Finding Your Way with Words: A Look at Natural Language in Robotics

by

Harel Biggie

B.A., University of Rochester, 2018

An area exam submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2023

This thesis area exam:
Finding Your Way with Words: A Look at Natural Language in Robotics
written by Harel Biggie
has been approved for the Department of Computer Science

_____

Dr. Christoffer Heckman

_____

Dr. Bradley Hayes

_____

Dr. James Martin

Date _____

Biggie, Harel (Ph.D., Computer Science)

Finding Your Way with Words: A Look at Natural Language in Robotics

Area Exam directed by Prof. Dr. Christoffer Heckman

Contents

Figures

Figure

Chapter 1

Introduction

Autonomous robots are continuously being deployed in more complex environments for safety-critical missions such as search and rescue. Advancements in perception, path planning, mapping, and autonomy have enabled teams of robots and humans to operate in complex subterranean environments at the DARPA Subterranean Challenge [5] with varying degrees of success. Even the top-performing teams at these events were only able to perform at a 50% success rate. In many situations, robots needed help from a human supervisor who gave inputs using a cumbersome waypoint-based interface. A chief limitation of most of these state-of-the-art systems was the ability of humans to send high bandwidth instructions in the form of natural language commands to the robots. The lack of this ability both prevents efficient supervision of the system and the ability for non-robotic experts to interact with the system. This survey will examine how other robotic systems have been connected to natural language with an emphasis on robotic navigation tasks.

Additionally, humans reason over landmarks using cognitive maps [31] rather than the traditional metric-based methods used in robotics. As such, translation mechanisms or reformulations are needed to transform traditional metric-based localization systems used by robots [13, 34, 42]. Natural language provides compelling mechanisms to help shift robots away from requiring metric-based localization in order to understand their environments.

Using natural language to communicate with robots can largely be divided into two facets [44]. The first is the challenge of language understanding which involves extracting meaning from

language as well as the meaning's relation to the physical world. This is usually known as grounding where a language utterance is interpreted in terms of both the robot's physical state and its surrounding environment. The second facet involves language generation where the robot interprets its surroundings and is able to communicate information about its state and its environment using natural language. While both of these facets share similar challenges, this review will focus on the language understanding problem within the context of robot navigation.

1.0.1 Preliminaries

Language grounding: or situated language refers to the challenge of interpreting language within the context of the physical world [12]. Regarding robotics, groundings can represent objects in the physical world, motor commands, and sequences of actions [33]. For example, a sentence such as "Go to the backpack on your right" would require a robot to use a camera (or similar) to determine that a backpack exists. The robot would then need a method to determine where the backpack is which it could do with a range sensor. Finally, the "Go to" part of the sentence would tell the robot to start walking or turning its wheels to reach the destination. Figure 1.1 shows the components that are needed for language understanding within the context of mobile robots.

How Language Can Be Represented The linguistics and cognitive science communities have spent extensive time studying the structure of natural language [24]. Natural language has a compositional syntax that follows a hierarchical structure which enables the creation of novel utterances. In certain cases this syntax can be exploited to create parse trees such as Context-Free Grammars (CFGs) [14]. CFGs divide sentences into determiners (DT), nouns (NN), noun phrases (NP), verbs (VB), and verb phrases (VP). From there, a parse tree can be created using the verb or verb phrase as the root. An example parse tree for the sentence "Grab an apple.", is shown in Figure 1.2.

Parse trees are incredibly useful because they can be formally verified which enables the creation of logical programs. For example, λ-calculus can be used to express a parse tree in terms of function arguments. Taking the same sentence the λ-calculus expression $\ni$ x(APPLE $\wedge$ GRAB(X))

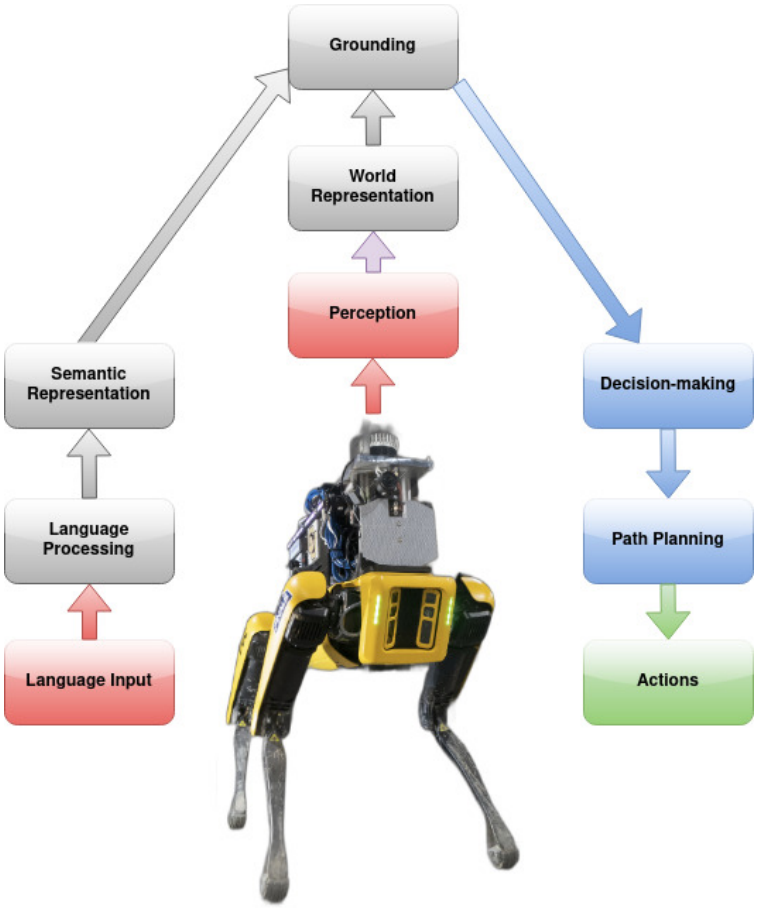


Figure 1.1: A system diagram showing the necessary components for visual language navigation in robotics.

