the annotation categories of Dialogue-AMR. Thus, the current paper constitutes the first evaluation of the proposed two-step pipeline and its components, as well as an evaluation of the extensibility of those components and the Dialogue-AMR schema itself to a new domain.

2.1 AMR & Dialogue-AMR

The two-step NLU pipeline of Bonial et al. (2020) leverages AMR, as it abstracts away from some idiosyncratic surface variation in favor of a more consistent representation for the same concept. This serves the purposes of a dialogue system well: AMR smooths over the nuances of language that may be unimportant for mapping a particular input to one of the robot's behaviors. Nonetheless, "Standard-AMR" does not represent some aspects of meaning that are critical for the human-robot dialogue domain, where the robot must be cued as to what the current dialogue state is, as well as what the current time and completion status of various instructions are. To capture this information, the NLU pipeline uses the "Dialogue-AMR" formalism (Bonial et al., 2020), which adds action time, completion status (i.e., limited tense, aspect) and speech act information to the Standard-AMR. Additionally, to facilitate the final step of mapping to one of the robot's behaviors, Dialogue-AMR further generalizes from the input language, converting a variety of surface realizations (e.g., *turn, rotate, pivot*) of a particular action relation into a single canonical numbered relation (e.g., `turn-01`) to represent one of the robot's behaviors (e.g., ROTATION). Standard-AMR and Dialogue-AMR are contrasted in Figs. 1 and 2.

In Dialogue-AMR, the content of the Standard-AMR is nested in a structure that adds the speech act information as the root predicate (e.g., `command-SA` in Figs. 1, 2). Additionally, the main action from the Standard-AMR (e.g., `move-01`) is converted to one of the action relations (e.g., `go-02`), termed the "robot-concept relation" that maps to an executable robot behavior. Information about the time of that behavior is added (in Fig. 2, the motion event will happen in the future, after the speaking time of the command; thus, it is represented as `:time after-now`).[1] Finally, the behavior completion status, a type of aspect information, is added—whether or not the instructed behavior is telic or contains a clear end point (in Fig. 2, indicated by `completable +`).[2]

Dialogue-AMR draws upon an inventory of 13 speech acts and 26 robot behaviors or "robot-concept relations." Action time and completion status are integrated into Dialogue-AMR by adopting the annotation schema of Donatelli et al. (2018), which categorizes the robot behavior as *past, present,* or *future,* and categorizes 4 aspectual labels: `:stable +/-`, `:ongoing +/-`, `:complete +/-`, and `:habitual +/-`. Dialogue-AMR uses the added category `:completable +/-` to signal whether or not a hypothetical event has an end-goal achievable for the robot.

2.2 Annotated Corpora

We draw from two datasets with Standard-AMR annotations, collected with the aim of developing an interactive agent for collaboration in grounded scenarios. We leverage the DialAMR corpus (Bonial et al., 2020) as training and evaluation data for the NLU pipeline within the human-robot dialogue domain. DialAMR encompasses 1122 instances of The Situated Corpus of Understanding Trans-

[1] In ongoing work to extend the Dialogue-AMR schema, we plan to refine the `:time` annotations to better capture the possibility that an instructed action could already be underway at speaking time, given that we observed that in highly collaborative dialogue, utterances often overlap with actions.

[2] End-point information is needed by a robot to execute a behavior in a low-bandwidth environment where there is a communications lag, precluding real-time voice teleoperation. What constitutes a fully specified behavior is somewhat task and robot-specific; for example, a robot with a static, front-facing camera can assume, as a default, that a picture taken for a user will be from this perspective unless the user specifies otherwise, but a robot with a movable, 360-degree view camera may need to ask the user to provide information on the desired camera angle.

actions (SCOUT), annotated with both Standard-AMR and Dialogue-AMR. SCOUT is comprised of over 80 hours of dialogues from the robot navigation domain (Marge et al., 2016, 2017), collected via a "Wizard-of-Oz" experimental design (Riek, 2012), in which participants directed what they believed to be an autonomous robot to complete search and navigation tasks. The DialAMR corpus was used in the development of the Dialogue-AMR schema, as well as training and testing of the components of the conversion system of Abrams et al. (2020), which we initially adopt, described in §3.2. The data from SCOUT selected for the DialAMR corpus includes a randomly selected, continuous 20-minute experimental trial, which contains 304 utterances (called the *Continuous-Trial* subset). This is the held-out test set that we use throughout our "in-domain" evaluation, as it is representative of an ongoing human-robot interaction.

