Tell Your Robot What To Do: Evaluation of Natural Language Models for Robot Command Processing[*]

Erick Romero Kramer, Argentina Ortega Sáinz,
Alex Mitrevski, and Paul G. Plöger

Hochschule Bonn-Rhein-Sieg, Sank[20],t Augustin, Germany
`erick.romero@smail.inf.h-brs.de`, {`argentina.ortega`,
`aleksandar.mitrevski`, `paul.ploeger`}`@h-brs.de`

Abstract. The use of natural language to indicate robot tasks is a convenient way to command robots. As a result, several models and approaches capable of understanding robot commands have been developed, which, however, complicates the choice of a suitable model for a given scenario. In this work, we present a comparative analysis and benchmarking of four natural language understanding models - Mbot, Rasa, LU4R, and ECG. We particularly evaluate the performance of the models to understand domestic service robot commands by recognizing the actions and any complementary information in them in three use cases: the RoboCup@Home General Purpose Service Robot (GPSR) category 1 contest, GPSR category 2, and hospital logistics in the context of the ROPOD project.

Keywords: Natural Language Understanding · Robot Commands · Comparative Analysis · Benchmarking.

1 Introduction

In the context of service and particularly domestic robots, using natural language to give robot commands is a convenient way of interacting with a robot since it requires no specialized knowledge on the part of the human operator. A natural language command is composed of at least one action and a set of arguments that provide additional context to the indicated action. Different ways to replicate this form of communication in order to improve human-robot interaction (HRI) and use it in the context of domestic service robots have been explored in the literature.

Developing a system capable of understanding natural language commands is not trivial. One of the major challenges is dealing with the abstractions present in the way people speak, for instance ignoring grammar rules, changing the order of words, and so forth. Because of the current rapid growth of the field of natural

[*] We gratefully acknowledge the support by the b-it International Center for Information Technology.

language understanding (NLU), an up-to-date comparative analysis of the state of the art models is missing. This lack of comparative analysis makes it difficult to determine which model would perform well when creating a system in which natural language commands should be used.

With this work, we provide a survey of the current state of the art of natural language models for robot command processing and analyze freely available options that can be used to develop such systems. In particular, we perform a comparative analysis of a selected set of available NLU frameworks - Mbot [20], Rasa [7], LU4R [4], and ECG [14] - and evaluate their effectiveness on three use cases: the RoboCup@Home General Purpose Service Robot (GPSR) category 1 contest [21], GPSR category 2, and hospital logistics in the context of the ROPOD project[1]. The models are evaluated using standard metrics, such as precision, recall, F-measure, and accuracy on predefined sets of natural language commands. The major objective of this work is to serve as a guideline for selecting a proper model to understand robot commands in a given context. A repository[2] has been set up containing the datasets and supplemental material used for the development of this paper.

2 State of the art

Fig. 1 presents an overview of different NLU approaches and their applications to robotics.

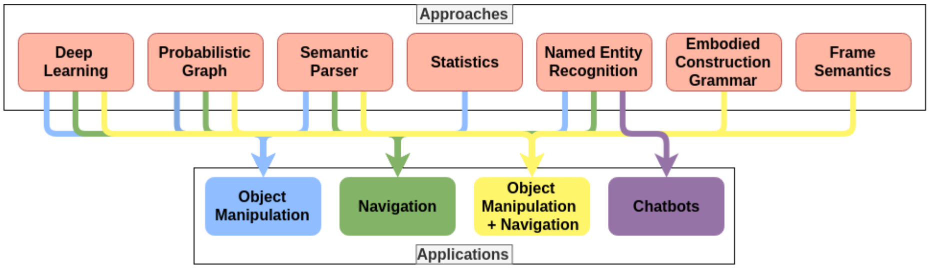


Fig. 1: Overview of different approaches that have been used for understanding natural language-based robot commands.

A recent survey in the field of NLP is presented by Wiriyathammabhum et al. [33], where the integration of computer vision techniques and NLP models for multimedia and robotics applications is introduced. Liu and Zhang [19] explore various methodologies implemented for HRI using natural language. Otter et al. [26] review deep learning methods applied to NLP.