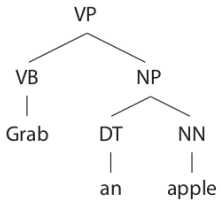


Figure 1.2: Example CFG for the sentence "Grab an apple." taken from [44]. The utterance is decomposed into the following: determiners (DT), nouns (NN), noun phrases (NP), verbs (VB), and verb phrases (VP).

can be made. This expression states that there exists an apple (x) that is being grabbed. Formal structures like these are important to robotics because navigation requires a degree of certainty so that the robot both drives to the correct location and does so in a safe manner.

CFGs have also successfully been converted into temporal logic. Temporal logic is a series of logical statements that are evaluated in sequence for an overall boolean true or false value for the entire statement [9, 25]. Taking a look at the sentence "Drive to the backpack", we can express it as $\Diamond$AtBackpack. AtBackpack is now a boolean proposition that can only be true when the robot arrives at the backpack. These types of formalization both ensure the robot will do what is expected of it and reduce the ambiguity naturally presented in language.

While these methods haven't been directly deployed within the context of robotic navigation, they highlight some of the early work toward goals researchers are trying to achieve with more modern methods. These early language parsing techniques are largely incompatible with robotics because they don't capture more complex language such as spatial relations and they also do not address the symbol grounding problem.

More modern language parsing techniques tend to rely on global vector representations [23] and contextual embeddings [27]. Of note, is the shift towards global embeddings such as Global vectors for word representations (GLOVE) [36] and Bidirectional Encoder Representations from Transformers (BERT) [7] which are used by many of the top language parsing systems.

In the rest of this review, we will examine three styles of approaches that are commonly used to connect natural language to robotics. First, we will look at logic-based methods and then we will look at probabilistic methods. Finally, we will take a look at large language model-based approaches.

Chapter 2

Probabilistic Models

In recent years, the challenge of grounding language to the physical world has been addressed by inferring over probabilistic graphs. [45, 16]. Early works in this domain collected a corpus of natural language directions using a user study where humans walked through the MIT campus and provided directions between rooms [21]. All of the subjects were familiar with the area before they were instructed to create instructions. In this work, the corpus was parsed into a sequence of Spatial Description Clauses (SDC). An SDC is a graphical model that consists of a *figure* or the subject of the sentence, a *verb* which is the action, and a spatial relation between the environment and the figure. An example of an SDC can be seen in Figure 2.1

The SDC planning system described in [21] then builds a topological map using inputs from a lidar and a camera. Inference is performed on the language by:

$$\arg\max_{p} p(P, S|O) = P(S|P, O) \times p(P|O) \tag{2.1}$$

where P is the path, S is a sequence of SDCs and O represents the observed landmarks. The authors factor Equation 2.1 into four parts which correspond to each field of the SDC. The authors solve the action grounding problem by limiting the robot to a series of control sets that can go left, right, or straight. The method can either operate in a global mode if an *a priori* map is known or it can greedily explore if the map is unknown. The overall success rate of this method in finding the "best path" which the authors describe as the path that ended up closest to the true destination was 59.3% percent.

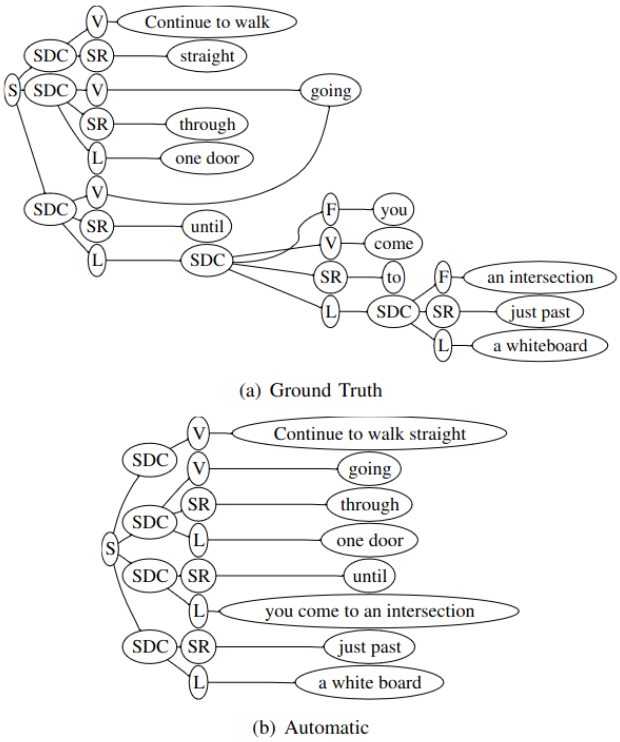


Figure 2.1: An example Spatial Description Clause for the sentence "Continue to walk straight, going through one door until you come to an intersection just past a whiteboard". The top shows the ground truth hand-annotated SDC while the bottom shows an SDC generated by an automatic parser as described in [21]. The figure is also taken from [21]

Overall, this early work sets a baseline for grounding graph-based methods but it is limited to sequentially structured language and is unable to handle nested clauses. In other words, when given a sentence such as: "Put the pen into the backpack" this method looks at the entire sentence in a flat way and is unable to separate the two arguments for "put". Both the backpack and pen are necessary to comprehend the true meaning of the sentence. "Put" requires two arguments and this method would produce two independent SDCs rather than a joint one.

