Define, Evaluate, and Improve Task-Oriented Cognitive Capabilities for Instruction Generation Models

♠Lingjun Zhao* and ♣Khanh Nguyen* and ♠♢Hal Daumé III
♠University of Maryland–College Park ♣Princeton University ♢Microsoft Research
lzhao123@umd.edu

Abstract

Recent work studies the cognitive capabilities of language models through psychological tests designed for humans. While these studies are helpful for understanding the general capabilities of these models, there is no guarantee that a model possessing sufficient capabilities to pass those tests would actually *use* those capabilities in performing real-life tasks. In this work, we formulate *task-oriented* cognitive capabilities, which are human-like cognitive capabilities that language models leverage to perform tasks. These capabilities are (i) the ability to quickly generate good candidate utterances (the search capability) (ii) the ability to predict how a listener interprets those utterances and choose the most appropriate one (the pragmatic capability). We design an evaluation scheme for comparing these capabilities of a language model with those of a human. Applying this scheme to examine various models in a navigation instruction generation problem, we find that their pragmatic capability is severely lacking. This insight leads us to augment them with better models of the listener and obtain a significant boost of 11% in success rate in guiding real humans. Our work advocates for having a principled procedure for aligning language models with humans that involves (i) formulating task-oriented capabilities, (ii) devising a method to quantify their deficiency, and (iii) iteratively improving them.

1 Introduction

To communicate successfully with humans, language models must possess cognitive capabilities similar to those that facilitate human communication. Examining the cognitive capabilities of language models is notoriously challenging because the operations of these models are largely unintelligible to humans. Psychologists faced similar challenges when investigating human cognition, and have devised various behavioral tests to diagnose human cognitive capabilities (Premack and Woodruff, 1978; Wimmer and Perner, 1983; Baron-Cohen et al., 1985; Gopnik and Astington, 1988). Recent work (Sap et al., 2022; Kosinski, 2023; Ullman, 2023) applies these tests to evaluate large language models by inputting the tests to these models as prompts and verifying whether they behave like a normal human would.

While this approach is helpful for understanding the general limitations of language models, it has two potential drawbacks. First, it is applicable to only large language models that can comprehend human-written prompts, entangling linguistic capability with reasoning capability. Second, it shows that a language model can or cannot demonstrate certain mental skills, but does not imply that the model would employ those skills to perform a downstream task. For example, passing false-belief tests does not guarantee that a model will reason about the interpretation of the readers when generating summaries. In general, scoring high on psychological tests may not be sufficient to ensure language models would behave like humans in real-life scenarios.

In this work, we take a different approach to evaluating the cognitive capabilities of language models. We define and evaluate *task-oriented* cognitive capabilities, which are human-like capabilities that a model actually employs to perform the task it is designed for. Enhancing these capabilities thus warrants improved performance on the task. To identify these capabilities, we build on two lines of work from socio-cognitive science: Bayesian models of cooperative communication (Wang et al., 2020; Goodman and Frank, 2016; Shafto et al., 2014) and studies on how humans implement Bayesian reasoning (Sanborn and Chater, 2016; Sanborn et al., 2010; Vul et al., 2014; Mamassian et al., 2002). We propose a mathematical cognitive model called *bounded pragmatic speaker*, which can reasonably characterize the rea-

*The first two authors contribute equally.

Findings of the Association for Computational Linguistics: ACL 2023, pages 3688–3706
July 9-14, 2023 ©2023 Association for Computational Linguistics

soning processes of both humans and language models. Casting humans and language models in the same way enables us to juxtapose their cognitive capabilities. We mathematically formulate two capabilities that a bounded pragmatic agent must possess in order to generate optimally pragmatic utterances. These conditions correspond to well-known cognitive capabilities of humans: (i) the ability to efficiently generate relevant utterances (the *search* capability) (Bloom and Fischler, 1980; Gold et al., 2000; Trosborg, 2010) and (ii) the ability to accurately simulate the listener's interpretations of their utterances (the *pragmatic* capability) (Premack and Woodruff, 1978; Gopnik and Astington, 1988; Tomasello, 2019; Call and Tomasello, 2011; Frank and Goodman, 2012). We design a simple procedure to quantitatively evaluate these capabilities of a language model. To evaluate each capability, we compute the task performance gap between the model and an *oracle* model, which is identical except that the evaluated capability of this model is at human level. Figure 1 illustrates our procedure, which theoretically can be applied to any language model.

We evaluate various language models on a navigation instruction generation problem (Anderson et al., 2018b), where a model generates English instructions to guide real humans in photo-realistic 3D environments.[1] Our evaluation reveals an interesting finding: all evaluated agents possess relatively efficient search capability but inadequate pragmatic capability. We improve the pragmatic capability of the evaluated models by enabling them to reason probabilistically about human listeners (Andreas and Klein, 2016; Fried et al., 2018a), employing state-of-the-art instruction-following agents (Magalhaes et al., 2019; Shen et al., 2022; Hong et al., 2021) as models of human listeners. We obtain significant improvement in success rate over the original agents, shrinking the gap with human performance on held-out data by 36%. Towards eliminating the remaining gap, we illustrate with empirical evidence a major challenge in developing better listener models. Specifically, when the instruction-following agents are employed as listener models for the instruction-generating agent, they are required to evaluate *AI-generated* instructions, which may be significantly different from human-generated

[1]Our human-evaluation dataset and interface are publicly released at `https://lingjunzhao.github.io/coop_instruction.html`.