[1] ROPOD is a Horizon 2020 project:
`http://cordis.europa.eu/project/rcn/206247_en.html`

[2] `https://github.com/ErickKramer/NLU_Benchmarking`

A "*functional benchmark*", named Functional Benchmark on Speech Understanding (FBM3), was presented in [32]. The goal there was to measure and evaluate the performance of different architectures in the context of service robots operating in a home environment and in service robot competitions. The analysis done in this benchmark focuses more on comparing speech recognition rather than NLU models.

Several models that apply deep learning for understand robot commands have been used in the context of object manipulation [1,5,31,35], navigation [6,30,23], and both object manipulation and navigation [3,20]. A different approach is the application of probabilistic graphs to understand robot commands; such models have been used in the context of object manipulation [9,27], navigation [11], and both tasks combined [2,17,18,25].

In the field of semantic parsers, we can find the work done in [8,13,22,29]. A statistical model using a conditional random field for grounding natural language instructions is presented in [24]. A different approach is the implementation of named-entiry recognition (NER) to capture the entities inside the sentences, as shown in [7,10]. In [14], construction grammars are used to understand robot commands, while the work done by [4] and [15] relies on frame semantics to do so.

One limitation in the existing literature is the missing implementation of state of the art word embedding representations, such as BERT [12] or ELMo [28]. Additionally, there is no comparative analysis and benchmark of open source models capable of understanding robot commands. In particular, the surveys presented here do not perform a quantitative comparison of the existing NLU models. Similarly, most comparative studies only focus on models that use the same approach (e.g. deep learning), but do not compare models using different approaches, such as for instance grammar-based models and deep learning models, on the same benchmark.

3 Qualitative Comparative analysis

Our analysis in this paper focuses on the following models:

- *Mbot* [20]: A model that follows a deep learning approach to understand robot commands. In particular, Mbot uses a recurrent neural network (RNN) with 500 long short-term memory (LSTM) cells to perform action detection and an RNN with two layers of 500 LSTM cells to perform slot filling. The action detection network identifies the corresponding action behind the commands, while the slot filling network assigns labels to all the words in the command following the IOB-format[3] and identifies slots such as *object, destination, source, sentence,* and *person.*
- *Rasa* [7]: A Python library that contains two modules: Rasa Core, which is used as a dialogue manager, and Rasa NLU, which can perform NER and classify intentions. We used Rasa NLU as a second model, using a conditional

[3] B- Beginning word of the entity, I- Inside word of the entity, O- Out word

random field (CRF) to perform NER, while an embedding classifier[4] is used to identify the actions in a command.
- *LU4R* [4]: A model that relies on the theory of frame semantics [16]. Its language understanding part contains four stages: (i) obtaining the morphological and syntactic information of the command, (ii) ranking the transcriptions obtained from a speech recognizer module, (iii) identifying the semantic frame that corresponds to the action in the command, and (iv) adding the proper labels as well as the corresponding category to the arguments in the command. The detection of the actions and arguments is performed using an support vector machine (SVM) with a Markovian formulation combined with the morphological and syntactic features found.
- *ECG* [14]: A model that uses the basic principles of construction grammars to construct a semantic formalism - the so-called denominated schemas - for expressing the meaning of commands. The language understanding part contains two main modules: (i) an *analyzer* that parses the sentences into a semantic specification schema using defined ontologies and grammars, and (ii) a *specializer* that extracts the information contained in the semantic specification using a set of predefined templates and generates an action specification schema.

These models were chosen based on the fact that they are freely available and, with the exception of LU4R, are open source; all models, with the exception of ECG, are also able to work offline and are capable of understanding robot commands.

Table 1 presents an overview of the different features each of the models have. Clarifying some of the terminology used in the table, "monologue" refers to sentences that contain more than one action. The "ready to use" feature expresses the fact that the authors have provided pre-trained models capable of dealing with a set of natural language commands. The linguistic knowledge required to train or adapt the models to a new domain is also expressed as a feature, such that a "high" level means that in order to use the full potential of the model, significant knowledge of linguistics is required, while a "moderate" level indicates that it is possible to implement the model without much linguistic knowledge. This feature is not applicable to LU4R since it cannot be adapted to new domains, due to the fact that it is not open source.

