



# Enhancing Speech Instruction Understanding and Disambiguation in Robotics via Speech Prosody

David Sasu<sup>1</sup>, Kweku Andoh Yamoah<sup>2</sup>, Benedict Quartey<sup>3</sup>, Natalie Schluter<sup>1,\*</sup>

<sup>1</sup>Computer Science, IT University of Copenhagen, Denmark

<sup>2</sup>Intelligent Agents Research Group, University of Florida, U.S.A

<sup>3</sup>Intelligent Robot Lab, Brown University, U.S.A

dasa@itu.dk, kyamoah@ufl.edu, benedict\_quartey@brown.edu, nael@itu.dk

## Abstract

Enabling robots to accurately interpret and execute spoken language instructions is essential for effective human-robot collaboration. Traditional methods rely on speech recognition to transcribe speech into text, often discarding crucial prosodic cues needed for disambiguating intent. We propose a novel approach that directly leverages speech prosody to infer and resolve instruction intent. Predicted intents are integrated into large language models via in-context learning to disambiguate and select appropriate task plans. Additionally, we present the first ambiguous speech dataset for robotics, designed to advance research in speech disambiguation. Our method achieves 95.79% accuracy in detecting referent intents within an utterance and determines the intended task plan of ambiguous instructions with 71.96% accuracy, demonstrating its potential to significantly improve human-robot communication.

**Index Terms:** speech recognition, human-computer interaction, computational paralinguistics, speech prosody, robot planning, speech understanding, speech disambiguation

## 1. Introduction

Understanding and executing natural language instructions is a fundamental goal in human-robot interaction. To accurately follow instructions, robots must grasp several key elements such as the speaker’s intent, relevant referents, and execution constraints. However, instructions conveyed through text alone often fail to capture the wide variety of meanings that people intend. This limitation arises because rich speech prosodic cues—such as intonation, stress, and rhythm—that significantly affect the meaning of instructions are lost in textual representations. Consider the ambiguous instruction shown in Figure 1: “Place the coke can beside the pringles on the counter”. One interpretation of this task is to place the coke can at a location beside the pringles, which happens to be on a counter. Another interpretation is to find the coke can (which is currently beside the pringles) and move it to the counter. Disambiguating such instructions relies on prosodic cues like pauses and emphasis, which convey the intended meaning.

Current speech to motion robot systems predominantly rely on automatic speech recognition (ASR) to transcribe speech into text before processing [1, 2, 3]. While effective at capturing the literal words, ASR strips away essential prosodic cues like stress, rhythm, and intonation. This loss renders robots incapable of interpreting the speaker’s true intent in ambiguous or complex instructions. Existing approaches attempt to resolve ambiguity through contextual or semantic analysis of the transcribed text,

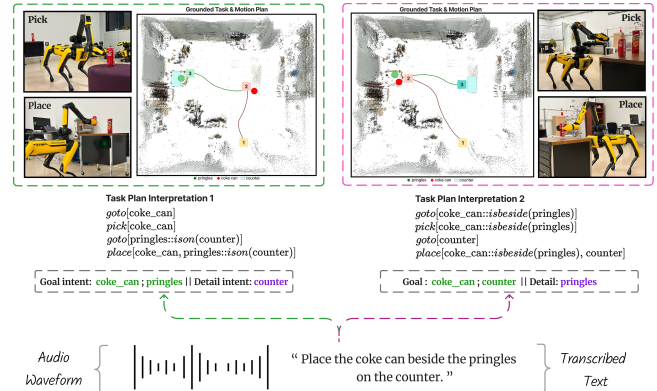


Figure 1: Illustration of two distinct interpretations of the instruction “Place the coke can beside the pringles on the counter.” **Left (green):** The robot picks up the coke can and places it next to the pringles on the counter. **Right (pink):** The robot locates the coke can already beside the pringles and relocates it onto the counter. Each plan is represented by a pointcloud of the scene and the robot’s motion path, while the robot images illustrate the pick-and-place actions.

but they fall short when critical information is embedded in the tonal aspects of speech [4, 5].