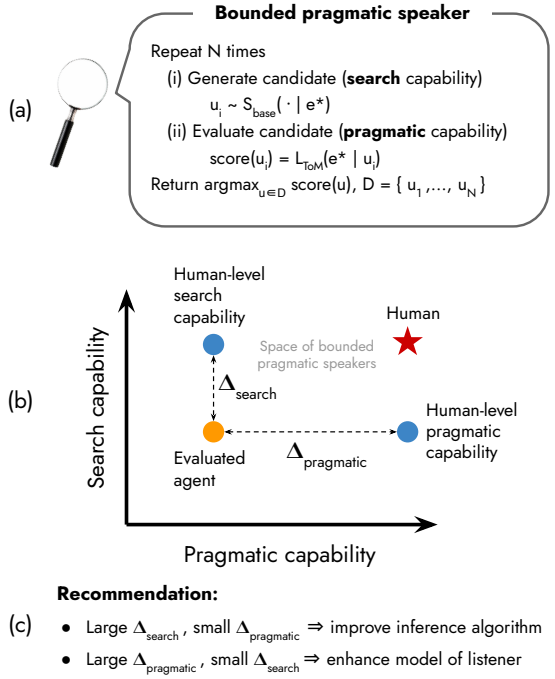

Figure 1: We propose a framework called *bounded pragmatic speaker* which can characterize pragmatic reasoning in both humans and language models (a). A bounded pragmatic speaker is composed of a base speaker S_{base}, representing prior knowledge that helps generate instructions efficiently, and a theory-of-mind (ToM) listener L_{ToM}, a hypothetical model of how the real listener interprets instructions. Viewing language models and humans through this unifying lens enables comparing their cognitive capabilities (b). To evaluate a capability of a model, we compare it with an oracle model which is identical except that the evaluated capability is at human level. The outcome of our evaluation can better inform the future direction for improving the model (c).

instructions. Hence, a standard supervised-learning training scheme that only exposes these models to human-generated instructions would be inadequate for learning reliable models. We thus call for construction of novel datasets, algorithms, and evaluation methods for developing the pragmatic capability of language models.

2 Related Work

Navigation Instruction Generation. Instruction generation has been commonly studied in navigation settings (Anderson et al., 1991; Byron et al., 2009; Koller et al., 2010; Striegnitz et al., 2011; Goeddel and Olson, 2012; Fried et al., 2018a,b). The Matterport3D simulator and the accompanying datasets (R2R (Anderson et al., 2018b), R4R (Jain et al., 2019), and RxR (Ku et al., 2020)) offer more challenging settings by combining photo-realistic

scenes with long, verbally rich instructions. Recent work on evaluating instruction generation agents (Zhao et al., 2021) reveals the ineffectiveness of standard learning and modeling approaches to this problem. Wang et al. (2021) improve the accuracy and interpretability of instructions in the RxR setting. Kamath et al. (2023) leverage this model to synthesize additional data for training instruction-following agents. Our work aim to offer useful principles to further improve these models.

Mathematical Models of Human Communication. Different from communication within agents (Lazaridou et al., 2020; Roman Roman et al., 2020), human communication is a cooperative act (Grice, 1975; Scott-Phillips, 2014; Tomasello, 2019). Pragmatic communication in humans may involve different cognitive capabilities like basic understanding of language and social rules (Trosborg, 2010) and reasoning about the physical world (Bender and Koller, 2020) and human behavior (Enrici et al., 2019; Rubio-Fernandez, 2021). Our work describes similar capabilities but provides a formal mathematical description. Development of mathematical models of human communication have been greatly useful for understanding human behaviors (Ho et al., 2016; Sumers et al., 2022) and building communication agents (Andreas and Klein, 2016; Fried et al., 2018a,b; FAIR, 2022; Lin et al., 2022; Zhu et al., 2021; Bao et al., 2022). Wang et al. (2020) unify these models under an optimal-transport framework. The model we propose in this work is a generalized version capturing the essence of these models.

Evaluating Cognitive Capabilities of Neural Networks. Many benchmarks for evaluating the cognitive capabilities of AI-based agents have been created, focusing on theory-of-mind capabilities (Le et al., 2019; Nematzadeh et al., 2018), grounding (Lachmy et al., 2022; Udagawa and Aizawa, 2019; Haber et al., 2019), or commonsense reasoning (Talmor et al., 2019; Levesque et al., 2012; Zellers et al., 2019; Sap et al., 2019). Large language models have demonstrated exceptional performance on following human instructions and solving complex reasoning tasks (Bubeck et al., 2023; Anil et al., 2023), raising the question of whether their cognitive capabilities are similar or as advanced as those of humans. Mahowald et al. (2023) advocate for separating formal competence (knowledge about linguistic rules and patterns) from their functional competence (knowledge about the world usage in the world) when assessing these models. Our bounded pragmatic speaker framework mathematically formalizes this description, allowing for quantitative evaluation of these competencies. Recent work (Sap et al., 2022; Kosinski, 2023; Ullman, 2023; Hu et al., 2023) examines cognitive capabilities of large language models through tests inspired by human psychological tests. The goal of these studies is to determine the limits of large language models, potentially calibrating the expectation on them. On the other hand, our focus is to devise a method that can be applied to language models of any size and benchmark cognitive capabilities that are relevant for accomplishing a specific task.

