Proceedings of the Society for Computation in Linguistics

Volume 3 Article 50

2020

Graph-to-Graph Meaning Representation Transformations for Human-Robot Dialogue

Mitchell Abrams
Georgetown University, mja284@georgetown.edu

Claire Bonial
U.S. Army Research Laboratory, claire.n.bonial.civ@mail.mil

Lucia Donatelli
Saarland University, donatelli@coli.uni-saarland.de

Recommended Citation
Abrams, Mitchell; Bonial, Claire; and Donatelli, Lucia (2020) "Graph-to-Graph Meaning Representation Transformations for Human-Robot Dialogue," *Proceedings of the Society for Computation in Linguistics*: Vol. 3 , Article 50.
DOI: https://doi.org/10.7275/4rqf-cz70
Available at: https://scholarworks.umass.edu/scil/vol3/iss1/50

Graph-to-Graph Meaning Representation Transformations for Human-Robot Dialogue

Mitchell Abrams[1], Claire Bonial[2], and Lucia Donatelli[3]

[1]Georgetown University, Washington DC
[2]U.S. Army Research Laboratory, Adelphi, MD
[3]Saarland University, Saarbrücken, Germany
mja284@georgetown.edu

Introduction. This research forms part of a larger project focused on natural language understanding (NLU) in the development of a two-way human-robot dialogue system in the search and navigation domain. We leverage Abstract Meaning Representation (AMR) to capture and structure the semantic content of natural language instructions in a machine-readable, directed, a-cyclic graph (Banarescu et al., 2013). Two key challenges exist for NLU in this task: (i) how to effectively map AMR to a constrained robot action specification within a particular domain; and (ii) how to preserve necessary elements for general understanding of human language with the goal that our robot may expand its capabilities beyond a single domain. To address these challenges, we establish a two-step NLU approach in which automatically-obtained AMR graphs of the input language are converted into in-domain meaning representation graphs augmented with tense, aspect, and speech act information. Here, we detail both rule-based and classifier-based methods to transform AMR graphs into our in-domain graphs, thereby bridging the gap from unconstrained natural language input to a fixed set of robot actions.

Background: Data & Annotations. To determine the type of language found in our task and how it is represented in AMR, we used a corpus of human-robot dialogue in which a person directs a remotely located robot to complete search and navigation tasks (Marge et al., 2016). We then manually selected 504 utterances made up of short, sequential excerpts of the corpus data that are representative of the variety of common exchange types that we see. These sentences were independently double-annotated (IAA 87.8% using the Smatch metric (Cai and Knight, 2013)) and adjudicated following current AMR guidelines.[1]

Notably absent from current AMR representation and essential to our task are two elements: (i) tense and aspect information; and (ii) speech act information regarding speaker intent. To address (i), we adapted the annotation system of Donatelli et al. 2018 for tense and aspect; see Bonial et al. 2019 for details. The absence of speech acts in AMR was anticipated, as existing AMR corpora are text-based.[2] For our task, however, an off-the-shelf taxonomy of speech acts was not ideal. The language found in our domain generally adheres to the division of *information-transfer* and *action-discussion* found in other dialogue act classification systems for conversational agents (e.g., Bunt et al. 2012), yet it also tends to group into specific categories related to our robot's abilities and the search-and-navigation task.

We therefore developed a set of 27 template-like AMRs specific to the task of human-robot dialogue, inspired by classical work on speech acts (Austin, 1975; Searle, 1969). These augmented AMR templates are skeletal AMRs in which the top, anchor node is a fixed relation corresponding to a speech act type (e.g., `assert-02` in the AMR lexicon); one of its numbered arguments, or 'ARGs', is a fixed relation corresponding to an action (e.g., `turn-01`) or the content of the speech act; and arguments of these relations are filled out to detail both dialogue relationships (utterance level) and action specifications (content level) (Bonial et al., 2019). These 27 speech acts are classified into 5 types, listed in Fig. 1 (number of subtypes in parentheses), along with example subtypes for the type `command`. Tense and aspect information are currently annotated only on the content level.