To address this limitation, we propose a novel approach that explicitly leverages speech prosody to resolve ambiguities in spoken robot instructions. Concretely, we extract prosodic features from speech and employ an architecture that assigns an **intent label**—either **goal** (entities or locations to be acted upon) or **detail** (qualifiers specifying attributes or relationships)—to each word in an utterance. Reexamining the earlier example, in the first interpretation “coke can” and “pringles” would be classified as **goal intent referents**, while “counter” is a **detail intent referent** that qualifies the location of “pringles”. In the alternative interpretation “coke can” and “counter” would be classified as **goal**, and “pringles” as a **detail** specifying the location of the coke can. We illustrate this distinction in interpretation and equivalent task plans in Figure 1. We make three main contributions in this work:

1. **Prosody-Aware Architecture.** We introduce an encoder-decoder model that exploits prosodic features for token-level intent classification, capturing long-range dependencies within spoken instructions.
2. **LLM Integration.** We demonstrate how prosody-driven intent predictions can be used with large language models, to select appropriate task plans from multiple candidates, bridging the gap between ambiguous human instructions and precise robot execution.

\* Currently at Apple.

3. **Ambiguous Speech Dataset.** We highlight speech instruction disambiguation as a key research area in robotics and curate a novel dataset of **1,540** ambiguous utterances, to advance future work.

## 2. Related works

Recent works have demonstrated the effectiveness of incorporating prosody into speech understanding tasks such as intent classification [6, 7, 8, 9]. For instance, Wei et al. [9] proposed a neural prosody encoder that leverages prosodic information for end-to-end dialogue act classification, using a learnable gating mechanism to assess and selectively retain critical prosodic features. Other studies have explored prosody-based attention and distillation to improve end-to-end Spoken Language Understanding (SLU) [8]. While these approaches underscore the importance of prosodic features in SLU systems, they remain underexplored in human-robot interaction scenarios, where precise understanding of spoken instructions is essential for effective task execution.

Early systems for robot instruction following from language often relied on hand-crafted grammars and symbolic representations to map natural language commands to executable actions [10]. Although these rule-based methods proved useful for navigation and simple tasks, their reliance on limited vocabularies and rigid syntactic structures restricted their ability to handle the nuances of spontaneous human speech. Subsequent research shifted toward *grounded language understanding*, where language is explicitly tied to a robot’s perception and motion. Seminal works[11, 12] introduced probabilistic models that map linguistic elements—like verbs and object references—to real-world entities and spatial relations. This enabled robots to interpret instructions such as “Place the cup on the table” by identifying and manipulating the correct objects and locations. In parallel, deep learning–based approaches have expanded the scope of language instruction following by shifting from rigid grammars to data-driven end-to-end strategies that learn directly from large corpora. Benchmarks and models integrating both visual and linguistic inputs have been introduced for a variety of household tasks, demonstrating the growing synergy between perception and language processing [13, 14]. More recently, the emergence of large-scale pretrained *foundation models* has allowed robots to interpret and solve complex natural language directives without any training [15, 16, 17].

Despite these advances, a critical gap remains in the use of prosodic features present in speech—such as stress, rhythm, and intonation—to enhance instruction comprehension. Current systems predominantly rely on text transcripts, discarding the acoustic cues that humans naturally employ to convey emphasis and resolve ambiguities. We address this gap by explicitly integrating prosodic cues into the instruction-following pipeline. Our approach enables robots to more effectively interpret emphasis, disambiguate intent, and achieve a robust understanding of tonally ambiguous instructions.

## 3. Speech instruction understanding and disambiguation

We introduce Bidirectional Long Short Term memory (BiLSTM) [18] and Transformer-based [19] encoder-decoder models designed to identify referent intents by leveraging prosody. Both architectures are optimized to capture long-range dependencies in prosodic features extracted from speech, enabling the disambiguation of **goal** and **detail** intents within a given utterance.