3 Problem Setting

We are concerned with instruction generation: learning a speaker agent r that generates language instructions to guide a human listener h to reach states in an environment.

Human Listener. We imagine a human listener h acting in a partially observed environment with states s. The human does not have access to states but only observations o^h and takes actions a^h. An *instruction* $\boldsymbol{u} \in \mathcal{U}$ is a language utterance consisting of words. A *trajectory* $\boldsymbol{e} = (s_1, o_1, a_1, \cdots, s_T, o_T, a_T)$ is an execution of an instruction. The human can follow instructions to generate trajectories in the environment. For example, in an indoor navigation setting, upon hearing "*go the kitchen and stop next to the oven*", a human walks to the specified location. We define $L_h(\boldsymbol{e} \mid \boldsymbol{u})$ as the probability that the human generates $\boldsymbol{e}$ upon hearing $\boldsymbol{u}$.

Speaker Agent. In each task, the speaker agent first imagines an *intended trajectory* $\boldsymbol{e}^\star = (s_1, o_1^r, a_1^r, \cdots, s_T, o_T^r, a_T^r)$, which specifies a path to get to an intended goal state s_T from the human's current state s_1. Because the human's actions and perception may differ from those of the speaker, they may not be able to comprehend $\boldsymbol{e}^\star$ even if it is presented to them. Thus, the speaker needs to translate the trajectory into an instruction $\hat{\boldsymbol{u}}$ that the human can understand and follow. To do so, it implements a *language model* $S_r(\boldsymbol{u} \mid \boldsymbol{e})$, and an *inference algorithm* $\text{Gen}(S_r, \boldsymbol{e})$ to craft instructions based on the model (e.g., greedy or beam-search decoding). The speaker's objective is to generate instructions that maximize the expected chance of

the listener reconstructing the intended trajectories

$$\arg\max_{S_r} \mathbb{E}_{\boldsymbol{e}^\star} \left[L_h(\boldsymbol{e}^\star \mid \text{Gen}(S_r, \boldsymbol{e}^\star)) \right] \quad (1)$$

Evaluation. The speaker is evaluated using a dataset $\mathcal{D}_{\text{eval}}$ of held-out trajectories. For each trajectory $\boldsymbol{e}_k^\star \in \mathcal{D}_{\text{eval}}$, we generate an instruction $\hat{\boldsymbol{u}}_k = \textsc{Gen}(S_r, \boldsymbol{e}_k^\star)$. The instruction is then presented to a (real) human listener to follow, producing a trajectory $\boldsymbol{e}_k^{\text{h}} \sim L_h(\cdot \mid \hat{\boldsymbol{u}}_k)$. The performance of the speaker, denoted by $\rho(r)$, is the average similarity, Ψ, between the human-generated and the intended trajectories:

$$\rho(r) \triangleq \frac{1}{|\mathcal{D}_{\text{eval}}|} \sum_{\boldsymbol{e}_k^\star \in \mathcal{D}_{\text{eval}}} \Psi(\boldsymbol{e}_k^{\text{h}}, \boldsymbol{e}_k^\star) \quad (2)$$

We will specify the choices for the metric Ψ in the experimental setup section (§6).

4 Task-Oriented Cognitive Capabilities

Human have evolved highly effective cognitive capabilities for communication. How can we endow a speaker agent with similar capabilities and quantify the degree of resemblance between its capabilities and those of a human?

We propose a mathematical framework that reasonably characterizes the human cognitive process for instruction generation (§4.1). We show that this model can also describe the operation of language models, which allows us to compare them with humans on specific cognitive capabilities. We identify two capabilities that are requisite for any agent implementing our framework to generate optimal instructions (§ 4.2), and introduce an evaluation scheme for collating these capabilities (§4.3).

4.1 A Mathematical Cognitive Model of Instruction Generation

To formulate how humans generate instructions, we build on mathematical models of cooperative communication (Wang et al., 2020; Goodman and Frank, 2016; Shafto et al., 2014). We consider a general version where a speaker agent constructs a *pragmatic speaker* model $S_{\text{prag}}(\boldsymbol{u} \mid \boldsymbol{e})$ based on two constituents: a *base speaker* model $S_{\text{base}}(\boldsymbol{u} \mid \boldsymbol{e})$ and a *theory-of-mind (ToM) listener* model $L_{\text{ToM}}(\boldsymbol{e} \mid \boldsymbol{u})$. The base speaker represents general knowledge of the agent about the world and the language it speaks. The ToM listener reflects situated knowledge about the listener, simulating how they would behave in the environment given an instruction. A pragmatic speaker aims to maximize the chance of the listener interpreting its instruction correctly, but it is still influenced by its general knowledge (e.g., social biases, language style). Formally, it is defined as:

$$S_{\text{prag}}(\boldsymbol{u} \mid \boldsymbol{e}) \propto L_{\text{ToM}}(\boldsymbol{e} \mid \boldsymbol{u}) S_{\text{base}}(\boldsymbol{u} \mid \boldsymbol{e}) \quad (3)$$