In addition to in-domain evaluation, we extend evaluation of the Dialogue-AMR schema and NLU pipeline by annotating and testing on the Minecraft Dialogue Corpus (Narayan-Chen et al., 2019). This corpus consists of 509 conversations and game logs, in which two humans communicate via the Minecraft gaming interface chat window while collaboratively building blocks structures. Standard-AMR annotations for the Minecraft corpus (Bonn et al., 2020) were obtained from the developers via a private data-sharing agreement. Our addition of Dialogue-AMR annotations to this corpus is described in §6.1.

3 Approach: Two-Step NLU Pipeline

We adopt and evaluate the two-step NLU pipeline described in Bonial et al. (2020) and Bonial et al. (2019), including both a Standard-AMR parser and a system for converting this into Dialogue-AMR. We describe our selection of an initial Standard-AMR parser and conversion system, both of which we retrain and improve upon, below.

3.1 Standard-AMR Retrained Parser

Standard-AMR provides an initial interpretation of an utterance to be transferred to the Dialogue-AMR. Therefore, an effective Standard-AMR parser is critical for the overall success of the NLU pipeline. We considered several open-source AMR parsers as candidates, and selected two recent releases, the parsers described in Zhang et al. (2019) and Lindemann et al. (2019), which both make use of BERT embeddings (Devlin et al., 2019) and were evaluated on AMR releases, thus providing us with baselines to compare them to each other and to assess our retrained models against their reported performances.

We were able to retrain both of these state-of-the-art AMR parsers on the AMR 2.0 corpus and the recently released AMR 3.0 corpus (a larger corpus including the 2.0 data), and then also retrain them on each of these individual releases of Standard-AMR together with the Standard-AMR subset of the DialAMR corpus of over 800 Standard-AMRs, to adapt them to our human-robot dialogue domain. We evaluated these particular combinations of training data because we wanted to explore whether or not the larger set of data in the AMR 3.0 corpus improved performance on the human-robot dialogue domain, or if it further washed out the distinctions from our smaller in-domain corpus. This yielded a total of eight parsers (see Table 1) for us to evaluate and select from for the purpose of then including in the full NLU parsing pipeline.

3.2 Conversion System

The next step in the NLU pipeline is a graph-to-graph conversion system that uses the input of the utterance text and the Standard-AMR graph to create a Dialogue-AMR graph. We leverage an existing conversion system, "Abrams+", and experiment with improvements to how it classifies the robot-concept relation in our own updated graph-to-graph conversion system, "G2G".

3.2.1 Abrams+ Conversion

We obtained a version of the conversion system described in Abrams et al. (2020), which had been updated by that author in two ways: i. expanded to handle the additional speech acts and robot-concept relation categories of the full Dialogue-AMR schema outlined in Bonial et al. (2020), not all of which were present during the original development, and ii. shifted from a Naïve Bayes to a SVM model for speech act classification. We refer to this system as "Abrams+". This graph-to-graph conversion system implements both rule-based and classifier-based methods in converting a Standard-AMR graph into a Dialogue-AMR graph, and leverages the original utterance and the structure of the Standard-AMR to produce the final Dialogue-AMR, which includes the speech act, tense and aspect information, and a designation of the robot-concept relation. As we use this system as our

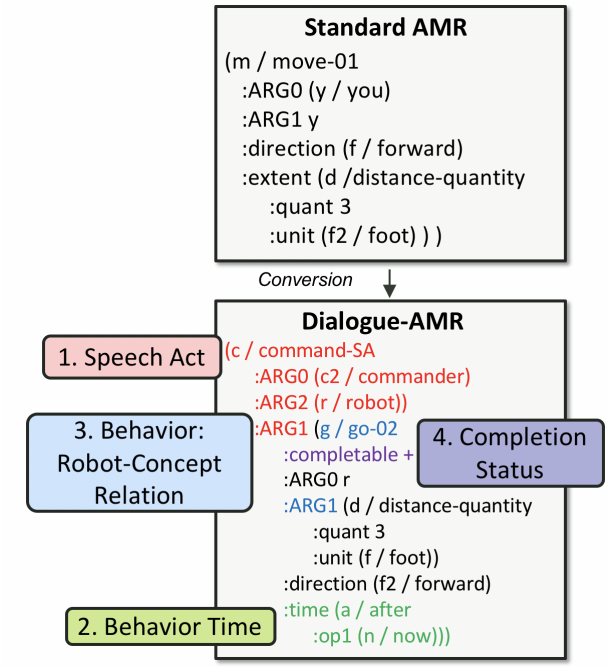

Figure 2: Standard and Dialogue-AMR comparison for Commander instructing robot *Move forward three feet*.

starting point for improvement, we will briefly describe how each of these additions are made in the order just listed, but refer the reader to Abrams et al. (2020) for full details.