### 3.1. Overview

For each speech sample in our dataset, we first perform **forced alignment** to determine the temporal boundaries of each word. Next, we extract **prosodic features** for each word using the Disvoice Python library <sup>2</sup>. Simultaneously, we generate a **text embedding** using OpenAI’s Text Embedding Large 3 model <sup>3</sup>. The extracted prosodic and textual features are then concatenated to form a unified feature representation. This integration is crucial as it allows the model to utilize both the semantic content of words and prosodic cues that can influence meaning and interpretation.

Formally, we represent the input sequence for the  $i$ -th speech utterance as:

$$\mathbf{X}^{(i)} = \{\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \dots, \mathbf{x}_{T_i}^{(i)}\} \quad (1)$$

where each  $x_t^{(i)}$  is the feature vector corresponding to the  $t$ -th word in the utterance, defined as:

$$x_t^{(i)} \in \mathbb{R}^{d_{\text{prosody}} + d_{\text{embed}}} \quad (2)$$

$d_{\text{prosody}}$  and  $d_{\text{embed}}$  denote the dimensionalities of the prosodic and textual embedding feature spaces, respectively. These constructed feature vectors are passed to our models, which generate a corresponding sequence of labels:

$$\mathbf{Y}^{(i)} = [y_1^{(i)}, y_2^{(i)}, \dots, y_{T_i}^{(i)}] \quad (3)$$

where each  $y_t^{(i)}$  represents the predicted label for the  $t$ -th word in the utterance. Each label indicates whether the word is *not of interest*, a **goal intent referent**, or a **detail intent referent**.

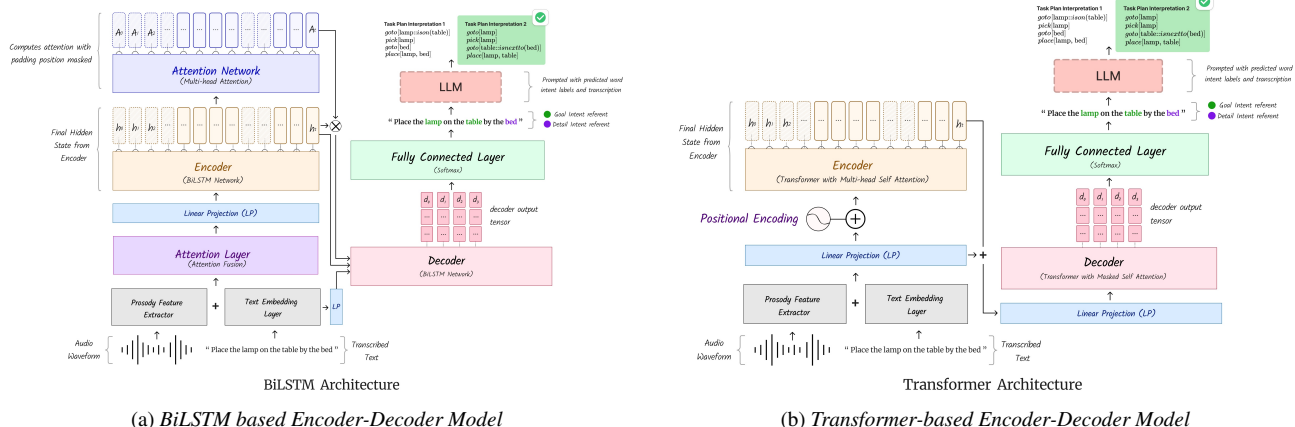
### 3.2. BiLSTM-based encoder-decoder architecture

In our BiLSTM-based architecture, input prosodic and text embedding features are first processed through an *Attention Fusion* layer, which dynamically assigns importance to different words in the utterance. Each word’s feature vector undergoes a linear transformation followed by a tanh activation to compute an attention score. The computed attention scores are normalized using a softmax function and subsequently applied to their corresponding feature vectors via element-wise multiplication. This results in a fused feature representation that preserves the sequence structure while enhancing the prominence of salient words and suppressing less informative ones. By selectively weighting the input features, the model effectively prioritizes contextually relevant patterns, improving its ability to capture fine-grained variations in speech.