An extension to the SDC approach for grounding language is presented in [45, 22] where a Generalized Grounding Graph (G^3) is introduced. The authors define a *grounding graph* as a dynamically instantiated probabilistic graph that takes the hierarchical structure of natural language command. Example graphs can be seen in Figure 2.2.

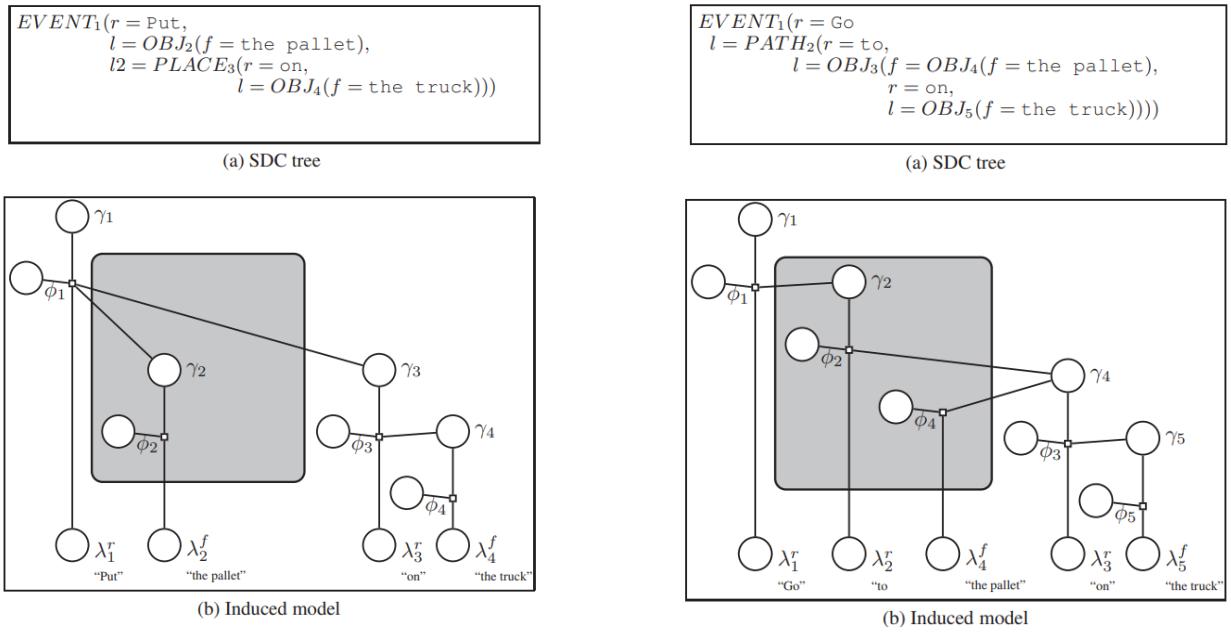


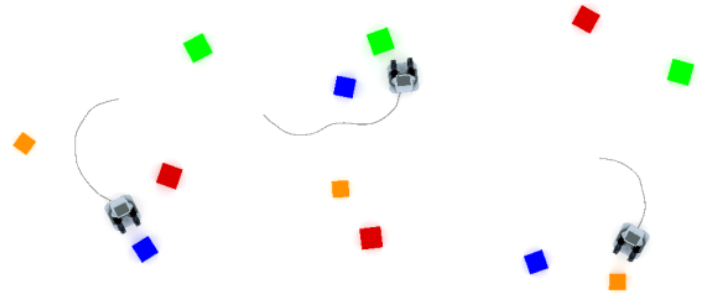
Figure 2.2: Examples of the G^3 model taken from [45]. The left is the parse tree for the sentence "Put the pallet on the truck." and the right is the parse tree for the sentence "Go to the pallet on the truck.

In order to bound the search space the authors limit the action sequences to events, objects, places, and paths. Events are in the form of an action sequence and objects are things in the world. Paths represent words that define a way to traverse through the world such as "towards". The inspiration for these classifications comes from early work on semantics and cognition presented in [19].

In the grounding graph, two types of nodes are present: random variables and factors. Each random variable is binary and can either be true or false. Groundings are represented as γ and the random variable ϕ_i is true if it corresponds to the grounding γ_i. There are random variables λ_i^f, λ_i^r, λ_i^l which correspond to figures, relations, and landmarks.

Generalized graphs can be automatically generated using conditional random fields. The system was able to learn the meaning of words such as "put" by pairing them with the corresponding robot action. Despite promising early results G^3 graphs still face many limitations. For instance, can't represent anaphora and negations. The search space also grows exponentially as the language present in the domain increases.

Distributed Correspondence Graphs (DCG) [16] are an extension to G^3 graphs that discretize the space of groundings into regions and motion constraints. These graphs also take advantage of only sampling plans that are near objects in the environment. This can be seen in Figure 2.3.



(a) "go to the blue box" (b) "move towards the green object" (c) "travel to the orange object"

Figure 2.3: Examples of planned paths using the DCG method presented in [16]. The figure is also taken from [16]

Like generalized grounding graphs, the DCG model can support infinite recursion of noun

phrases. Additionally, the method is agnostic to the planning method because it is only used to form constraints for a planner. The method only works when all constraints intersect each other and is therefore incompatible with complex spatial language. Additionally, the method requires training data that aligns with the domain and the data needs to be hand annotated.