Following the numbering of the example in Fig. 2, the first step in the transformation process employs a SVM model with token unigrams features to predict the speech act from the original utterance—critical information for human-robot communication that cannot be gleaned from the Standard-AMR graphs alone.[3] After classification, the speech act label is then stored as a slot to be added to the Dialogue-AMR graph and referenced for decision-making processes downstream. Second, to add behavior time, another classifier—a Naïve Bayes model using token unigrams as features—determines if the event corresponding to the robot behavior pertains to a *past*, *present*, or *future* action. Third, designation of the robot behavior is implemented through a keyword-based approach, which extracts the top root relation (keyword) in the Standard-AMR and checks it against a keyword dictionary of similar actions, and maps it to a single robot-concept relation. Fourth, particular combinations of speech act, tense, and the presence or absence of certain arguments of the robot-concept relation trigger an aspectual label that corresponds to an action's completion status. In the final step of transformation process, the system's rule-based methods use pattern matching techniques to serve multiple functions, including slot filling and slot changing (e.g., transforming mentions of *you* to the fixed role of the `addressee` in Dialogue-AMR).

[3]We acknowledge that the interpretation of speech acts, and indirect speech acts in particular, can be affected by context. Following (Hinkelman and Allen, 1989), we start with only the linguistic signal in the first phase. Since the restricted domain is predictable, it is usually sufficient, but further research aims to leverage situational information and dialogue context where necessary, e.g., to disambiguate an ability question from an indirect instruction.

3.2.2 G2G: Our Updated Conversion System

While we hypothesize speech act, tense, and aspect classification may be fairly robust to language in a new domain, we readily acknowledge that new domains will require the robot to engage in novel behaviors, for example, BUILDING in the Minecraft domain. Thus, although there are many different aspects of the conversion system that we could attempt to improve upon (e.g., classifier types, ordering of components), we saw an opportunity to have the most impact on system performance in multiple domains by focusing on varying the robot-concept relation classification approach. We describe three variants (one keyword-based and two classifier-based) of our updated G2G conversion system below.

G2G *Expanded* Keyword-Based Variant We expanded upon the keyword approach of the Abrams+ system, which was restricted to searching for keyword matches with the top, root relation of the Standard-AMR. We found that this restriction was problematic because the same root relation in the Standard-AMR could correspond to multiple robot-concept relations. *Move* and *go*, generally parsed as `move-01` and `go-02`, are particularly prevalent and could correspond to either front-back MOVEMENT or a ROTATION behavior; both of these were keywords triggering front-back movement in Abrams+, which therefore incorrectly categorized utterances like *Move right 45 degrees* (a ROTATION behavior). In our expansion, the G2G keyword variant searches for matches within all utterance tokens, AMR relations, and arguments. Furthermore, the keyword dictionary was informed by a data-driven analysis in which we created histograms of all utterance tokens and Standard-AMR relations within an instance mapped to a particular robot-concept relation in the manual Dialogue-AMR annotations. In this way, we could see which words and relations occurred with multiple robot-concept relations, like `move-01`, and therefore

remove these from our keyword dictionaries, while adding keywords that are unique to a particular robot-concept relation in the data, such as *degrees*, which consistently cues a ROTATION behavior.

G2G One-Hot and GloVe Variants We also experimented with classifier-based approaches to robot-behavior classification, which we hypothesized may be more efficient to extend to a new domain than a keyword-based approach. The classifiers are Support Vector Machines with different vectorization methods including one-hot encoding and word embeddings from GloVe. Training data for the robot-concept relation classifier comes from examples of each robot-concept category in Bonial et al. (2020), gold-standard labels from the *Continuous-Trial* subset utterances 101-305 (those not used in a held-out test set), and examples pulled from speech act classifier training bins. There are a total of 26 labels for this task, and while many of the movement actions were abundant from these other sources, some of the minority labels (e.g., `equip-01`, `wait-01`, `clarify-10`) required up-sampling to balance training proportions.