To convey an intended trajectory $\boldsymbol{e}^\star$, this speaker utters an instruction of maximum probability under its model:

$$\begin{aligned} \hat{\boldsymbol{u}}_{\text{prag}} &\triangleq \arg\max_{\boldsymbol{u} \in \mathcal{U}} S_{\text{prag}}(\boldsymbol{u} \mid \boldsymbol{e}^\star) \\ &= \arg\max_{\boldsymbol{u} \in \mathcal{U}} L_{\text{ToM}}(\boldsymbol{e}^\star \mid \boldsymbol{u}) S_{\text{base}}(\boldsymbol{u} \mid \boldsymbol{e}^\star) \quad (4) \end{aligned}$$

Humans as bounded pragmatic speakers. The pragmatic speaker model accounts for human behaviors highly accurately on problems where $\mathcal{U}$ is a small discrete space (Frank and Goodman, 2012). However, in problems like instruction generation where $\mathcal{U}$ is an unbounded set of linguistic expressions, it is unlikely that humans, which are known to be agents with bounded rationality (Simon, 1957), are able to compute the optimal utterance in Eq 4 exactly. A hypothesis, supported by empirical evidence, is that humans perform approximate inference via Monte-Carlo sampling (Sanborn and Chater, 2016; Sanborn et al., 2010; Vul et al., 2014; Mamassian et al., 2002). Applying this hypothesis to our setting, we derive a more practical model of how human generate instructions, in which they perform the search for the best utterance on only a subspace $\mathcal{U}_{\text{sub}}$ of $\mathcal{U}$ defined by a set of candidates sampled from S_{base}

$$\hat{\boldsymbol{u}}_{\text{bounded-prag}} \triangleq \arg\max_{\boldsymbol{u} \in \mathcal{U}_{\text{sub}} \subset \mathcal{U}} L_{\text{ToM}}(\boldsymbol{e}^\star \mid \boldsymbol{u}) \quad (5)$$

where $\mathcal{U}_{\text{sub}} = \{\boldsymbol{u}_i \sim S_{\text{base}}(\cdot \mid \boldsymbol{e}^\star) \mid 1 \leq i \leq N\}$. We call an agent that generates instructions according to Eq 5 a *bounded pragmatic speaker* (Figure 2). For such a speaker, instruction generation involves two tasks: candidate generation (performed by S_{base}) and candidate evaluation (performed by L_{ToM}). The former task ensures that the generation of an instruction is efficient, while the latter guarantees the generated instruction conveys the intended meaning to the human listener.

4.2 Formulating Task-Oriented Cognitive Capabilities

What cognitive capabilities enable humans to generate effective instructions? Viewing humans as bounded pragmatic speakers allows us to mathematically characterize those capabilities. Specifi-

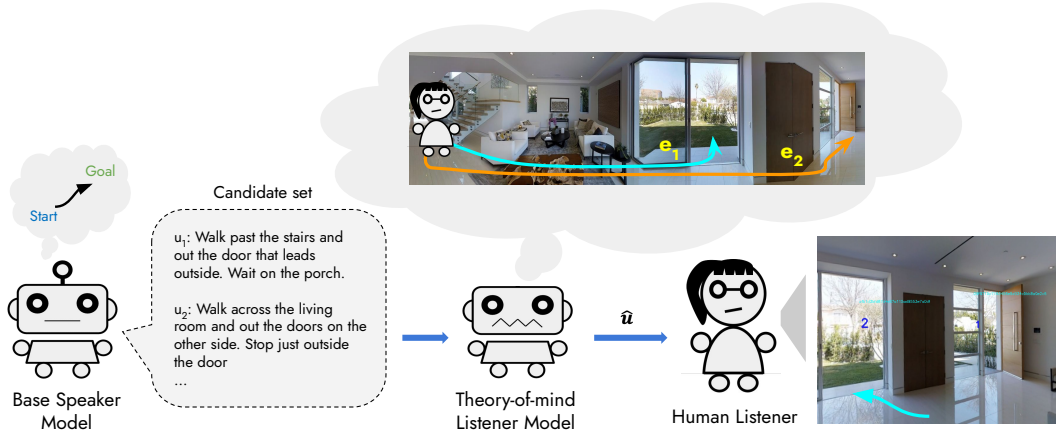

Figure 2: The cognitive process of a bounded pragmatic speaker. In every task, the speaker first imagines a trajectory it wants to convey to the human listener. To reduce the search space, it then uses the *base speaker* to generate a small set of relevant candidate instructions. After that, it employs the *theory-of-mind listener* to simulate how the human listener would follow each instruction in the candidate set. The speaker finally elects the candidate instruction that causes the theory-of-mind listener to generate the trajectory most similar to the intended trajectory. The output instruction is finally sent to the human listener for a real execution in the environment.

cally, we require a bounded pragmatic speaker to be able to output the optimal utterance, i.e. satisfying

$$\hat{\boldsymbol{u}}_{\text{bounded-prag}} = \boldsymbol{u}^\star \triangleq \arg\max_{\boldsymbol{u}} L_h(\boldsymbol{e}^\star \mid \boldsymbol{u}) \quad (6)$$

where L_h is the human listener.