The fused features are passed to the encoder, which begins with a linear projection followed by a ReLU activation to reduce the dimensionality of the input. The output of the linear projection is then fed into a stack of BiLSTM layers, which encode rich contextual representations by capturing both forward and backward dependencies within the sequence. However, since BiLSTM layers struggle to model long-range dependencies effectively [20, 19], we incorporate a *Multi-Head Attention* layer to enhance the model’s ability to capture relationships between distant elements in the sequence. The attention scores computed from the *Multi-Head Attention* layer are then applied to the final

<sup>2</sup>Link to Disvoice library: <https://disvoice.readthedocs.io/en/latest/index.html>

<sup>3</sup>Link to OpenAi’s Text Embedding Large 3 model: <https://platform.openai.com/docs/guides/embeddings>



(a) BiLSTM based Encoder-Decoder Model

(b) Transformer-based Encoder-Decoder Model

Figure 2: Overview of our architectures for word-level intent classification. (a) **BiLSTM-based Model**: Prosodic features are fused with text embeddings and fed into a BiLSTM encoder. An attention layer refines the fused representations before a decoder predicts intent labels for each token. (b) **Transformer-based Model**: Prosodic and textual inputs are combined and passed through a Transformer encoder with multi-head self-attention and positional encodings. A masked self-attention decoder then generates token-level intents. In both models, predicted intent labels are passed to a large language model (LLM) for final task plan selection.

hidden states of the BiLSTM layers via element-wise multiplication. We concatenate the resulting vectors from this computation with the original word embeddings of the utterance to obtain a final representation,  $z_t$ , that encapsulates both prosodic and textual information, ensuring that the decoder receives semantically and acoustically rich features. In the decoder, a stack of BiLSTM layers and a final fully connected layer process these features and assign labels to each word in the input utterance.

### 3.3. Transformer-based encoder-decoder architecture

Transformers inherently lack temporal inductive biases [19], thus in our Transformer-based model we first apply a linear transformation to our input prosodic and text embedding features then integrate fixed positional encodings into input features to provide word order information. This sequence is then processed through a stack of Transformer encoder layers. Each layer employs multi-head self-attention mechanisms to capture global dependencies and relationships between words, irrespective of their positional distances. The depth of the encoder—determined by the number of layers—enables the model to progressively learn abstract representations of the input data. Before decoding, we concatenate the encoder outputs with the initially projected input features and apply another linear projection. This process ensures that the decoder benefits from the detailed acoustic and semantic features present in the original input and the global contextual information encoded by the Transformer.

The decoder generates per-word labels in a left-to-right fashion, consistent with the sequential nature of the prediction task. During training, each target word’s ground truth label is shifted by one position and prefixed with a special start-of-sequence (SOS) token, which serves as the initial input to the decoder. Reintroducing the original embeddings into the decoder improves the model’s ability to capture specific word properties during label generation. The decoder uses masked self-attention to maintain the autoregressive property required for sequential prediction. This causal masking ensures that each output position can only attend to preceding positions, preventing information leakage from future tokens. Finally, a fully connected layer with softmax activation maps decoder outputs to a probability distribution over all possible class labels for each word in the sequence.

### 3.4. Large language modelling head

We incorporate the per-word goal and detail intent predictions from our speech understanding models into a Large Language Model (LLM) to determine appropriate task plans a robot may follow to satisfy the given instruction. Specifically, we feed the transcribed text utterance along with the predicted intents and in-context examples into an LLM prompt, enabling the model to infer the most appropriate interpretation and subsequent task plan. We leverage the prompting structure and Composable Referent Descriptor (CRD) syntax proposed by [15]. To evaluate the effectiveness of different LLMs in this context, we experiment with three high-performing language models from OpenAI: GPT-4o<sup>4</sup>, o1-mini<sup>5</sup>, and o3-mini<sup>6</sup>.