4 In-Domain Evaluation

4.1 In-Domain Standard-AMR Parsing

We evaluated the retrained parsers on the SCOUT *Continuous-trial* dataset. We note substantial improvement in Standard-AMR parsing Smatch scores on this set when training with DialAMR in addition to the base training sets (AMR 2.0 and 3.0).[4] Results for the AMR parsing models are presented in Table 1. The noticeably high scores on the parsers retrained on the AMR 3.0 + DialAMR is due in large part to the nature of the speakers' language in the SCOUT corpus and the high levels of similarity in participants' instructions to the robot. This underscores how critical evaluation in another dialogue domain is. We note that, at the segment level as well as can be seen in the Table 1, the Lindemann et al. (2019) parser retrained with DialAMR data evaluated across-the-board to higher scores than the comparably retrained Zhang et al. (2019) parser. Of those two Lindemann et al. (2019) parsers whose Smatch scores did not differ significantly, we selected the one trained with the larger 3.0 dataset with its larger language model as the first component in the full parsing pipeline.

[4]Smatch is an evaluation algorithm for scoring AMR graphs (Cai and Knight, 2013).

Parser	Training	P	R	F
Zhang et al.	AMR 2.0	.47	.77	.58
	2.0 + DialAMR	.73	.77	.75
	AMR 3.0	.52	.80	.63
	3.0 + DialAMR	.88	.89	.89
Lindemann	AMR 2.0	.53	.77	.63
	2.0 + DialAMR	**.92**	.94	**.93**
	AMR 3.0	.55	.81	.65
	3.0 + DialAMR	.91	**.95**	**.93**

Table 1: Retrained AMR parser Smatch results on SCOUT *Continuous-trial* test set.

4.2 In-Domain Conversion to Dialogue-AMR

To pinpoint the performance of the conversion system alone (without error introduced by the automatic Standard-AMR parsing), we report results with gold-standard, manually assigned input Standard-AMR parses. Results are summarized in Evaluation Domain A of Table 2. Focusing initially on the overall Smatch Precision, Recall, and F-scores of the conversion system, our updated system, G2G, leveraging the classifier with one-hot vectorization achieves the highest precision (.85) and F-score (.83) in our domain. All approaches perform comparably overall, especially given that Smatch scores can vary slightly (Opitz et al., 2020) because Smatch is a non-deterministic, greedy hill-climbing algorithm with a preset, default number of random restarts (Cai and Knight, 2013).

Drilling down into the accuracy of the individual component classification tasks, we find accuracy scores of 1.00 for speech acts, .93 for tense, and .93 for aspect across all system variants, as these components are unchanged, and we only alter the robot-concept classification. Again, we note that these accuracy scores are extremely high, given the repetitive nature of the language and prevalence of certain types of commands and feedback assertions. For robot-concept classification, the G2G *expanded* keyword approach (.97 accuracy) does outperform the Abrams+ baseline keyword method (.94 accuracy). Both keyword approaches outperform the G2G classifier-based approaches: one-hot vectorization achieves an accuracy of .90 and GloVe an accuracy of .84. Notably, higher accuracy on the robot-concept classification task does not necessarily translate to higher Smatch F-scores overall. High component accuracy but lower overall F-Score generally indicates that while the system is correctly determining all of the information being added to the Dialogue-AMR, it is not always putting these pieces together correctly. In

Conversion Variant	Evaluation Domain A: SCOUT test data				Evaluation Domain B: Minecraft test data			
	Smatch			Robot Concept	Smatch			Robot Concept
	P	**R**	**F**	**Accuracy**	**P**	**R**	**F**	**Accuracy**
Abrams+	.81	**.82**	.82	.94	.71	.63	.67	.30
G2G-Keyword	.82	**.82**	.82	**.97**	.72	**.64**	**.68**	**.32**
G2G-One-Hot	**.85**	**.82**	**.83**	.90	.73	.62	.67	.20
G2G-GloVe	.84	.81	.82	.84	**.74**	.62	.67	.24
Extended G2G-Keyword	.82	.81	.82	**.94**	.73	**.67**	.70	.41
Extended G2G-One-Hot	**.85**	**.82**	**.83**	.93	**.77**	.65	**.71**	**.54**
Extended G2G-GloVe	.84	.81	.82	.89	.76	.65	.70	.45

Table 2: Summary of Smatch scores & Robot-Concept Relation classification accuracy for each variant conversion system, including our G2G system before and after Minecraft domain extension, tested on SCOUT and Minecraft.

other words, the final step in the conversion system, where slots are captured and changed from the original Standard-AMR structure to the structure of the Dialogue-AMR, is where some of the error reflected in Smatch scores stems from.