For this equation to hold, the constituent models S_{base} and L_{ToM} of the bounded pragmatic speaker must meet certain conditions. The condition for S_{base} is that the candidate set it generates must contain the optimal instruction, i.e. $\boldsymbol{u}^\star \in \mathcal{U}_{\text{sub}}$. This condition requires S_{base} to be capable of quickly generating candidates and placing high probability on $\boldsymbol{u}^\star$ so the instruction can be found by sampling a few candidates. We refer to this capability as the *search capability*.

Meanwhile, the condition for L_{ToM} is that it has to rank $\boldsymbol{u}^\star$ first among the candidates in $\mathcal{U}_{\text{sub}}$. Meeting this condition demands having the capability of constructing a mental emulation of the human listener and simulating the actions of the listener after receiving an instruction. We refer to this capability as the *pragmatic capability*.

The search and pragmatic capabilities are orthogonal and complementary. An agent with flawless pragmatic capability can evaluate the goodness of instructions given to it, but may not be able to efficiently generate good instructions by itself. In contrast, an agent with effective search capability can quickly bring to attention highly relevant utterances but cannot select the best one to output if its ToM model is erroneous.

4.3 Evaluating Task-Oriented Cognitive Capabilities

We have defined two cognitive capabilities that are requisite for humans in instruction generation. In this section, we will prove that a language model can also be cast as a bounded pragmatic speaker. Hence, we can compare it with a human on the two cognitive capabilities.

Language models as bounded pragmatic speakers. We consider a speaker agent r that learns a language model $S_r(\boldsymbol{u} \mid \boldsymbol{e})$ and runs an inference algorithm to compute an output $\hat{\boldsymbol{u}}_{\text{infer}} = \text{GEN}(S_r, \boldsymbol{e}^\star) \approx \arg\max_{\boldsymbol{u}\in\mathcal{U}} S_r(\boldsymbol{u} \mid \boldsymbol{e}^\star)$. Generative LSTM- or Transformer-based models that implement greedy or beam-search decoding are examples of this agent. We make the following assumption about the inference algorithm.[2]

Assumption (Better-than-sampling inference algorithm)**.** *We assume the inference algorithm is better at finding* $\arg\max_{\boldsymbol{u}\in\mathcal{U}} S_r(\boldsymbol{u} \mid \boldsymbol{e}^\star)$ *than drawing a small number of* N *samples from* S_r*. Formally, let* γ *be the probability of drawing* $\boldsymbol{e}^\star$ *and a set of* N *instructions from* S_r *such that* $S_r(\hat{\boldsymbol{u}}_{infer} \mid \boldsymbol{e}^\star) > \max_{\boldsymbol{u}\in\mathcal{U}_{sub}} S_r(\boldsymbol{u} \mid \boldsymbol{e}^\star)$*, where* $\hat{\boldsymbol{u}}_{infer} = \text{GEN}(S_r, \boldsymbol{e}^\star)$*. We assume that* γ *is large for a small integer* $N > 0$*.*

[2]We empirically verify that this assumption holds for the agents we evaluate with $N = 10$ and γ ranging from 0.7 to 0.9. We estimate γ by computing the fraction of evaluation examples where the agent's model ranks $\hat{\boldsymbol{u}}_{\text{infer}}$ above N samples drawn from it.

If this assumption holds, then with high probability, the agent r behaves identically to a bounded pragmatic speaker that computes its output as:

$$\hat{\boldsymbol{u}} \triangleq \arg\max_{\boldsymbol{u} \in \mathcal{U}^r_{\text{sub}}} S_r(\boldsymbol{u} \mid \boldsymbol{e}^\star) \quad (7)$$

$$\mathcal{U}^r_{\text{sub}} \triangleq \{\hat{\boldsymbol{u}}_{\text{infer}}\} \cup \{\boldsymbol{u}_i \sim S_r \mid 1 \leq i \leq N\} \quad (8)$$

This agent uses S_r as both the base speaker S_{base} and ToM listener L_{ToM}. Due to our assumption, on most inputs, the agent outputs $\hat{\boldsymbol{u}}_{\text{infer}}$, similar to the original agent. We employ this bounded pragmatic speaker as the proxy for the original agent in comparisons with humans, and also refer to it as r.

Evaluation scheme. To evaluate a cognitive capability (search or pragmatic) of a speaker r, we compute the performance gap between it and an oracle agent that is at human level on the evaluated capability, but is equally good as it is at the other capability. Specifically, we define $r^\star_{\text{search}}$ to be an oracle speaker that employs S_r as the ToM model but is given a "gold" candidate set $\mathcal{U}^\star_{\text{cand}}$ that always contains a human-generated reference instruction $\boldsymbol{u}^\star$. It selects its output as follows

$$\boldsymbol{u}^\star_{\text{search}} \triangleq \arg\max_{\boldsymbol{u} \in \mathcal{U}^\star_{\text{cand}}} S_r(\boldsymbol{u} \mid \boldsymbol{e}^\star) \quad (9)$$

This agent has similar pragmatic capability as r but human-level search capability. Next, we construct $r^\star_{\text{pragmatic}}$, an oracle that generates candidates using S_r but employs a human L_h to rank the candidates

$$\boldsymbol{u}^\star_{\text{pragmatic}} \triangleq \arg\max_{\boldsymbol{u} \in \mathcal{U}^r_{\text{sub}}} L_h(\boldsymbol{e}^\star \mid \boldsymbol{u}) \quad (10)$$

with $\mathcal{U}^r_{\text{sub}}$ from Eq 8. The search capability of $r^\star_{\text{pragmatic}}$ is as good as r but its pragmatic capability is that of a human.