## 4. Experimental setup

### 4.1. Data

To evaluate our approach, we collected a novel accented speech dataset of **35 ambiguous instructions**, inspired by [15]. Each instruction in this dataset can be interpreted in two distinct ways, resulting in **two separate sets of intent referents** per instruction. Notably, the goal and detail referents vary between interpretations, introducing structured ambiguity that our models aim to resolve. To create the speech dataset, we recruited **22 participants** (10 males, 12 females) to record all 35 instructions, providing both possible interpretations. Each participant provided **44 samples** recordings resulting in **1,540 voice samples** with a total duration of approximately **121 minutes**. The recordings were collected using a Google Pixel 8 smartphone. The training set consisted of 1,056 samples, whereas the validation and test sets contained 264 and 220 samples respectively. To encourage generalization, we designed the dataset split such that the instructions present in the evaluation sets were not included in the training data. In our experiments, we conduct feature ablation studies to examine the impact of different input representations on intent detection performance

<sup>4</sup><https://openai.com/index/hello-gpt-4o/>

<sup>5</sup><https://openai.com/o1/>

<sup>6</sup><https://openai.com/index/openai-o3-mini/>

Table 1: Performance in Classifying Goal and Detail Referent Intents for Transformer and BiLSTM Models.

Model	Intent	Prosody				Raw				Prosody+Raw			
		Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
Transformer	Goal	<b>92.61%</b>	<b>85.41%</b>	<b>75.60%</b>	<b>80.20%</b>	90.81%	76.79%	76.79%	76.79%	90.96%	84.50%	66.51%	74.43%
	Detail	<b>94.70%</b>	<b>83.85%</b>	<b>81.82%</b>	<b>82.82%</b>	91.10%	80.08%	57.27%	66.78%	89.11%	70.16%	52.73%	60.21%
	Overall	93.31%	<b>90.04%</b>	<b>90.25%</b>	<b>90.06%</b>	<b>93.92%</b>	88.13%	88.73%	88.07%	88.59%	83.22%	83.90%	82.95%
BiLSTM	Goal	<b>93.04%</b>	<b>97.21%</b>	66.75%	<b>79.15%</b>	91.48%	84.00%	<b>70.33%</b>	76.56%	91.90%	84.94%	67.46%	75.20%
	Detail	<b>94.22%</b>	<b>76.13%</b>	<b>91.82%</b>	<b>83.24%</b>	91.43%	73.95%	69.70%	71.76%	91.81%	72.89%	75.76%	74.29%
	Overall	<b>95.79%</b>	<b>92.87%</b>	<b>92.14%</b>	<b>91.80%</b>	94.31%	88.89%	89.39%	88.95%	94.74%	89.50%	89.77%	89.39%

Table 2: LLM Model Performance in Choosing the Correct Task Plans.

	LLM+ASR	LLM+Prosody-	LLM+Prosody-
		BiLSTM	Transformer
Gpt-4o	50%	<b>56.84%</b>	<b>71.96%</b>
o1-mini	50%	52.10%	62.5%
o3-mini	50%	54.54%	61.83%

beyond prosodic features alone. In one setting, we extract only raw audio features using the Librosa library<sup>7</sup>. In another, we integrate both prosodic and raw audio features as input to our models. This experimental setup allows us to systematically evaluate whether prosodic features alone can effectively enhance the detection of goal and detail intents or if a combination of prosodic and raw features leads to improved performance.