The features presented here indicate that Rasa NLU and Mbot are both quite attractive options. Both present customizable action and argument labels, which means that that the models can be used to identify most actions and entities. In addition, it is possible to change the interpretation format and use it to build a planner to execute actions. These models are also capable of ignoring unnecessary information in commands, such as intros and capturing entities composed of more than one word, similar to LU4R. For the case of Mbot, the model can be used out of the box to understand a large variety of actions, while for Rasa NLU, it is quite appealing that it can be used in any language. Regarding LU4R, it would be better if it was possible to retrain the model and adapt it to new

[4] Based on the StarSpace model [34].

domains, although the available model is already able to understand 18 semantic frames. Additionally, similar to Mbot, LU4R can deal with sentences containing multiple actions in them. The major drawback of ECG is the required linguistic knowledge to properly use the model; in addition, the model cannot understand entities composed of more than one word, such as "living room".

Models Features	Mbot	Rasa NLU	LU4R	ECG
Customizable action labels	☑	☑	☒	☑
Customizable action arguments labels	☑	☑	☒	☑
Customizable output format	☑	☑	☒	☒
Supports sentences with intros	☑	☑	☑	☑
Supports multiple word entities	☑	☑	☑	☒
Supports monologues	☑	☒	☑	☒
Ready to use	☑	☒	☑	☒
Language	English	Any[5]	English, Italian	English
Programming language	Python	Python	Java	Java, Python
Used for robots	☑	☒	☑	☑
Linguistic knowledge required to adapt	Moderate	Moderate	NA	High

Table 1: Model comparison. ☑: a model has a given feature, ☒: it does not.

4 Model Benchmarking

In this section, we set up a quantitative comparative analysis of the above models on three use cases. Our first and second use cases are the General Purpose Service Robot (GPSR) contest category 1 and category 2, which are part of the Robocup@Home competition. GPSR categories 1 and 2 concern tasks with low to moderate degrees of difficulty. The set of actions that were required for these use cases are *answer, find, follow, guide, take, tell, go, and meet.*

The third use case is the ROPOD project, where multiple logistics robots are deployed to a hospital for the purpose of transporting items, such as carts and beds, between different places in the hospital. The idea here is that the robotic platforms can be commanded around the hospital without the need of a GUI. For this use case, we defined the set of actions *attach, find, follow, guide, push, detach, and go.*

4.1 Datasets

We used three datasets for evaluating the selected NLU models - One for GPSR category 1, one for GPSR category 2, and one for ROPOD. The datasets for

[5] As it was claimed in the documentation of the library `http://rasa.com/docs/rasa/nlu/language-support/`

category 1 and 2 were created with the help of the GPSR command generator tool[6]. We started by generating a random set of 10,000 sentences for each category and preprocessed the sentences by (i) removing those that were not commands and (ii) converting them to lowercase letters. For each category, we chose a total number of 110 random sentences, ensuring that all actions involved are equally covered. We organized the sentences of each dataset in two inputs files, one containing single action sentences and one containing multiple action sentences. In those files, the sentences were organized in groups based on the action behind the command for the single action sentences and based on the number of actions for the multiple actions sentences. The dataset for ROPOD was built by manually creating a total number of 97 sentences. We included commands that we believe were suitable in the context of a hospital environment. The sentences in this dataset were split in a similar manner to the previous two datasets. For each single and multiple actions file, we manually developed an output file containing all the expected interpretations of the sentences following the interpretation format presented by Mbot [20]. We found this format to be quite useful as it displays the intention behind the sentences as well as the complementary arguments in a clear and concise way. All our experiments were performed on an Asus ROG GL552V with an Intel core i7 and 12GB of RAM.

Example sentences from the three datasets are shown in Table 2.