We calculate the *prospective performance gain* (PPG) with respect to each capability as follows

$$\text{PPG}_{\text{search}}(r) \triangleq \rho(r^\star_{\text{search}}) - \rho(r) \quad (11)$$

$$\text{PPG}_{\text{pragmatic}}(r) \triangleq \rho(r^\star_{\text{pragmatic}}) - \rho(r) \quad (12)$$

where ρ is the performance on held-out data (Eq 2 of § 3). Each metric computes the potential improvement if the corresponding capability is upgraded to match with that of a human. Comparing the two metrics reveals which of the two capabilities of r is currently more deficient and informs future development direction for the agent. For example, if $\text{PPG}_{\text{search}}(r)$ is large and $\text{PPG}_{\text{pragmatic}}(r)$ is small, it means that r is scoring the candidate instructions highly accurately but it is bad at finding high-score instructions. In this case, developers may want to focus on devising a more effective inference algorithm to improve the search capability of r. On the other hand, if r estimates poorly calibrated scores, signified by $\text{PPG}_{\text{pragmatic}}(r)$ being large, enhancing its inference algorithm is fruitless, but endowing it with a module that simulates the listener's behavior more accurately would boost its performance.

5 Improving Pragmatic Capability with Ensemble Instruction-Following Agents

In cases where our evaluation scheme indicates that the pragmatic capability of a language model is deficient, we improve it by installing a better ToM listener model. A common approach to learning this listener model is to use the same dataset used for learning the speaker model (Andreas and Klein, 2016; Fried et al., 2018a,b). We argue that this approach has a potential drawback. A ToM model learned in this way is only exposed to human-generated input instructions. At deployment time, it would likely experience a *covariate shift* because as a ToM model, the model is then asked to score instructions generated by a speaker model, not by humans. These instructions may be incorrect, ungrammatical, or may simply have a different style than human-generated instructions. This covariate shift would hamper the model's judgement. Our preliminary experiments (Appendix § A.6) confirm that using a listener trained on only human-generated inputs as the ToM model hurts rather than improves the performance of various speakers.

We show that this problem can be alleviated by employing ToM models that have calibrated uncertainty on unseen instructions. We obtain calibrated models through ensembling (Lakshminarayanan et al., 2017): we train listener models $\hat{L}^{(k)}(\boldsymbol{e} \mid \boldsymbol{u})$, $k = 1 \ldots K$, each on a random 90% subset of the training data with different random initial seeds.

We also leverage access to a simulation of the environment to construct better ToM models. Note that the probability that a ToM model L_{ToM} assigns to an instruction can be seen as an expectation of a binary metric: $L_{\text{ToM}}(\boldsymbol{e}^\star \mid \boldsymbol{u}) = \mathbb{E}_{\boldsymbol{e} \sim L_{\text{ToM}}(\cdot \mid \boldsymbol{u})}[\mathbb{1}\{\boldsymbol{e} = \boldsymbol{e}^\star\}]$, which does not award credit if $\boldsymbol{e}$ overlaps only partially with $\boldsymbol{e}^\star$. We propose two augmentations: (i) replace the binary metric with a soft metric $\Psi(\boldsymbol{e}, \boldsymbol{e}^\star)$ that can measure partial similarity between trajectories and (ii) approximate the expectation by executing listeners $\hat{L}^{(k)}$ in the simulated environment to sample trajec-

tories. Our final model selects its instruction as:

$$\hat{\boldsymbol{u}}_{\text{augment-ToM}} \triangleq \arg\max_{\boldsymbol{u} \in \mathcal{U}^r_{\text{sub}}} L_{\text{ToM}}(\boldsymbol{e}^\star \mid \boldsymbol{u}) \tag{13}$$

$$L_{\text{ToM}}(\boldsymbol{e}^\star \mid \boldsymbol{u}) \propto \frac{1}{KM} \sum_{k=1}^{K} \sum_{j=1}^{M} \Psi(\boldsymbol{e}_j(\hat{L}^{(k)}, \boldsymbol{u}), \boldsymbol{e}^\star)$$

$$\mathcal{U}^r_{\text{sub}} \triangleq \{\hat{\boldsymbol{u}}_{\text{infer}}\} \cup \{\boldsymbol{u}_i \sim S_r \mid 1 \le i \le N\}$$

where $\boldsymbol{e}(L, \boldsymbol{u})$ denotes a trajectory sampled from a listener model L conditioned on an instruction $\boldsymbol{u}$, and M is the number of trajectories we sample from each listener. Essentially, the score $L_{\text{ToM}}(\boldsymbol{e}^\star \mid \boldsymbol{u})$ of each candidate instruction is the average performance metric of K listeners, each of which attempts to follow the instruction M times.