#### 4.2. Training procedure

We train our models by minimizing the *cross-entropy loss* between the predicted label sequences,  $\hat{Y}^{(i)}$ , and the corresponding ground truth,  $Y^{(i)}$ . To identify the optimal hyperparameter configurations, we conduct a *grid search*, using the Optuna hyperparameter optimization framework<sup>8</sup>, exploring a range of values for the learning rate, hidden dimension, number of layers, dropout rate, attention-layer depth, and weight decay. The values that maximize the *F1-score* on the validation set are selected. The *optimal learning rate* for the Transformer-based model is found to be  $2.22 \times 10^{-4}$ , while the BiLSTM-based model performs best with a learning rate of  $4.2 \times 10^{-3}$ . The *hidden dimension* of the linear layers is optimized at 448 for the Transformer-based model and 512 for the BiLSTM-based model. Regarding model depth, the *best-performing Transformer model* consists of *three layers*, whereas the *optimal BiLSTM model* is a *single-layer architecture*. For *regularization*, the dropout rate is optimized at 0.25 for the Transformer model and 0.45 for the BiLSTM model. The Transformer model’s attention mechanism benefits from a deeper configuration, consisting of *eight attention layers*, whereas the BiLSTM model operates effectively with a *single attention layer*. Finally, weight decay, which plays a crucial role in controlling overfitting, is optimized at  $2.25 \times 10^{-6}$  for the Transformer-based model and  $5.44 \times 10^{-5}$  for the BiLSTM-based model. The total number of parameters for our BiLSTM-based model is *18,153,476* while the total number of parameters for our Transformer-based model is *19,003,014*. During training, we optimize model parameters using the Adam [21] optimizer with weight decay.

<sup>7</sup><https://github.com/librosa/librosa>

<sup>8</sup><https://optuna.org>

A *learning-rate scheduler* is employed to dynamically adjust the step size if progress stagnates. It takes on average 8 hours to perform a hyperparameter search and train our models.

## 5. Results

### 5.1. Referent detection task performance

Our experimental results indicate that utilizing **prosodic features** alone consistently yields the highest performance in goal and detail referent detection tasks. These findings suggest that while raw acoustic features are informative, **prosodic signals exhibit superior discriminative power**, effectively highlighting key acoustic patterns associated with instruction intent that may otherwise be obscured within raw audio signals. Architecturally, the **BiLSTM model slightly outperforms the Transformer model** in referent detection, which aligns with expectations, as Transformer models are known to benefit more significantly from larger training datasets [22].

### 5.2. Intent Disambiguation task performance

For the intent disambiguation task, which involves selecting the appropriate task plan given a speech utterance, our experimental results, presented in Table 2, demonstrate that augmenting LLM input prompts with goal and detail intent referents detected by our proposed prosody-based models significantly enhances performance. Notably, the best performance on this task is achieved by the combination of our Prosody-Transformer model and GPT-4o. We hypothesize that GPT-4o outperforms the other LLMs due to its likely larger model size and richer internal representations, which enable superior generalization across diverse tasks.

## 6. Limitations and conclusion

Our work is limited by a relatively small dataset (1540 samples) and the narrow age range (18–22) of participants, potentially restricting the generalizability of our findings. Our future efforts will focus expanding the dataset size and participant diversity to enhance model robustness and applicability across varied speaker populations. In this study, we have demonstrated the pivotal role of prosodic information in accurately interpreting speech instructions for human-robot interaction. By incorporating prosodic cues, robots can more effectively infer intent, thereby improving the naturalness of human-robot communication. We also introduce speech disambiguation as a key research area in robotics and present a first-of-its-kind dataset to catalyze further advances in this field.