Dataset	**Input**	**Output**
GPSR Cat 1 (Single)	locate the pringles in the dining room	find object pringles destination dining room
	give to tracy at the kitchen the soap from the towel rail	take person tracy destination kitchen object soap source towel rail
GPSR Cat 1 (Multiple)	grasp the noodles from the towel rail and place it on the bookshelf	take object noodles source towel rail take object it destination bookshelf
	navigate to the bathroom, locate someone, and tell the time	go destination bathroom find person someone tell sentence the time
GPSR Cat 2 (Single)	bring me the peach from the bookshelf	take person me object peach source bookshelf
	guide morgan to the coffee table, you may find him at the shower	guide person morgan destination coffee table source shower
GPSR Cat 2 (Multiple)	get the pear from the center table and put it on the fireplace	take object pear source center table take object it destination fireplace
	go to the cabinet, look for the banana, and deliver it to taylor at the tv coach	go destination cabinet find object banana take object it person taylor destination tv coach
ROPOD (Single)	guide the nurse to the corridor	guide person nurse destination corridor
	undock from the station b	detach object station b
ROPOD (Multiple)	follow the green robot and attach to the station f	follow object green robot attach object station f
	go to the entrance, find the nurse and guide her to the room 10	go destination entrance find person nurse guide person her destination room 10

Table 2: Examples of sentences from the datasets in the three different use case. The colors here indicate either an intention or a slot.

[6] https://github.com/kyordhel/GPSRCmdGen

4.2 Evaluation metrics

Similar to the benchmark presented in [32], we evaluated the performance of the models in terms of the following metrics:

- *Action Classification (AC)*: Measures the ability of the models to perform correct detection of the actions in the sentences. AC will be measured through the *precision* (Eq. 1), *recall*, (Eq. 2), and the *F1 score* or *F-measure* (Eq. 3) [10,32][7].

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3}$$

- *Full Command Recognition (FCR)*: Measures the ability of the models to understands the commands completely, namely their ability to recognize the correct actions and complementary information. FCR will measured through the *accuracy* (Eq. 4) [20].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

- *Runtime*: Measures the time required for the models to process the complete dataset. The values are going to be expressed in seconds.

4.3 Training and adaptation

The three open source models - Mbot, Rasa, and ECG - required training and adaptation for our use cases. Both Mbot and Rasa were trained using labeled datasets[8] containing *200,000* commands for GPSR category 1 and 2, and *199,997* commands for ROPOD. Due to the nature of the ECG framework, it was not necessary to train it, but to add new vocabulary to the grammar file and the ontology. In order to do so, we took advantage of the ECG workbench[9] tool.

5 Results

The results obtained from the experiments with single action sentences are reported in Table 3 and plotted in Fig. 2a. The results shown in Table 4 and plotted in Fig. 2b corresponds to the experiments with multiple action sentences.

[7] TP = True Positive, TN = True Negative, FP = False Positive, and FN = False Negative

[8] Generated using a modified version of a data generator script provided by the authors of Mbot in `https://github.com/socrob/mbot_natural_language_processing`.

[9] `https://github.com/icsi-berkeley/ecg_workbench_release`.

Single action sentences															
	Action classification									Full command recognition					
Metrics	Precision			Recall			F1			Accuracy			Run-time		
Datasets Models	Cat1	Cat2	ROPOD	Cat1	Cat2	ROPOD	Cat1	Cat2	ROPOD	Cat1	Cat2	ROPOD	Cat1	Cat2	ROPOD
Mbot	**1.0**	**0.97**	0.92	**1.0**	**1.0**	**1.0**	**1.0**	**0.98**	0.96	**1.0**	**0.75**	0.77	3.85	3.36	4.23
LU4R	0.71	0.57	0.74	0.61	0.57	0.53	0.66	0.58	0.62	0.41	0.29	0.35	4.95	2.89	1.82
Rasa NLU	**1.0**	0.94	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**0.98**	**1.0**	0.90	0.66	**0.89**	**0.14**	**0.17**	**0.11**
ECG	**1.0**	0.83	0.95	0.14	0.06	0.29	0.24	0.12	0.44	0.08	0.05	0.21	NaN	NaN	NaN

Table 3: Results of the experiments with single action sentences. The blue-colored values represent the best values obtained for each metric on each dataset.