Chapter 3

Abstract Meaning Representations

Abstract Meaning Representations (AMRs) are a series of semantic representations for language [1]. The resulting structures can be interpreted in both logical and graph-based formats as seen in Figure 3.1. AMRs are a popular method for parsing language because they consider the whole-sentence parsing task rather than separating out parsing into identifying nouns, verbs, etc. This approach produces better results than when each individual task is solved in isolation [1].

LOGIC format:

$\exists$ w, b, g:
instance(w, want-01) $\wedge$ instance(g, go-01) $\wedge$
instance(b, boy) $\wedge$ arg0(w, b) $\wedge$
arg1(w, g) $\wedge$ arg0(g, b)

AMR format (based on PENMAN):

```
(w / want-01
  :arg0 (b / boy)
  :arg1 (g / go-01
          :arg0 b))
```

GRAPH format:

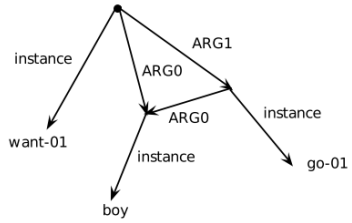


Figure 3.1: Example AMR formatting for the sentence "They boy wants to go". Figure taken from [1].

AMRs are rooted and directed graphs where both edges and leaves are labeled. AMRs

contain English words such as “robot”, PropBank framesets [20], or special keywords. PropBank framesets are a series of labels for syntactic nodes with specific argument labels that are able to ensure similar sentences resolve to the same arguments. For example, the sentence “The robot broke” would resolve to the same role sets as “The robot was broken by Jim.” Each frameset has a series of frame arguments that are labeled in the following convention: :arg0, :arg1, :arg2, :arg4, :arg5. Special keywords can include general semantic relations such as destination or accompanier. These are particularly useful in robotics because they have the potential to resolve ambiguity when directing a mobile robot.

In recent years extensions have been made to the AMR framework enabling the structure to be used in similar tasks to robotic navigation [2, 3]. Traditional AMRs are limited to a small subset of spatial relations such as **:destination**, **:path**, **:location**. They can account for relations that follow a preposition with one argument format and have a single reference frame. However, with more complex language such as ”Go to the right of the backpack that is on your right,” the basic role sets fail to account for the fact that the backpack is both to the right of you and that you need to end up to the right of the backpack. Spatial AMRs extend the AMR role sets to account for multiple spatial relations.[3].

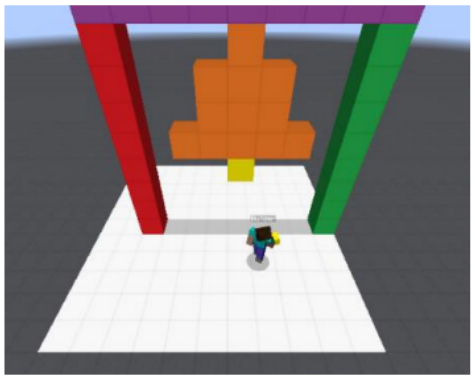


Figure 3.2: Example scene from a 3D mincraft world. The figure is from [3].

The training corpus is taken from a 3D Minecraft world as seen in Figure 3.2. The new role sets directly enable the movement in 3D-like grid worlds which is a typical methodology for

representing the environment within the context of mobile robot navigation [15]. The ability of Spatial AMRs to resolve language that details 3D grid-like worlds makes them a promising choice for the language grounding block shown in Figure 1.1.

An additional extension to the AMR framework known as Dialog-AMRs [2] also directly relates to mobile robot navigation. The paper expands the traditional AMR role sets to allow for dialog conversations between robots and humans by capturing the speaker's intent. The work also helps reduce the amount of manual annotation required for AMR generation. This is done using a graph-to-graph transformation which is then mapped to a robot behavior as shown in Figure 3.3

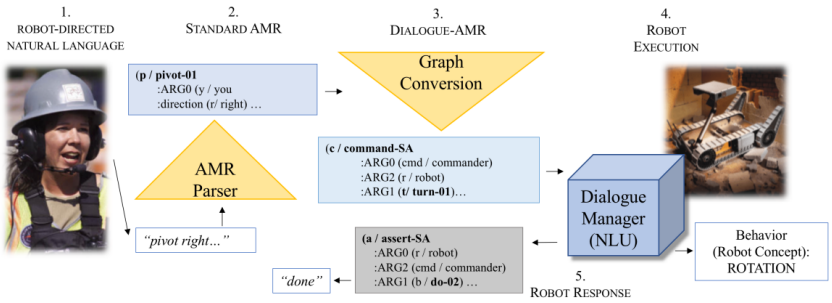


Figure 3.3: Dialog AMR graph generation pipeline using a graph to graph transformer. Image is taken from [2]

.

The training dataset for Dialog AMRS was taken from the Situated Corpus of Understanding Transactions (SCOUT) dataset which contains 80 hours of human-robot dialogue from 83 participants [30, 29]. Data was collected using a "Wizard-of-Oz" style experiment where humans acted as surrogates for the robots.

An example of the additional information captured by a dialog AMR can be seen in Figure 3.4. Here we can see that the AMR captures both the speaker as well as who the command is intended for which in this case is the robot. It also captures temporal information such as the sequence a robot would have to follow to complete the task.

The ability of AMRs to resolve complex language phenomena such as spatial relations and dialogues makes them a promising candidate for future mobile robot with natural language navigation research. Previous structures such as SDCs are unable to capture the meaning of more

```
(a) (d / drive-01 :mode imperative
        :ARG0 (y / you)
        :destination (d2 / door))
(b) (c / command-SA
        :ARG0 (c2 / commander)
        :ARG2 (r / robot)
        :ARG1 (g / go-02 :completable +
           :ARG0 r
           :ARG3 (h / here)
           :ARG4 (d/ door)
           :time (a2 / after
                :op1 (n / now))))
```

Figure 3.4: Example taken from [2] of a dialog AMR for the sentence "Drive to the door."