5 Minecraft Domain Evaluation

In this section, we report on the Minecraft domain performance of the NLU pipeline with the retrained Standard-AMR parser, the Abrams+ conversion system, and our updated G2G system variants prior to any domain adaptation in order to determine how vital domain extension really is in somewhat similar instruction-giving domains. Given that theoretically speech acts, tense and aspect are somewhat consistent in language regardless of the domain, we hypothesize that these features of our annotation schema and the components of the conversion system capturing them will perform reasonably well on the new Minecraft dialogue domain. However, the main actions or behaviors involved in the collaboration of interlocutors in the original search and navigation domain are quite different from those of building virtual structures from blocks in the new Minecraft domain. We therefore expect that the conversion system will fail to correctly map many of the main action predicates in the Minecraft dialogues to an executable robot behavior. However, we accept this as an interesting question of domain extension for moving our robot to a new task: Is it more efficient to expand a rule-based approach for capturing these new behaviors, or to use a classifier-based approach?

5.1 Minecraft Standard-AMR Parsing

We test the parser selected as the first pipeline component (described in §4.1) on Minecraft data, scoring the parser output on 100 sequential instances of Minecraft dialogue against manually assigned Standard-AMR annotations.[5] The overall Smatch F-score is .57, with a Precision of .63 and Recall of .52. Thus, despite the potential similarity in the two instruction-giving dialogue domains, it is clear that the automatic parsing performance is significantly worse for the Minecraft data than our original domain (where the best Smatch F-score was .93). Error analysis reveals some extremely complicated language phenomena, including dimensions and frequency expressions capturing, for example, the repetition of a placement action: *For the four squares that come out from the middle blocks, add two blue blocks on.* Although this indicates that the parser would benefit from retraining with Minecraft data,[6] in our immediate research we focus on domain extension of the conversion system in order to explore how robust the conversion system might be to noise in the parser input.

5.2 Minecraft Conversion to Dialogue-AMR

This evaluation compares the conversion system output against manually assigned Dialogue-AMRs for the same 100-instance, sequential subset of utterances from the Minecraft corpus used as the test set for the Standard-AMR parser (see §6.1 for Dialogue-AMR annotation details); again, we use gold-standard, manually assigned Standard-AMR parses as input to the conversion system. Results are summarized in Evaluation Domain B of Ta-

[5]The Minecraft AMR corpus includes AMRs for the locations of blocks (expressed as Cartesian coordinates) as each movement takes place; because our focus is natural language dialogue, we removed these instances from our test set.

[6]Bonn et al. (2020) report an F-score of .66 on a Minecraft test set after retraining the Zhang et al. (2019) parser on Minecraft data.

ble 2. Focusing first on overall Smatch scores, our updated system variant leveraging the *expanded* keyword approach performs slightly better (.68 F-score) than both the baseline Abrams+ (.67 F-score) and the classifier-based approaches (.67 F-scores). Although the scores have dropped about 15 points from the original domain, they remain comparable across variants.

When drilling down into the accuracy of the individual components of the conversion system, we find that robot concept classification yields the lowest accuracy scores, with a range of .20-.32. Among the variant approaches to robot-concept classification explored, the *expanded* keyword approach achieves the highest accuracy. The speech act and tense have the same accuracy scores across all versions, .44 and .56, respectively, since these classifiers are stable within the system variants. In this evaluation, aspect varies slightly across approaches as it depends on combinations of speech act and robot-concept relation slot values—its accuracy ranges from .25-.49, with the Abrams+ variant obtaining the highest result. Thus, we see that our hypothesis that speech act, tense, and aspect classification may be fairly robust to a new domain is partially confirmed: robot-concept classification is certainly the most challenging with the lowest accuracy, but the performance of all components is significantly worse than the original domain, suggesting more widespread differences in the language of the two domains.

6 Domain Extension

Here, we describe the small amount of domain extension done to tailor our G2G conversion system to the Minecraft domain, beginning with extensions of the annotation schema itself.

6.1 Extending Dialogue-AMR Schema

One expert Standard-AMR and Dialogue-AMR annotator provided manual Dialogue-AMR annotations to a continuous 100-instance subset of the Minecraft corpus to serve as a test set. This was done by manually augmenting the Standard-AMR release of the Minecraft corpus, maintaining all of the Standard-AMR annotation choices. Additionally, a separate, continuous 200-instance subset of the data was annotated with speech acts and the corresponding robot-concept relations of Dialogue-AMR to serve as training data for the speech act classifier and robot-concept relation classification.[7]

In providing the manual Dialogue-AMR annotation of the Minecraft data, we noted several changes and additions that needed to be made to the annotation schema to account for novel concepts arising in the collaborative building domain, as well as novel dialogue phenomena. First, as expected, we added agent behaviors that would be needed for this domain: BUILDING, represented with the relation `build-01` (e.g., *What are we **building** this time?*), and PLACING, represented with the relation `move-01` (e.g., *Please **place** two red blocks on top of each side...*).