Multiple action sentences															
	Action classification									Full command recognition					
Metrics	Precision			Recall			F1			Accuracy			Run-time		
Datasets Models	Cat1	Cat2	ROPOD	Cat1	Cat2	ROPOD	Cat1	Cat2	ROPOD	Cat1	Cat2	ROPOD	Cat1	Cat2	ROPOD
Mbot	**1.0**	**1.0**	0.97	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.98	**1.0**	**0.83**	0.73	2.10	4.02	3.56
LU4R	**1.0**	0.83	0.89	0.67	0.48	0.57	0.80	0.60	0.68	0.17	0.07	0.15	1.62	1.34	0.91
Rasa NLU	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.58	0.80	**0.81**	**0.06**	**0.14**	**0.09**
ECG	**1.0**	**1.0**	0.96	0.25	0.23	0.40	0.40	0.36	0.56	0.04	0	0.15	NaN	NaN	NaN

Table 4: Results of the experiments with multiple action sentences. The blue-colored values represents the best values obtained for each metric on each dataset.

Based on these results, we can notice that both Mbot and Rasa NLU obtained better precision than LU4R and ECG across the three use cases. LU4R obtained the worst precision for all the datasets because it misclassified commands involving actions such as *tell, guide, and detach*. ECG had a decent performance in terms of precision. It is also worth mentioning that Mbot practically guessed the actions behind the commands with words that are not in its dictionary.

In terms of recall, Mbot and Rasa obtained a full score for all the datasets, which means that they were able to provide an interpretation for all the commands. The low recall obtained by ECG shows that from all the commands, very few were actually classified. The results obtained by LU4R shows that it struggled to provide an interpretation for commands that contained actions not covered by the semantics frames on which it was already trained.

The F1 score gives us more insight into the actual ability of the models to classify the actions behind the given commands. We can see that for GPSR categories 1 and 2, both Mbot and Rasa NLU had an almost perfect score, which means that they were always able to both interpret the actions when receiving a command and correctly classify the interpreted actions. In the case of ROPOD, Rasa NLU outperformed Mbot, as Mbot misclassified some of the actions. This shows that the use of pretrained word embeddings by Mbot limits it to only understand words that are already present in its dictionary; since Rasa NLU created its own word embeddings during training, it was able to understand uncommon words such as *uncharge* or *undock*. ECG obtained the worst results on all the datasets, showing that it is likely not a good model to classify actions.

The results obtained by LU4R were better than expected taking into account that it was not possible to retrain the model to our domains.

In terms of FCR, Mbot outperformed the other models for GPSR categories 1 and 2, while Rasa NLU obtained the best results for ROPOD. Rasa NLU had troubles differentiating between source and destination entities, which cause the model to perform suboptimally on the multiple sentence dataset for GPSR category 1. The results obtained for ECG show that the model failed to understand complete commands. The major reason for this is the inability of the model to understand two words tokens, which appear quite frequently in the datasets. LU4R failed mostly because it does not support some of the commands that were present in the datasets; in addition, the model struggled with complementary information involving people and locations.