.

complicated spatial relations. While AMRs have not yet directly been used on mobile robots, the concept presents promising future research opportunities.

Chapter 4

Large Language Models

Large language models (LLM) have recently entered the spotlight with the release of ChatGPT [10]. LLMs refer to language models that contain tens to hundreds of billions of parameters [48]. Traditionally, these models have been limited to text-based inputs and outputs but recently multimodal versions have emerged that are able to take in images [35], embodied sensor data [8], and even generate arbitrary programs [43]. LLMs have shown a phenomenal ability to generalize across different contexts which greatly reduces the need to hand-label training data. In this section, we will explore how LLMs are being used in robotics and their future potential for them.

Visual-language models (VLMs) are extremely capable of generalizing pre-trained data available from the Internet such as pictures with captions to other domains [37, 26]. For instance, LSeg [26] is a transformer-based encoder that performs pixel-level semantic segmentation on images even when the caption is unknown. An example of this can be seen in Figure 4.1.

In stark contrast to traditional Convolutional Neural Network (CNN) based visual detection methods such [46, 28, 38] these methods do not require manually annotating data. For robotics, this is a potentially game-changing capability as robots will frequent unknown environments to them when they are navigating through the world. Recently, visual language maps for robot navigation have been created by exploiting the transformer-based architectures of LLMs [17, 41, 11].

LM-Nav [41] is a method of doing robotic navigation that is trained on unannotated large datasets of trajectories. LM-Nav combines a vision model (CLIP) [37], a language model (GPT-3) [10], and a robot navigation model ViNG [40]. ViNG is a learning-based navigation system that

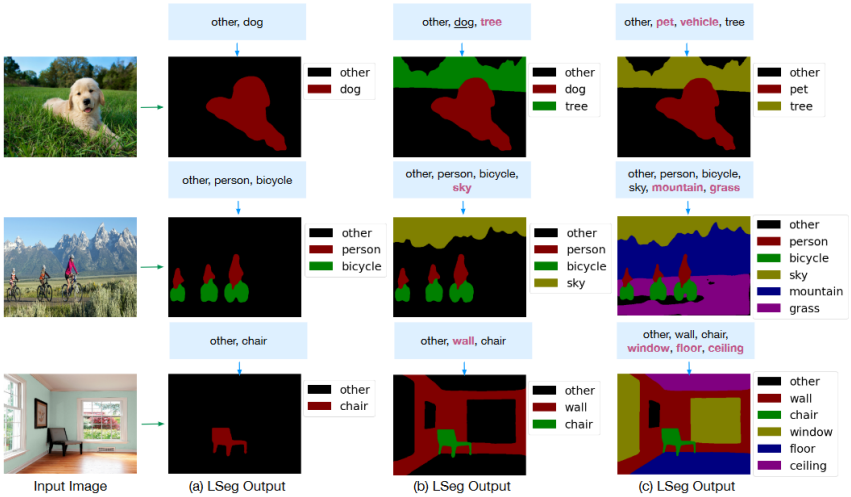


Figure 4.1: Example pixel level segmentation learned by the Lseg method. Image is taken from [26] Of note, is that the method is able to handle labels that were not a part of the training set.

is capable of reaching visually indicated goals. ViNG is capable of building a plan by learning a topological graph from ego-centric images. The topological graphs generated by the method are only associated with image frames and do not form a bird's eye view. An example path can be seen in Figure 4.2. ViNG is capable of successfully navigating using only offline experience but it can also adapt to new environments including season changes.

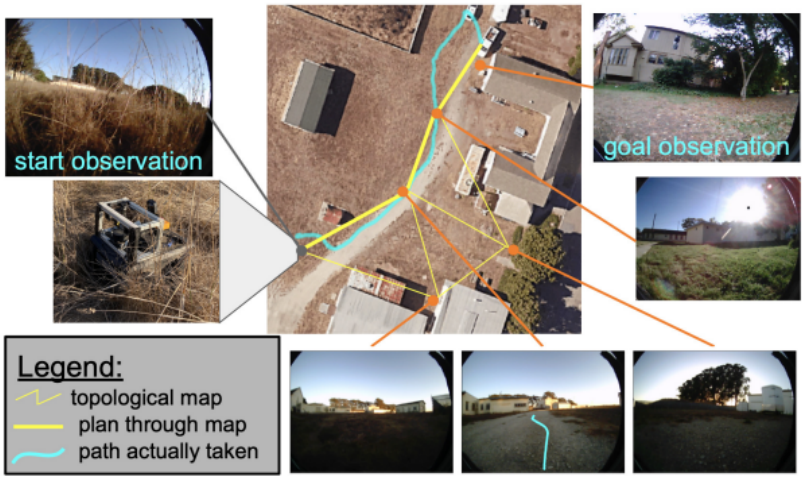


Figure 4.2: Example path generated by Ving [40].