## 7. References

- [1] D. T. Tran, D. H. Truong, H. S. Le, and J.-H. Huh, "Mobile robot: automatic speech recognition application for automation and stem education," *Soft Comput.*, vol. 27, no. 15, p. 10789–10805, Feb. 2023. [Online]. Available: <https://doi.org/10.1007/s00500-023-07824-7>
- [2] A. Davila, J. Colan, and Y. Hasegawa, "Voice control interface for surgical robot assistants," 2024. [Online]. Available: <https://arxiv.org/abs/2409.10225>
- [3] B. N. Rout, A. Das, and S. R. Sankar, "Speech recognition system for robotic control and movement," *International Journal of Scientific & Technology Research*, vol. 4, pp. 221–225, 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:60663713>
- [4] I. McLoughlin and N. Indurkha, *AI, Human–Robot Interaction, and Natural Language Processing*, ser. Cambridge Handbooks in Language and Linguistics. Cambridge University Press, 2023, p. 436–454.
- [5] R. Liu, Y. Guo, R. Jin, and X. Zhang, "A review of natural-language-instructed robot execution systems," *AI*, vol. 5, no. 3, pp. 948–989, 2024. [Online]. Available: <https://www.mdpi.com/2673-2688/5/3/48>
- [6] E. Nöth, A. Batliner, V. Warnke, J. Haas, M. Boros, J. Buckow, R. Huber, F. Gallwitz, M. Nutt, and H. Niemann, "On the use of prosody in automatic dialogue understanding," *Speech Communication*, vol. 36, no. 1–2, pp. 45–62, 2002.
- [7] E. Shriberg and A. Stolcke, "Prosody modeling for automatic speech recognition and understanding," in *Mathematical Foundations of Speech and Language Processing*. Springer, 2004, pp. 105–114.
- [8] S. Rajaa, "Improving end-to-end slu performance with prosodic attention and distillation," *arXiv preprint arXiv:2305.08067*, 2023.
- [9] K. Wei, D. Knox, M. Radfar, T. Tran, M. Müller, G. P. Strimel, N. Susanj, A. Mouchtaris, and M. Omologo, "A neural prosody encoder for end-to-end dialogue act classification," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7047–7051.
- [10] M. MacMahon, B. Stankiewicz, and B. Kuipers, "Walk the talk: Connecting language, knowledge, and action in route instructions," *Def.*, vol. 2, no. 6, p. 4, 2006.
- [11] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 25, no. 1, 2011, pp. 1507–1514.
- [12] T. Kollar, S. Tellex, D. Roy, and N. Roy, "Toward understanding natural language directions," in *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2010, pp. 259–266.
- [13] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Motlaghi, L. Zettlemoyer, and D. Fox, "Alfred: A benchmark for interpreting grounded instructions for everyday tasks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 740–10 749.
- [14] A. Jaafar, S. S. Raman, Y. Wei, S. Juliani, A. Wernerfelt, B. Quartey, I. Idrees, J. X. Liu, and S. Tellex, "Lanmp: A language-conditioned mobile manipulation benchmark for autonomous robots," *arXiv preprint arXiv:2412.05313*, 2024.
- [15] B. Quartey, E. Rosen, S. Tellex, and G. Konidaris, "Verifiably following complex robot instructions with foundation models," *arXiv preprint arXiv:2402.11498*, 2024.
- [16] D. Paulius, A. Agostini, B. Quartey, and G. Konidaris, "Bootstrapping object-level planning with large language models," *arXiv preprint arXiv:2409.12262*, 2024.
- [17] J. X. Liu, A. Shah, G. Konidaris, S. Tellex, and D. Paulius, "Lang2tl-2: Grounding spatiotemporal navigation commands using large language and vision-language models," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 2325–2332.
- [18] S. Zhang, D. Zheng, X. Hu, and M. Yang, "Bidirectional long short-term memory networks for relation classification," in *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, H. Zhao, Ed., Shanghai, China, Oct. 2015, pp. 73–78. [Online]. Available: <https://aclanthology.org/Y15-1009/>
- [19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf)
- [20] P.-H. Li, T.-J. Fu, and W.-Y. Ma, "Why attention? analyzing and remedying bilstm deficiency in modeling cross-context for ner," *ArXiv*, vol. abs/1910.02586, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:203838097>
- [21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: <https://api.semanticscholar.org/CorpusID:6628106>
- [22] X. Niu, B. Bai, L. Deng, and W. Han, "Beyond scaling laws: Understanding transformer performance with associative memory," *arXiv preprint arXiv:2405.08707*, 2024.