As an example of how our augmented AMRs work, a template for `command:move` is shown in Fig. 2(b); in Fig. 2(c), this template is filled in with the specifics of the utterance *Move to the wall.* Fig. 2(a) shows the original AMR. Note, although

[1] https://github.com/amrisi/amr-guidelines/blob/master/amr.md

[2] https://amr.isi.edu/download.html

Proceedings of the Society for Computation in Linguistics (SCiL) 2020, pages 459-462.
New Orleans, Louisiana, January 2-5, 2020

SPEECH ACT TYPES

```
c / command (6) →   command:move
a / assert (9)      command:turn
r / request (4)     command:send-image
q / question (3)    command:repeat
e / express (5)     command:cancel
                    command:stop
```

Figure 1: Speech act types with example subtypes.

```
(a)  (m / move-01 :mode imperative
        :ARG0 (y / you)
        :ARG1 y
        :ARG2 (w / wall))
(b)  (c / command-02
        :ARG0-commander
        :ARG1-impelled agent
        :ARG2 (g / go-02 :completable +
            :ARG0-goer
            :ARG1-extent
            :ARG3-start point
            :ARG4-end point
            :path
            :direction
            :time (a / after
                :op1 (n / now))))
(c)  (c / command-02
        :ARG0 (c2 / commander)
        :ARG1 (r / robot)
        :ARG2 (g / go-02 :completable +
           :ARG0 r
           :ARG3 (h / here)
           :ARG4 (w / wall)
           :time (a2 / after
                :op1 (n / now))))
```

Figure 2: The utterance *Move to the wall* represented in (a) AMR form, (b) domain specific bare template form, and (c) as a filled-in domain specific graph.

absent in the utterance itself, our template captures key information such as start point and who is addressing whom. It also generalizes across utterances related to movement: whether the instruction uses the word *move, drive* or *proceed*, the in-domain representation is the same. The original AMR captures any lexical differences. The template-like structure further helps identify any critical missing information that may prohibit the robot from successfully completing a given action with required roles and aspectual annotation that specify the existence of an achievable goal (`:completable` ±; see Bonial et al. 2019 for discussion).

To establish a gold standard set of in-domain graphs, two authors manually transformed and adjudicated a subset of 290 single-sentence utterances from the larger human-robot dialogue corpus of 504 AMRs described earlier.

Graph-to-Graph Transformations. We convert AMRs, such as that seen in Fig. 2(a)[3], into our in-domain graphs (e.g., Fig. 2(c)) through a mixed methods approach of both rule-based and classifier-based systems, outlined in Figure 3. Following this transformation pipeline, the system requires both the original AMR and original natural language utterance as input. From the utterance, classifiers first determine the speech act and tense information. The classified speech act then triggers one of the corresponding templates. The speech act subtype is identified by matching the root action predicate in the original AMR to any predicates in a dictionary of keywords associated with each subtype. Aspectual information is triggered by specific patterns of speech act and tense combinations. Next, regex searches the original AMR to extract additional relevant arguments and action predicates that correspond to slots in each template, transforming them when necessary (e.g., *you* to *robot*). Details on each step follow.

While there exists a neural AMR graph converter for a related task (Liu et al., 2015), neural systems require substantial training data in the form of annotated input and output graphs. In contrast, our partially rule-based approach leverages the highly structured AMR information and a relatively small data set of natural language text with speech act or tense labels to train the classifiers. Additionally, our two-step approach, in which we maintain both the original parsed AMR as well as the augmented in-domain AMR, allows us to keep track of both the sentence meaning determined by the linguistic signal alone, and the speaker meaning particular to our context (Bender et al., 2015).

Speech Acts. A speech act classifier predicts one of the five speech act types from the original utterance, triggering the appropriate in-domain template for use. Since natural language is variable, we implement a classifier that will be robust to any language input, rather than rely on a rule-based approach in this step. We implement an off-the-shelf Naive Bayes multinomial classifier for our baseline from the scikit-learn library, using unigrams as features (Pedregosa et al., 2011).[4]

In order to classify the speech act subtype (e.g., `command:move`, `command:turn`), the pipeline uses regex to find the root predicate

[3]We plan to obtain AMRs using automatic parsers including Lindemann et al. 2019.

[4]Though we explored using *unigrams*, *bigrams*, and *unigrams + bigrams*, unigrams performed best as our domain is fairly restricted and predictable from individual words. Higher-order n-grams were not effective due to sparsity issues from a small training set and introduced noise into our system.