LM-Nav extends ViNG by leveraging LLMs to navigate in novel environments. LM-Nav uses

GPT-3 to parse instructions into a list of landmarks. The landmarks are then passed into CLIP to ground the landmarks to image space. ViNG is then used to predict the capability of the robot to navigate between each pair of nodes in the generated graph. This is done using temporal distances and the actions are then generated on a real robot. The full pipeline can be seen in Figure 4.3

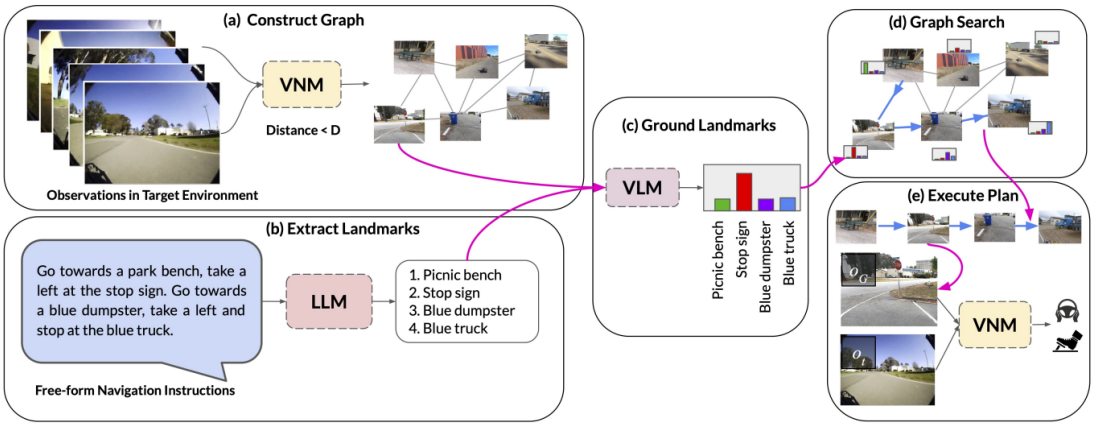


Figure 4.3: Pipeline showing the LM-Nav architecture as seen in [41].

Lm-Nav was implemented on a Clearpath Jackal UGV. A key limitation of the system is that the GPT and CLIP queries need to be pre-computed on a remote computer. This is due to computing constraints on board the robot. LM-Nav also heavily relies on landmarks and doesn't account for other verbs in a sentence. As such the method doesn't know how to "drive the robot slowly past the dog". Slow isn't inferable from an image and therefore is unknown to this architecture.

An alternative navigation approach is presented in [17] which emphasizes the generation of spatial maps. VLMaps [17] are generated directly from videos and are indexed by natural language. Specifically, they are created by fusing pre-trained visual-language features with reconstructions of the environment. The system uses a Lseg visual encoder to create a segmented map. Landmarks are indexed using a text encoder. The full architecture can be seen in Figure 4.4

Experiments are run on 91 different task sequences. Starting positions are randomly specified and there are four subgoals that the robot needs to navigate towards.+ Competing methods such as LM-Nav are only able to go to 2 subgoals. VLMap outperforms LM-Nav by a significant margin

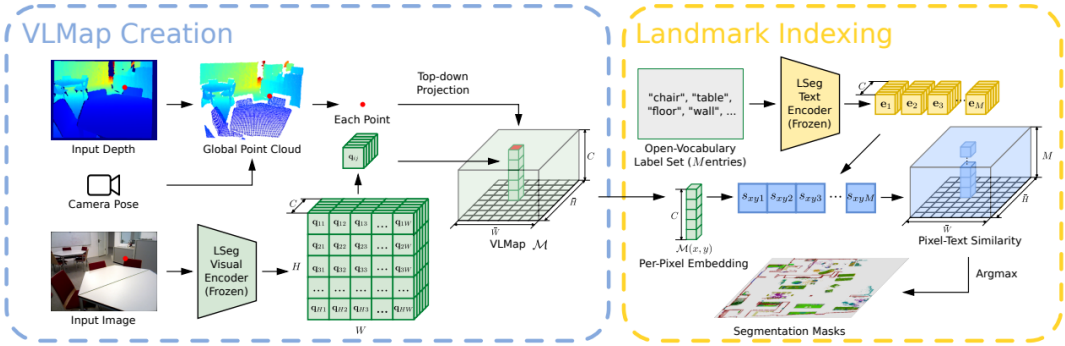


Figure 4.4: The full architecture for generating VLmaps as described in [17].

(62 vs 5 successful tasks) when considering subgoals. An example map generated by VLMap can be seen in Figure 4.5

Additional methods have leveraged CLIP [11] to perform zero-shot robotic navigation. For instance, CLIP on Wheels builds a saliency map which it uses to generate a segmentation mask. Specifically, Grad-CAM [39], a gradient-based interpretability localization map generator is leveraged to build the mask. That mask is then planned over using a frontier-based [47] exploration method. Experiments were performed both in simulation and on an HSR mobile robot.

Perhaps the most intriguing example of LLMs being used in the context of robotics comes from the PaLM-E model [8]. PaLM-E is a 562 billion parameter model that is trained on robotic tasks but can also perform Visual Question and Answering. Palm-E's inputs consist of text and continuous observations which are interleaved within the text. The model interleaves observations that happen between two image frames in the encoded input. Palm-E's output is text that conditions low-level commands.

Palm-E is also able to learn across different domains. The authors observe that the model performs over 50% better when using text and visual observations than using either domain alone. This is shown in Figure 4.6.