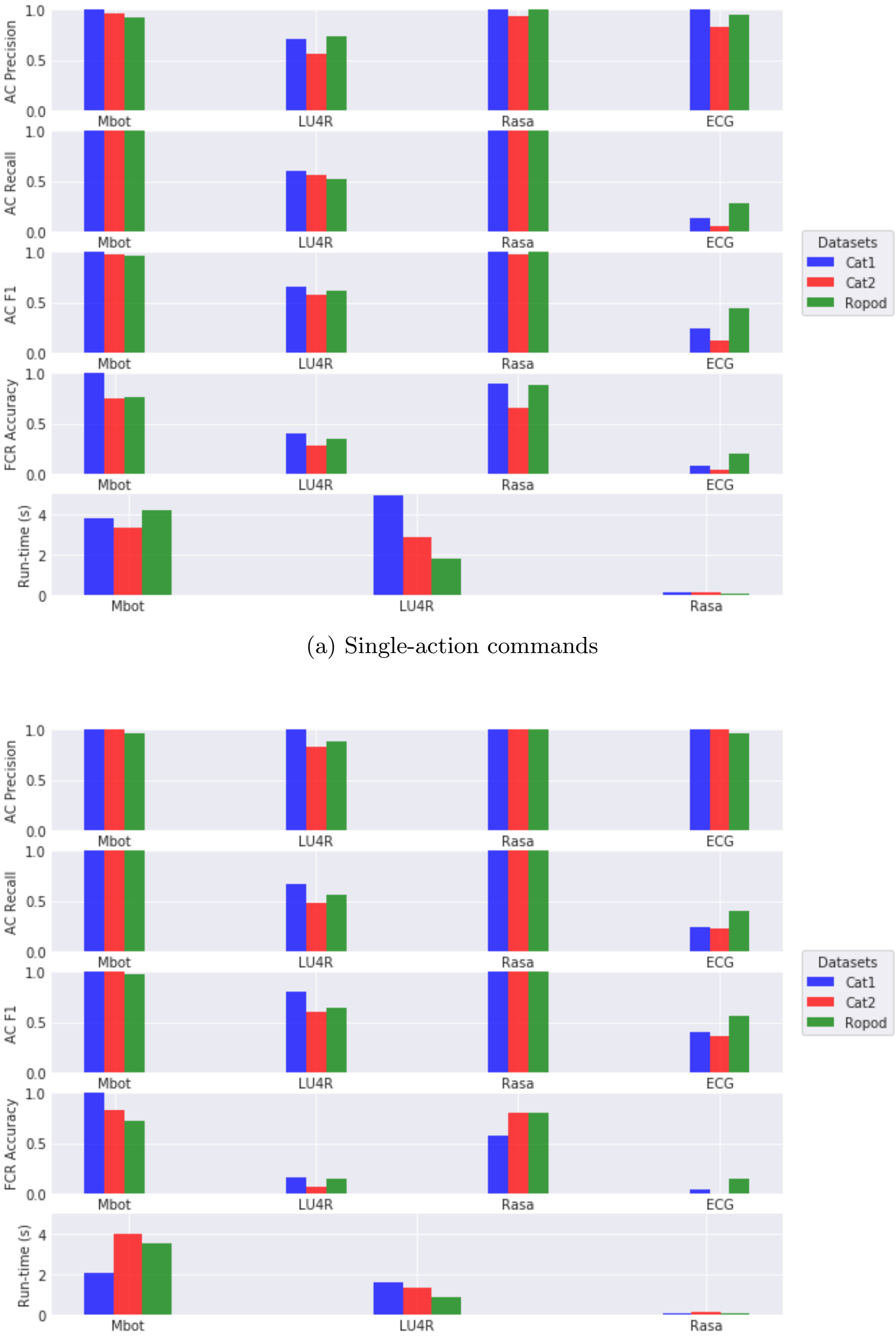
The results in terms of runtime show that Rasa NLU was the fastest model across all the datasets. This could be due to the shallow structure of the CRF that performs NER and the embedding classifier that identifies the intentions. Both Mbot and LU4R were considerably slower than Rasa NLU, with LU4R having a shorter runtime than Mbot for GPSR category 2 and ROPOD, but only because LU4R could not understand some of the commands given in them, which means that no interpretation time was spent on those. We could not measure the runtime of ECG because of the delay in the communication process with the ECG analyzer; in other words, it was necessary to manually send the commands in the datasets one by one in order to verify that the generated interpretation coincided with the command sent.

6 Conclusion

This work presented a survey of the existing natural language understanding models for interpreting robot commands. A comparative analysis of a selected set of freely available models - Mbot, Rasa, LU4R, and ECG - was also performed. These models were benchmarked on three use cases: GPSR category 1, GPSR category 2, and ROPOD. Based on the obtained results, we can conclude that both Mbot and Rasa are suitable for robot command understanding; however, Mbot is slightly more suitable since Rasa has troubles differentiating certain location entities between *destination* and *source* categories.

To improve the results of the existing models and particularly Mbot and Rasa, state of the art word embedding representations could be used. Implementing an approach that takes into account the grounded information obtained from a semantic map to resolve ambiguous interpretations - similar to LU4R - could also be explored. For properly spliting multiple sentences into phrases, Google Syntaxnet[10] could be used, where each phrase would contain an action that needs to be executed. Finally, a pronoun resolution approach to properly identify implicit information in commands would be useful to develop.