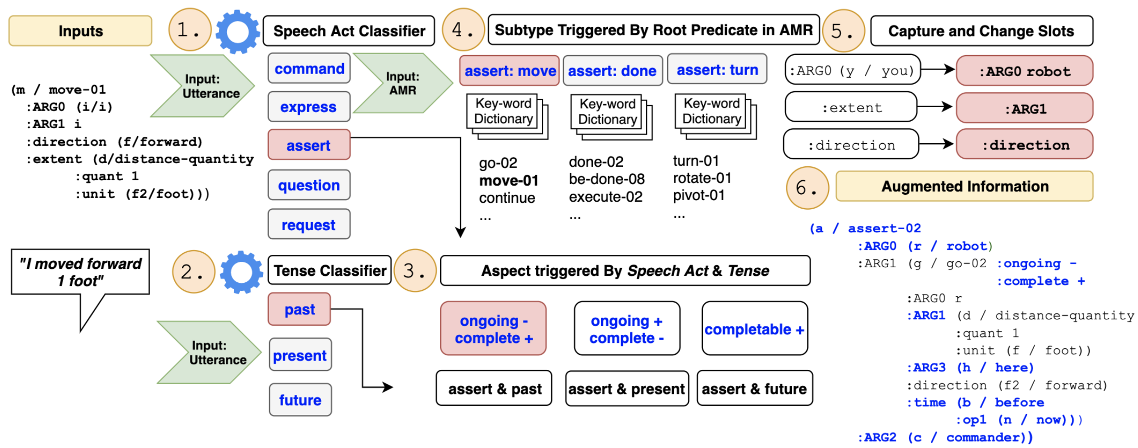

Figure 3: Mixed methods approach to graph transformation. Classifiers and rule-based systems collect and store relevant slot information from the original AMR and original utterance to output an augmented AMR

in the original AMR. In our domain, only a small group of predicates correspond to specific subtypes within each speech act category. For instance, assuming an utterance is classified as a `command`, if `go-02` or `move-01` is found to be the root predicate of the original AMR graph, then the subtype is determined to be `command:move`. While `move-01` is a predicate shared among other subtypes, namely `assert:move` and `request:instruct`, the pipeline only searches for subtypes that belong to the classified speech act category.

Tense Classifier. A tense classifier determines if the original natural language utterance pertains to a *past*, *present*, or a *future* action. While it seems reasonable to use a pre-trained classifier for this three-label classification task, we built our own classifier to handle challenging cases found in our particular domain. For instance, a common shorthand command for taking a picture is *"image"*. Our framework labels the `send-image` action embedded in this command as future, but this word is not inherently associated with the future tense, nor is there any morphological information that would signal this. We implemented the same classifier from the scikit-learn library, using unigrams as features (Pedregosa et al., 2011).[5]

Rule-Based Slot Filling. The rule-based portion of this pipeline relies on regex to find and extract portions of the original AMR to fill the appropriate slots in the in-domain template. For example, for the input utterance *Move to the wall* in Fig. 2, the `command:move` template is triggered with the relation `command-02` anchoring the template and the relation `go-02` capturing the impelled action, `ARG2`, of `command-02` (Fig. 2(b)). In our restricted domain, the `ARG0` *agent* slot of `command-02` is fixed as the *Commander*, (the human instructing the robot) and `ARG1` *entity-commanded* as the *Robot* (Fig. 2(c)). The `ARG0` (mover) and `ARG1` (moved) for `move-01` in the original graph (here, *you*) is converted into the `ARG0` self-directed mover of `go-02`, and this slot is reassigned to *Robot* (Fig. 2(c)). The system then looks for the required *end point* `ARG4` slot in the original AMR, *door* in this case. The precise rules vary depending upon the template triggered, as well as the original verbal predicate used.

Aspect. Finally, we used rule-based methods for capturing aspectual information, as aspectual annotations following Donatelli et al. 2018 revealed consistent patterns associated with speech acts and subtypes. Commands consistently contain the `:completable`$\pm$ annotation indicating if the commanded action is goal-oriented, which is required for execution in our problem domain (a low-bandwidth environment in which lag time in communications is expected, such that all commands require a clear endpoint in advance of execution). From this pattern, we created a rule that if an argument conveying the end point was present in the AMR, then the AMR was given a `:completable +` annotation. For common `move` and `turn` commands, the end point can be realized as the `:extent` slot (e.g., *move forward **five feet***), or the `:destination` slot (e.g., *move to **the wall***). Other speech act types present more

[5]We explored both word and character n-grams; while character n-grams can capture morphological information that signals tense (e.g., *-ed* and *-ing*), data sparsity was an issue. Unigrams, again, proved to be the most effective method.

nuanced patterns and require using speech act and tense information together. For example, assertions contain the `:ongoing - :complete +` aspectual labels within *past* tense.