Palm-E experiments are run in a Task and Motion Planning Environment (TAMP). Example tasks include opening drawers and moving blocks around. Palm-E is able to perform zero-shot

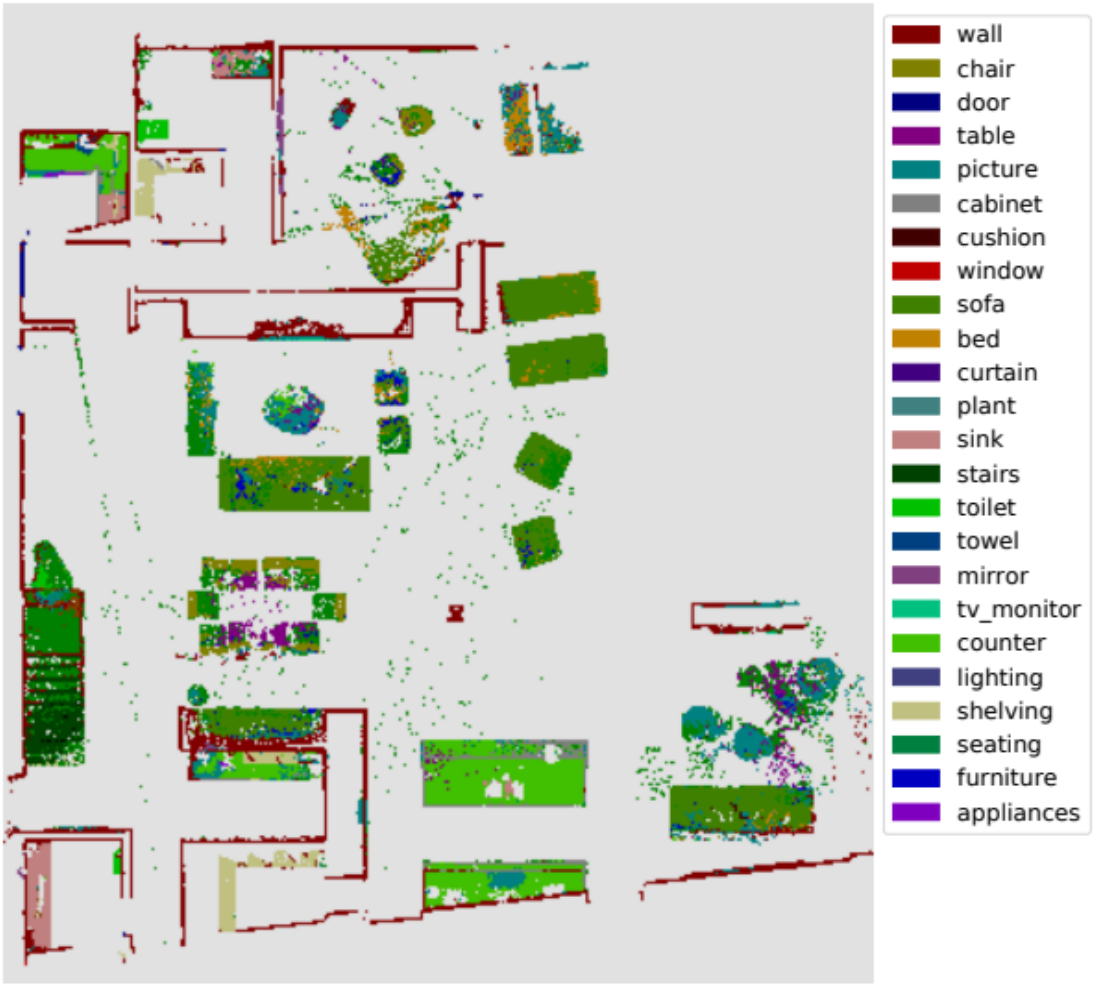


Figure 4.5: Example VLMap from [17].

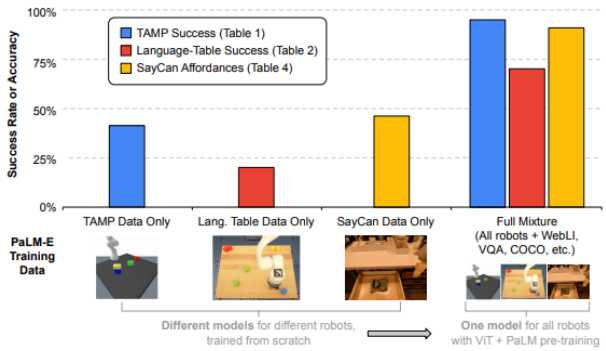


Figure 4.6: Palm-E is able to successfully learn across different domains [8].

planning. Palm-E is also capable of chaining together multiple sequences of instructions.

ViperGPT [43] is a model that is able to perform visual inference and reasoning by executing Python programs. ViperGPT combines a series of models and generates Python code using GPT-3 Codex [4]. An example is shown in Figure 4.7 This allows for the programmatic composition of specialized vision, language, math, and logic functions. Importantly any module can be substituted because ViperGPT is only generating the specification for the program rather than the implementation. For robotics, this is important because it enables traceability and explainability. When end-to-end methods fail it is difficult to understand why. However, by combining other approaches with ViperGPT it may be possible to see what triggers failures and also provide a level of assurance that a robot will actually perform the task that was requested of it.

Figure 4.7: Example generated code from ViperGPT [43].

Chapter 5

Conclusion

Using natural language with robotics has come a long way since early logic-based approaches. Grounding graph-based methods were some of the pioneering work in navigating with physical robots but they rely on hand-annotated data which makes it hard for them to generalize. However, they provide confidence scores which makes it much easier to understand system-level failures. AMRs have the potential to be the next form of symbolic grounding because of their widespread adoption and ability to capture complex language structures. Datasets already exist for AMRs in the robotic context which makes them a promising choice for future research. Finally, LLMs show incredible promise but the sheer size of the models makes it difficult to deploy them on real robots without remote computing. Additionally, many of these methods are not explainable and do not provide any safety guarantees. Overall, it is still to be determined which family of methods will stand up to the test of time.