6 Experimental Setup

Environment and Dataset. We employ Matterport3D (Anderson et al., 2018b), a photo-realistic simulator of the visual perception of a person walking in an indoor environment. At any location, an agent is provided with RGB images capturing the 360-degree panoramic view when looking from that location.

We train and evaluate our models using the Room-to-Room (R2R) language-based navigation dataset. Each data point was collected by asking an English-speaking crowd-worker to write a verbal description of a path in an environment. The dataset is split into a training set (61 environments, 4,675 paths), a seen validation set (environments seen during training, 340 paths), and an unseen validation set (11 environments unseen during training, 783 paths). We train the models using the training set and perform model selection on the unseen validation set. Performance metrics are computed on the seen validation set.

Speaker Models. We evaluate three speaker architectures: (1) a decoder-only GPT-2 pre-trained on text (Radford et al., 2019); (2) an LSTM encoder-decoder (Shen et al., 2022); (3) a Transformer encoder-decoder (Vaswani et al., 2017). Parameters of the latter two models are randomly initialized. Details are in Appendix §A.2.

Human Evaluation. We evaluate each speaker model on 75 paths in the seen validation data split. In the end, we have annotated 1,200 instructions generated by 16 different systems (humans, 3 speaker models, and their ablated and augmented versions). To evaluate a speaker model, we present its generated instructions to a human annotator and ask them to follow the instructions to navigate in Matterport3D environments. We adapt the PanGEA tool[3] to setup a web navigation interface and create a task on Amazon Mechanical Turk (MTurk). We recruit 213 human evaluators in total. More details about the setting are given in Appendix §A.5.

Performance Metrics. The quality of a speaker is determined by the similarity between the intended trajectories and the actual trajectories that the human evaluators generate by following the speaker's instructions. We compute these similarity metrics: **Success rate (SR)** averages binary indicators of whether the final location of a human-generated trajectory is within three meters of the final location of the intended trajectory; **SPL** (Anderson et al., 2018a) weights the success indicator with the ratio between the intended traveling distance and the actual one; and **NDTW and SDTW** are metrics based on dynamic time-warping alignment (Magalhaes et al., 2019), capturing the similarity between two point sequences. NDTW computes only a sequence similarity score while SDTW weights the score with the success indicator.

7 Experiments

We investigate the following questions:

(a) *How well do the speakers perform on our problem?* We find that, despite implementing advanced architectures, these speakers perform poorly compared to human speakers.

(b) *What causes their performance deficiency?* Using our evaluation scheme, we identify that the speakers possess decent search capability but inadequate pragmatic capability.

(c) *Can we improve the speakers by equipping them with better ToM listeners?* We employ ensembles of state-of-the-art instruction-following agents as ToM listeners for the speakers, and obtain significant improvements.

(d) *What are the challenges in bridging the performance gap with human speakers?* We show that instruction-following agents trained with only human-generated instructions are not optimal for serving as ToM listener models.

How well do the speakers perform on our problem? As seen in Figure 3, there is a wide margin between the agent speakers and the human

[3] https://github.com/google-research/pangea

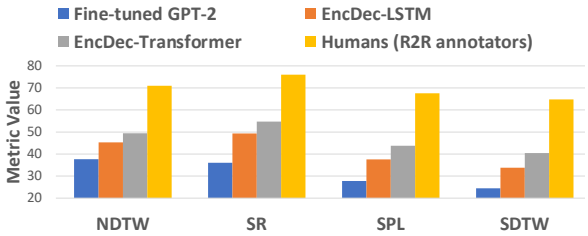

Figure 3: Performance of different speakers on held-out evaluation data, grouped by performance metrics (NDTW, SR, SPL, SDTW). Human speakers are annotators of the R2R dataset. There is a considerable gap between model and human speakers.

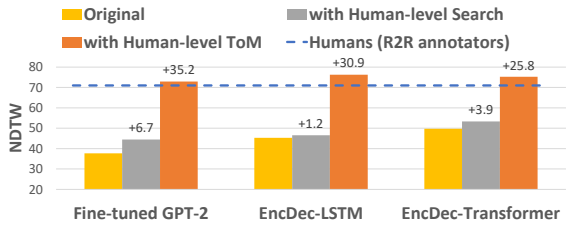

Figure 4: Performance (in NDTW) of the speakers and their human-augmented versions. Possessing human-level pragmatic capability improves performance of the speakers, showing that their original pragmatic capability is highly deficient compared to that of a human.

instructors. The best model speaker (EncDec-Transformer) lags behind the humans by 21.6 NDTW points. The encoder-decoder architecture with cross-attention of EncDec-Transformer outperforms the decoder-only self-attention architecture of GPT-2 (+11.7 NDTW), indicating that fusing the vision and language features too early in an architecture may be detrimental. On the other hand, EncDec-Transformer leads over EncDec-LSTM by 4.1 points NDTW, suggesting that the Transformer architecture is more effective than LSTM in this problem.