Second, we noted novel dialogue phenomena that we had not observed in the SCOUT data. Speech acts were often nested in this data, such that the content of one speech act was not a typical agent behavior (e.g., a speech act of commanding a ROTATION behavior), but instead another speech act. For example, there were frequent requests for evaluation, often after each building step was completed: *How's this?* and *Is this good?*[8] As a result, we had to shift our annotation schema and conversion system in order to allow for speech act relations to sit where we would normally expect the robot-concept relation.

Finally, we noted frequent use of the verb *need* as an indicator of a less direct command in the Minecraft data: *This will **need** to be placed as far right as you can....* This was interpreted by the interlocutor as a command, i.e., *Place this as far right as you can*. Thus, the *need* relation that roots the Standard-AMR ultimately mapped to the `command-SA` relation of the Dialogue AMR. This phenomenon has significant ramifications for the conversion system, as it was generally assumed, for the SCOUT data, that the utterance and Standard-AMR provides propositional content cuing the robot-concept relation, but we did not expect AMR relations corresponding to the speech act in our

[7] Contact the first author for Minecraft Dialogue-AMR annotations used for train/test.

[8] Following Bunt et al. (2012), Dialogue-AMR speech acts are distinguished between Information Transfer Functions and Action Discussion Functions. Thus, while syntactically questions, cases such as *How's this?* are not annotated using the Dialogue-AMR `Question` speech act, which is reserved for questions that obligate the addressee to introduce new information content into the conversation and demonstrate a commitment to the answer assertion (Traum, 2003). In contrast, these cases obligate the addressee to evaluate the current state of play while simultaneously providing feedback that common conversational ground has been achieved with respect to the desired structure. Indeed, common responses such as *Excellent, Builder* do not fit with a question interpretation.

domain, although plausible (e.g., ***I command*** *you to move forward*).

6.2 Extending Robot-Concept Classification

We added to our *expanded* keyword dictionary to test the effectiveness of a rule-based approach in domain extension. Only two additional concepts were required, `build-01` and `move-01`, but these robot concepts are extremely prevalent in the data. Additionally, in order to test how well a classifier-based approach would capture new behaviors and extend the conversion system to a new domain, we retrained the robot-concept classifier on 166 new manually-annotated training instances of robot concepts from the Minecraft domain. Domain extension also included retraining the speech act classifier on 224 speech acts found in 200 instances of manually annotated Minecraft data.

6.3 Domain-Extended G2G Evaluation

After domain extension, the G2G variant leveraging the one-hot classifier (.71 F-score) very slightly outperforms the keyword (.70 F-score) and GloVe variants (.70 F-score) (again, comparing system output against manually assigned Dialogue-AMRs for the continuous, 100-instance Minecraft test set). Results are summarized in the bottom three rows of Evaluation Domain B of Table 2. The scores remain comparable across all three variants, but we do see improvement overall when comparing against system variants prior to domain extension.

Turning to analysis of the accuracy of individual components of the conversion system, the additional training instances improve speech act classification (from .44 prior to retraining to .57 after) and robot-concept classification for the Minecraft domain. Prior to domain extension, the *expanded* keyword variant achieved the highest accuracy for robot-concept classification (.32), but classifier-based methods with more training data outperform even a domain-extended, data-driven keyword approach, which achieves an accuracy of .41, while one-hot vectorization achieves an accuracy of .54 and GloVe .45. Error analysis reveals that the keyword-based approach struggles to classify robot concepts in this domain, in part, because of language that contains vocatives (e.g. *Excellent, builder*)–which triggers a top `say-01` relation in the Standard-AMR graph–and various uses of *need*, which trigger a `need-01` relation. As noted in the discussion of domain extension of the annotation schema (§6.1), both of these root relations do not cue any domain robot concept, but rather provide information about speech acts and speaker/listener roles, which were consistently implicit in our original domain. Thus, we are currently updating the system to allow for certain relations in the Standard-AMR (e.g., `need-01`) to cue for or map to particular speech acts (e.g., `command-SA`).