Bibliography

[1] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In Proceedings of the 7th linguistic annotation workshop and interoperability with discourse, pages 178–186, 2013.

[2] Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. Dialogue-amr: abstract meaning representation for dialogue. In Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 684–695, 2020.

[3] Julia Bonn, Martha Palmer, Jon Cai, and Kristin Wright-Bettner. Spatial amr: Expanded spatial annotation in the context of a grounded minecraft corpus. In Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020),, 2020.

[4] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374, 2021.

[5] DARPA. DARPA Subterranean Challenge, 2022. https://www.darpa.mil/program/darpa-subterranean-challenge.

[6] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, et al. Robothor: An open simulation-to-real embodied ai platform. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 3164–3174, 2020.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[8] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. arXiv preprint arXiv:2303.03378, 2023.

[9] E Emerson. Temporal and modal logic in handbook of theoretical computer science, chapter 16, leeuwen j., editor, 1990.

[10] Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. Minds and Machines, 30:681–694, 2020.

[11] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Clip on wheels: Zero-shot object navigation as object localization and exploration. arXiv preprint arXiv:2203.10421, 2022.

[12] Stevan Harnad. The symbol grounding problem. Physica D: Nonlinear Phenomena, 42(1-3):335–346, 1990.

[13] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar slam. In 2016 IEEE international conference on robotics and automation (ICRA), pages 1271–1278. IEEE, 2016.

[14] John E Hopcroft, Rajeev Motwani, and Jeffrey D Ullman. Introduction to automata theory, languages, and computation. Acm Sigact News, 32(1):60–65, 2001.

[15] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. Autonomous robots, 34:189–206, 2013.

[16] Thomas M Howard, Stefanie Tellex, and Nicholas Roy. A natural language planner interface for mobile manipulators. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 6652–6659. IEEE, 2014.

[17] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. arXiv preprint arXiv:2210.05714, 2022.

[18] Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. arXiv preprint arXiv:2212.10403, 2022.

[19] Ray S Jackendoff. Semantics and cognition, volume 8. MIT press, 1985.

[20] Paul R Kingsbury and Martha Palmer. From treebank to propbank. In LREC, pages 1989–1993, 2002.

[21] Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. Toward understanding natural language directions. In 2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 259–266. IEEE, 2010.

[22] Thomas Kollar, Stefanie Tellex, Matthew Walter, Albert Huang, Abraham Bachrach, Sachi Hemachandra, Emma Brunskill, Ashis Banerjee, Deb Roy, Seth Teller, et al. Generalized grounding graphs: A probabilistic framework for understanding grounded commands. arXiv preprint arXiv:1712.01097, 2017.

[23] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. Information, 10(4):150, 2019.

[24] Angelika Kratzer and HEIM Irene. Semantics in generative grammar, volume 1185. Blackwell Oxford, 1998.

[25] Hadas Kress-Gazit, Morteza Lahijanian, and Vasumathi Raman. Synthesis for robots: Guarantees and feedback for robot behavior. Annual Review of Control, Robotics, and Autonomous Systems, 1:211–236, 2018.

[26] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. arXiv preprint arXiv:2201.03546, 2022.

[27] Qi Liu, Matt J Kusner, and Phil Blunsom. A survey on contextual embeddings. arXiv preprint arXiv:2003.07278, 2020.

[28] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, pages 21–37. Springer, 2016.

[29] Matthew Marge, Claire Bonial, Brendan Byrne, Taylor Cassidy, A William Evans, Susan G Hill, and Clare Voss. Applying the wizard-of-oz technique to multimodal human-robot dialogue. arXiv preprint arXiv:1703.03714, 2017.

[30] Matthew Marge, Claire Bonial, Ashley Foots, Cory Hayes, Cassidy Henry, Kimberly Pollard, Ron Artstein, Clare Voss, and David Traum. 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, 2017.

[31] Timothy P McNamara, James K Hardy, and Stephen C Hirtle. Subjective hierarchies in spatial memory. Journal of Experimental Psychology: Learning, Memory, and Cognition, 15(2):211, 1989.

[32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

[33] Raymond J Mooney. Learning to connect language and perception. In AAAI, pages 1598–1601. Chicago, 2008.

[34] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. IEEE transactions on robotics, 31(5):1147–1163, 2015.

[35] OpenAI. Gpt-4 technical report, 2023.

[36] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543, 2014.

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In International conference on machine learning, pages 8748–8763. PMLR, 2021.

[38] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.

[39] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision, pages 618–626, 2017.

[40] Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. Ving: Learning open-world navigation with visual goals. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pages 13215–13222. IEEE, 2021.

[41] Dhruv Shah, Błażej Osiński, Sergey Levine, et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In Conference on Robot Learning, pages 492–504. PMLR, 2023.

[42] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS), pages 5135–5142. IEEE, 2020.

[43] Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. arXiv preprint arXiv:2303.08128, 2023.

[44] Stefanie Tellex, Nakul Gopalan, Hadas Kress-Gazit, and Cynthia Matuszek. Robots that use language. Annual Review of Control, Robotics, and Autonomous Systems, 3:25–55, 2020.

[45] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew Walter, Ashis Banerjee, Seth Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 25, pages 1507–1514, 2011.

[46] Xiu-Shen Wei, Chen-Wei Xie, Jianxin Wu, and Chunhua Shen. Mask-cnn: Localizing parts and selecting descriptors for fine-grained bird species categorization. Pattern Recognition, 76:704–714, 2018.

[47] Brian Yamauchi. A frontier-based approach for autonomous exploration. In Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation', pages 146–151. IEEE, 1997.

[48] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. arXiv preprint arXiv:2303.18223, 2023.