[10] https://opensource.google.com/projects/syntaxnet



(a) Single-action commands

(b) Multi-action commands

Fig. 2: Results of the NLU models on our three use cases.

References

1. Ahn, H., Choi, S., Kim, N., Cha, G., Oh, S.: Interactive Text2Pickup Network for Natural Language based Human-Robot Collaboration. CoRR **abs/1805.10799** (2018), `https://arxiv.org/abs/1805.10799`
2. Arkin, J., Walter, M.R., Boteanu, A., Napoli, M.E., Biggie, H., Kress-Gazit, H., Howard, T.M.: Contextual awareness: Understanding monologic natural language instructions for autonomous robots. In: Robot and Human Interactive Communication (RO-MAN), 2017 26th IEEE Int. Symp. pp. 502–509 (2017)
3. Arumugam, D., Karamcheti, S., Gopalan, N., Williams, E.C., Rhee, M., Wong, L.L., Tellex, S.: Grounding Natural Language Instructions to Semantic Goal Representations for Abstraction and Generalization (2017)
4. Bastianelli, E., Croce, D., Vanzo, A., Basili, R., Nardi, D.: A Discriminative Approach to Grounded Spoken Language Understanding in Interactive Robotics. In: IJCAI. pp. 2747–2753 (2016)
5. Bisk, Y., Shih, K.J., Choi, Y., Marcu, D.: Learning Interpretable Spatial Operations in a Rich 3D Blocks World. CoRR **abs/1712.03463** (2017), `https://arxiv.org/abs/1712.03463`
6. Blukis, V., Brukhim, N., Bennett, A., Knepper, R.A., Artzi, Y.: Following High-level Navigation Instructions on a Simulated Quadcopter with Imitation Learning. CoRR **abs/1806.00047** (2018), `https://arxiv.org/abs/1806.00047`
7. Bocklisch, T., Faulker, J., Pawlowski, N., Nichol, A.: Rasa: Open source language understanding and dialogue management. CoRR **abs/1712.05181** (2017), `https://arxiv.org/abs/1712.05181`
8. Boldt, B., Gavran, I., Darulova, E., Majumdar, R.: Precise but Natural Specification for Robot Tasks. CoRR **abs/1803.02238** (2018), `https://arxiv.org/abs/1803.02238`
9. Broad, A., Arkin, J., Ratliff, N., Howard, T., Argall, B.: Real-time natural language corrections for assistive robotic manipulators. The International Journal of Robotics Research **36**(5-7), 684–698 (2017)
10. Chesworth, D., Harmon, N., Tanner, L., Guerlain, S., Balazs, M.: Named-entity recognition and data visualization techniques to communicate mission command to autonomous systems. In: Systems and Information Eng. Design Symp. (SIEDS), 2016 IEEE. pp. 233–238 (2016)
11. Chung, I., Propp, O., Walter, M.R., Howard, T.M.: On the performance of hierarchical distributed correspondence graphs for efficient symbol grounding of robot instructions. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ Int. Conf. pp. 5247–5252 (2015)
12. Devlin, J., Chang, M., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018), `https://arxiv.org/abs/1810.04805`
13. Dukes, K.: Semeval-2014 task 6: Supervised semantic parsing of robotic spatial commands. In: Proc. 8th Int. Workshop on Semantic Evaluation (SemEval 2014). pp. 45–53 (2014)
14. Eppe, M., Trott, S., Raghuram, V., Feldman, J.A., Janin, A.: Application-Independent and Integration-Friendly Natural Language Understanding. In: GCAI. pp. 340–352 (2016)
15. Evangelista, D., Villa, W.U., Imperoli, M., Vanzo, A., Iocchi, L., Nardi, D., Pretto, A.: Grounding natural language instructions in industrial robotics (2017)