Results. We evaluated the overall graph-to-graph transformation output against the 290 gold-standard in-domain graphs including all speech act categories, tense and aspect information. The *Smatch* score for this task is **F-score: .78**.[6] This system performs especially well on the `command`, `assertion`, and `express` categories, where the language tends to be predictable within this domain. Sources of errors either stem from speech act misclassification or from the rule-based methods failing to capture language variety. Misclassification of speech act and subtype can lead to more downstream errors since these elements trigger the template. Questions and requests, in particular, prove to be challenging to classify as the language of these categories are more varied. For example, *Can you describe it another way?* could be seen as a polite command, a request, or a question even to human annotators; thus, we are also evaluating the quality of the speech act distinctions. We present the results of the classifier performance using 10-fold cross-validation in Table 1.

Speech Act	Precision	Recall	F-1
Assert	.96	.96	.96
Command	.98	.94	.96
Question	.69	.81	.71
Request	.70	.92	.76
Express	.94	.83	.86
		Accuracy:	.94

Table 1: Speech act classifier performance

Other misclassification results from commands that strayed from expected language. This mainly includes statements of location (e.g., *the cleaning room*), which function as implicit movement commands in our domain. Finally, the system failed to capture certain root action predicates in the original AMRs as they were overlooked and not included in our rule-based methods—a dictionary that signals speech act subtypes.

Conclusions & **Future Work.** This paper introduces a novel yet simple approach to AMR graph-to-graph transformation, in which parser-output AMRs are converted to augmented AMRs specific to human-robot dialogue and search-and-navigation tasks. Preliminary results are quite promising, reflected by high F-1 and *Smatch* scores. However, we have yet to see this translate into performance in the end-to-end system we are working to implement. Future work will address handling truly ambiguous speech acts that cannot be determined from the language alone, which we hope to resolve by leveraging dialogue context and computer vision.

[6]Our f-score is high when compared to another AMR graph transformation task (Liu et al., 2015), but, to our knowledge, there is no directly comparable task.

References

J. L. Austin. 1975. *How to Do Things with Words*, 2nd edition. Harvard University Press and Oxford University Press.

L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proc. LAW*, pages 178–186.

E. M. Bender, D. Flickinger, S. Oepen, W. Packard, and A. Copestake. 2015. Layers of interpretation: On grammar and compositionality. In *Proc. of ICCS*, pages 239–249.

C. Bonial, L. Donatelli, S. M. Lukin, S. Tratz, R. Artstein, D. Traum, and C. Voss. 2019. Augmenting abstract meaning representation for human-robot dialogue. In *Proc. of DMR*, pages 199–210.

H. Bunt, J. Alexandersson, J. Choe, A. C. Fang, K. Hasida, V. Petukhova, A. Popescu-Belis, and D. Traum. 2012. ISO 24617-2: A semantically-based standard for dialogue annotation. In *Proc. of LREC*, pages 430–437.

S. Cai and K. Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proc. of ACL*, volume 2, pages 748–752.

L. Donatelli, M. Regan, W. Croft, and N. Schneider. 2018. Annotation of tense and aspect semantics for sentential AMR. In *Proc. of LAW-MWE-CxG*, pages 96–108.

M. Lindemann, J. Groschwitz, and A. Koller. 2019. Compositional semantic parsing across graphbanks. In *Proc. of ACL*, pages 4576–4585.

F. Liu, J. Flanigan, S. Thomson, N. Sadeh, and N. A. Smith. 2015. Toward abstractive summarization using semantic representations. In *Proc. of NAACL-HLT*.

M. Marge, C. Bonial, B. Byrne, T. Cassidy, W. Evans, S. Hill, and C. Voss. 2016. Applying the wizard-of-oz technique to multimodal human-robot dialogue. In *Proc. of RO-MAN*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, and Grisel O. 2011. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830.

J. R. Searle. 1969. *Speech acts: An essay in the philosophy of language*. Cambridge University Press.