This demonstrates a weakness of the keyword-based approach in general: unforeseen linguistic phenomena such as vocatives can strongly affect the accuracy of this approach, while the classifier approach is more robust to these differences since it considers all tokens in the utterance for robot-concept relation prediction, thereby avoiding mis-classification due to this kind of "noise" in the data. When considering our earlier hypothesis that the classifier-based approach to robot-concept classification would be more efficient to extend to a new domain than the keyword-based approach, the results and error analysis here provide modest support for this hypothesis. Both approaches are similarly time-efficient as far as the initial extension efforts are concerned: the keyword approach requires manual observation of the data and subsequent selection and addition of keywords to the dictionaries associated with certain robot-concept relations, while the classifier approach requires some additional manual annotation in the new domain. However, empirically the classifier-based approach slightly outperforms the keyword-based approach in the Minecraft domain, and extending the keyword-based approach requires additional changes in traversal of the graph in order to find the appropriate concept to serve as the keyword for matching, so the effort necessarily goes beyond merely selecting and adding keywords.

Turning back to our original SCOUT test set after Minecraft domain extension (results summarized in the bottom three rows of Evaluation Domain A in Table 2), we find that tailoring the conversion system to Minecraft and expanding the coverage of language that the system can handle has little negative effect on performance in our original domain. We see comparable results for the classifier-based model using one-hot vectorization, maintaining an F-score of .83, which was also the best-performing model for the original domain.

6.4 Full Automatic Pipeline Evaluation

In order to scale up to real-time use, the two-step NLU pipeline will leverage the retrained automatic Standard-AMR parser described in §3.1; however,

up to this point we have reported conversion system results using manually obtained, gold-standard Standard-AMR parses in order to explore the validity of our conversion system approaches without the noise from parsing. Table 3 summarizes the performance of the overall best-performing (across both Smatch scores and component accuracy) *expanded* keyword and one-hot vectorization classifier G2G variants, after domain extension, given Standard-AMR input from the parser. The *expanded* keyword variant is the best-performing model with automatic input, but the scores are close. Although the Smatch F-score has dropped from .71 (with gold-standard input) to .59, we still find this to be very encouraging performance, given the challenges of semantic parsing in a new domain.

Conversion Variant	**SCOUT**			**Minecraft**		
	P	**R**	**F**	**P**	**R**	**F**
Ext. G2G Keyword	.75	.76	.75	**.67**	**.53**	**.59**
Ext. G2G One-Hot	**.83**	**.80**	**.81**	.62	.52	.57

Table 3: Smatch scores for best-performing domain-extended (ext.) G2G variants **using automatically obtained Standard-AMR input** from retrained parser.

7 Related Work

This research is part of a growing body of work in representing various levels of interpretation in existing meaning representation frameworks, and in AMR in particular. We briefly note especially relevant work here. Bastianelli et al. (2014) present their Human Robot Interaction Corpus (HuRIC) following the same Penman Notation (Penman Natural Language Group, 1989) syntax of AMR, but significantly altering AMR to use the sense distinctions and semantic role labels of FrameNet (Fillmore et al., 2012), thereby rendering the use of automatic parsers trained on AMR data challenging. Shen (2018) presents a small corpus (266 instances) of manually annotated AMRs for spoken language to explore the validity of using AMR for spoken language understanding, with promising results but noting that additional data is needed. There is also a neural AMR graph converter for abstractive summarization (producing summary graphs from source graphs) (Liu et al., 2015); however, neural approaches require substantial training data in the form of annotated input and output graphs. The current motivation for the multi-step approach explored here is to handle a low resource problem, as we lack sufficient data to experiment with employing a neural network.

8 Conclusions & Future Work

This paper evaluates and improves upon a two-step NLU pipeline that gradually tames the variation of language so that it can be understood and acted upon by a robot with a limited repertoire of domain concepts and behaviors. After enumerating the extensions needed for the annotation schema itself and contributing a dataset of Dialogue-AMR for the new Minecraft collaborative dialogue domain, we achieve promising results with roughly 200 instances of training data.