16. Fillmore, C.J.: Frames and the semantics of understanding. Quaderni di semantica **6**(2), 222–254 (1985)
17. Howard, T.M., Tellex, S., Roy, N.: A natural language planner interface for mobile manipulators. In: Robotics and Automation (ICRA), 2014 IEEE Int. Conf. pp. 6652–6659 (2014)
18. Kollar, T., et al.: Generalized Grounding Graphs: A Probabilistic Framework for Understanding Grounded Commands. CoRR **abs/1712.01097** (2017), `https://arxiv.org/abs/1712.01097`
19. Liu, R., Zhang, X.: A Review of Methodologies for Natural-Language-Facilitated Human-Robot Cooperation. CoRR **abs/1701.08756** (2017), `https://arxiv.org/abs/1701.08756`
20. Martins, P.H., Custódio, L., Ventura, R.: A deep learning approach for understanding natural language commands for mobile service robots. CoRR **abs/1807.03053** (2018), `https://arxiv.org/abs/1807.03053`
21. Matamoros, M., Rascon, C., Hart, J., Holz, D., Beek, L.: RoboCup@Home 2018: Rules and Regulations (2018), `http://www.robocupathome.org/rules/2018_rulebook.pdf`
22. Matuszek, C., Herbst, E., Zettlemoyer, L., Fox, D.: Learning to parse natural language commands to a robot control system. In: Experimental Robotics. pp. 403–415 (2013)
23. Mei, H., Bansal, M., Walter, M.R.: Listen, Attend, and Walk: Neural Mapping of Navigational Instructions to Action Sequences. In: AAAI. vol. 1, p. 2 (2016)
24. Misra, D.K., Sung, J., Lee, K., Saxena, A.: Tell me Dave: Context-sensitive grounding of natural language to manipulation instructions. The International Journal of Robotics Research **35**(1-3), 281–300 (2016)
25. Nyga, D., Roy, S., Paul, R., Park, D., Pomarlan, M., Beetz, M., Roy, N.: Grounding Robot Plans from Natural Language Instructions with Incomplete World Knowledge. In: Conf. Robot Learning. pp. 714–723 (2018)
26. Otter, D.W., Medina, J.R., Kalita, J.K.: A Survey of the Usages of Deep Learning in Natural Language Processing. CoRR **abs/1807.10854** (2018), `https://arxiv.org/abs/1807.10854`
27. Paul, R., Arkin, J., Aksaray, D., Roy, N., Howard, T.M.: Efficient grounding of abstract spatial concepts for natural language interaction with robot platforms. The Int. Journal of Robotics Research **37**(10), 1269–1299 (2018)
28. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. CoRR **abs/1802.05365** (2018), `https://arxiv.org/abs/1802.05365`
29. Sales, J.E., Freitas, A., Handschuh, S.: An open vocabulary semantic parser for end-user programming using natural language. In: Semantic Computing (ICSC), 2018 IEEE 12th Int. Conf. pp. 77–84 (2018)
30. Shah, P., Fiser, M., Faust, A., Kew, J.C., Hakkani-Tur, D.: FollowNet: Robot Navigation by Following Natural Language Directions with Deep Reinforcement Learning. CoRR **abs/1805.06150** (2018), `https://arxiv.org/abs/1805.06150`
31. Sugiura, K., Kawai, H.: Grounded language understanding for manipulation instructions using GAN-based classification. In: Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE. pp. 519–524 (2017)
32. Vanzo, A., Iocchi, L., Nardi, D., Memmesheimer, R., Paulus, D., Ivanovska, I., Kraetzschmar, G.K.: Benchmarking Speech Understanding in Service Robotics. In: 4th Int. Workshop Artificial Intelligence and Robotics (AIxIA). vol. 2054, pp. 34–40 (2017)

33. Wiriyathammabhum, P., Summers-Stay, D., Fermüller, C., Aloimonos, Y.: Computer vision and natural language processing: recent approaches in multimedia and robotics. ACM Computing Surveys (CSUR) **49**(4), 1–44 (2017)
34. Wu, L., Fisch, A., Chopra, S., Adams, K., Bordes, A., Weston, J.: StarSpace: Embed All The Things! CoRR **abs/1709.03856** (2017), `https://arxiv.org/abs/1709.03856`
35. Zhong, J., Ogata, T., Cangelosi, A., Yang, C.: Understanding Natural Language Sentences with Word Embedding and Multi-modal Interaction. Development and Learning and Epigenetic Robotics (ICDL-Epirob) (2017)