We have integrated our updated pipeline into a software stack for a physical robot and are now performing a series of experiments where we use the same dialogue-management system, but vary the NLU component in order to compare task success with the two-step NLU pipeline against a baseline NLU system with a simple syntactic parser. We hypothesize that the NLU pipeline described here, and the deeper semantics of Dialogue-AMR specifically, will be especially advantageous for tracking and grounding user utterances involving coreference (e.g., *Go to **the sign** and send a picture of **it**.*), light verb constructions, which AMR represents identically to parallel synthetic verbs (e.g., *make a left turn; turn left*), negation (e.g., *no, not the door on the right, the left!*), and complex, nested prepositions (e.g., *move through the doorway in front of you on the left*)—all utterances where a simple syntactic parse has been found to lack information needed for interpretation of the intent and grounding. The extrinsic evaluation will also provide an opportunity to explore whether or not the conversion system variant with the best overall Smatch scores corresponds to the best real-world performance, or if we should consider other metrics, such as S^2match (Opitz et al., 2020) and SemBleu (Song and Gildea, 2019). As our results did not demonstrate a clear "best" rule-based, keyword or classifier approach to domain extension, we will continue to experiment with all three variants and consider which is the most time-efficient to extend, either by adding to the keyword dictionary or adding annotations. Overall, we are optimistic that the semantic representation of Dialogue-AMR, which provides a deeper understanding of both what a person said and what they really meant in the conversational context, will enhance human-robot collaboration.

References

Mitchell Abrams, Claire Bonial, and Lucia Donatelli. 2020. Graph-to-graph meaning representation transformations for human-robot dialogue. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 250–253, New York, New York. Association for Computational Linguistics.

Emanuele Bastianelli, Giuseppe Castellucci, Danilo Croce, Luca Iocchi, Roberto Basili, and Daniele Nardi. 2014. HuRIC: a human robot interaction corpus. In *LREC*, pages 4519–4526.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie M. Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. 2020. Dialogue-AMR: Abstract Meaning Representation for dialogue. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 684–695, Marseille, France. European Language Resources Association.

Claire Bonial, Lucia Donatelli, Stephanie M. Lukin, Stephen Tratz, Ron Artstein, David Traum, and Clare Voss. 2019. Augmenting Abstract Meaning Representation for human-robot dialogue. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 199–210, Florence, Italy. Association for Computational Linguistics.

Julia Bonn, Martha Palmer, Zheng Cai, and Kristin Wright-Bettner. 2020. Spatial AMR: Expanded spatial annotation in the context of a grounded Minecraft corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4883–4892, Marseille, France. European Language Resources Association.

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 Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 430–437.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Lucia Donatelli, Michael Regan, William Croft, and Nathan Schneider. 2018. Annotation of tense and aspect semantics for sentential AMR. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 96–108.

Charles J Fillmore, Russell Lee-Goldman, and Russell Rhodes. 2012. The FrameNet Constructicon. *Sign-based construction grammar*, pages 309–372.

Elizabeth Hinkelman and James Allen. 1989. Two constraints on speech act ambiguity.

Matthias Lindemann, Jonas Groschwitz, and Alexander Koller. 2019. Compositional semantic parsing across graphbanks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4576–4585, Florence, Italy. Association for Computational Linguistics.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A Smith. 2015. Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Matthew Marge, Claire Bonial, Brendan Byrne, Taylor Cassidy, A. William Evans, Susan G. Hill, and Clare Voss. 2016. Applying the Wizard-of-Oz technique to multimodal human-robot dialogue. In *RO-MAN 2016: IEEE International Symposium on Robot and Human Interactive Communication*.

Matthew Marge, Claire Bonial, Ashley Foots, Cory Hayes, Cassidy Henry, Kimberly Pollard, Ron Artstein, Clare Voss, and David Traum. 2017. Exploring variation of natural human commands to a robot in a collaborative navigation task. In *Proceedings of the First Workshop on Language Grounding for Robotics*, pages 58–66.

Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. 2019. Collaborative dialogue in Minecraft. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5405–5415, Florence, Italy. Association for Computational Linguistics.

Juri Opitz, Letitia Parcalabescu, and Anette Frank. 2020. AMR similarity metrics from principles. *Transactions of the Association for Computational Linguistics*, 8:522–538.

Penman Natural Language Group. 1989. The Penman user guide. *Technical report, Information Sciences Institute*.

Laurel Riek. 2012. Wizard of Oz Studies in HRI: A Systematic Review and New Reporting Guidelines. *Journal of Human-Robot Interaction*, 1(1).

Hongyuan Shen. 2018. *Semantic Parsing in Spoken Language Understanding using Abstract Meaning Representation*. Ph.D. thesis, Brandeis University.

Linfeng Song and Daniel Gildea. 2019. Sembleu: A robust metric for amr parsing evaluation. *arXiv preprint arXiv:1905.10726*.

David Traum. 2003. Semantics and pragmatics of questions and answers for dialogue agents. In *proceedings of the International Workshop on Computational Semantics*, pages 380–394.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. AMR Parsing as Sequence-to-Graph Transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Florence, Italy. Association for Computational Linguistics.