What causes the speakers' deficiency? Next, we investigate whether the lack of search or pragmatic capability is responsible for the deficiency of the speakers. The prospective performance gains presented in Figure 4 show that it is under-performed pragmatic capability that primarily causes the models to perform poorly. Specifically, while equipping the models with oracle search capability only improves their performance by 9.4% on average, granting them oracle pragmatic capability nearly doubles their performance metrics. In fact, the search capability of the models is already as good as that of the humans we employ, because the models with oracle pragmatic capability achieve even slightly higher NDTW scores than the human speakers.

Can we improve the speakers by equipping them with better ToM models? Following the procedure described in Section §5, we train state-of-the-art instruction-following agents to serve as ToM listener models for the speakers. Performances of different combinations of speakers and listeners are given in Table 1. We see the largest improvement (+7.9 NDTW) over the best base speaker (EncDec-Transformer) by augmenting this speaker with an ensemble of 10 EnvDrop-CLIP listeners as the ToM model. In Figure 5, we show a qualitative example where having a ToM listener enables the speaker to generate a more accurate instruction. More examples are shown in Appendix §A.7.

We observe that ensemble models consistently outperform single models. More results about the effectiveness of ensemble listeners compared to single listeners are given in Appendix §A.6.

What are the challenges in bridging the performance gap with human speakers? Despite the promising improvements, there remains a large gap of 17.9 NDTW points between our best speaker and the human speakers. As suggested by Figure 4, this gap can be closed by developing accurate ToM models. We argue that optimal ToM models cannot be simply obtained by learning optimal instruction-following agents, because the latter is learned to execute *human-generated* instructions while the former is asked to rank *model-generated* instructions. To illustrate the difference, we measure the agreement between human and model listeners on instructions generated by different speakers. We define the agreement score between a human L_h and a model $\hat{L}$ as

$$\begin{aligned}&\text{Agreement}(L_h, \hat{L})\\&= \text{Average}_{\boldsymbol{u}\in\mathcal{D}_{\text{eval}}}\left(\text{NDTW}(\boldsymbol{e}_h(\boldsymbol{u}), \hat{\boldsymbol{e}}(\boldsymbol{u}))\right) \quad (14)\end{aligned}$$

where $\boldsymbol{e}_h(\boldsymbol{u})$ and $\hat{\boldsymbol{e}}(\boldsymbol{u})$ are the trajectories generated by L_h and $\hat{L}$ given $\boldsymbol{u}$, respectively, and $\mathcal{D}_{\text{eval}}$ denotes the R2R seen validation set.

As seen in Table 2, the listener agents agree more with the humans on human-generated instructions than on model-generated ones. The results imply even an optimal instruction-following agent can fail to improve a base speaker in the presence of an input distribution mismatch. We thus advocate for developing ToM models that are robust or can adapt quickly against covariate shift, and for evaluating

ToM listener L_{ToM}	Base speaker S_{base} Fine-tuned GPT-2	EncDec-LSTM	EncDec-Transformer
None	37.7 (▲ 0.0)	45.3 (▲ 0.0)	49.4 (▲ 0.0)
Single VLN-BERT (Majumdar et al., 2020)	38.9 (▲ 1.2)	39.8 (▼ 5.5)	46.2 (▼ 3.2)
Ensemble of 10 EnvDrop-CLIP (Shen et al., 2022)	37.8 (▲ 0.1)	53.1† (▲ 7.8)	57.3† (▲ 7.9)
Ensemble of 10 VLN↻BERT (Hong et al., 2021)	43.4 (▲ 5.7)	56.4‡ (▲ 11.1)	54.2 (▲ 4.8)
Humans (skyline)	72.9‡ (▲ 35.2)	76.2‡ (▲ 30.9)	75.2‡ (▲ 25.8)

Table 1: Performance (in NDTW) of the speakers when equipped with different ToM models. Each base speaker generates 11 candidates (i.e. $N = 10$). Ensemble listeners significantly improve performance. ‡ and † indicate results that are significantly higher than those of "None" (row 1) with $p < 0.05$ and $p < 0.1$, respectively (two-related-sample t-test).

Instructions generated by	Listener VLN-BERT	EnvDrop-CLIP	VLN↻BERT
Humans (R2R dataset)	65.4 (▼ 0.0)	47.2 (▼ 0.0)	65.0 (▼ 0.0)
Fine-tuned GPT-2	43.1‡ (▼ 22.3)	31.6‡ (▼ 15.6)	39.9‡ (▼ 25.1)
EncDec-LSTM	50.0‡ (▼ 15.4)	43.7 (▼ 3.5)	49.3‡ (▼ 15.7)
EncDec-Transformer	52.1‡ (▼ 13.3)	41.5 (▼ 5.5)	51.9‡ (▼ 13.1)

Table 2: Agreement (in NDTW) of human and model listeners on instructions generated by different speakers. The level of agreement decreases substantially when shifting from human-generated to model-generated instructions. ‡ indicate results that are significantly lower than the human skyline (row 1) with $p < 0.05$ (according to a two-related-sample t-test).

Human: Turn around and walk down the stairs to the bottom. Walk into the kitchen and **stand near the kitchen table**.

EncDec-Transformer: Go down the stairs and **stop at the bottom of the stairs**. [*correct destination is next to dining table*]

EncDec-Transformer + ToM Listener (Ensemble of 10 VLN↻ BERTs): Walk down the stairs and **wait by the dining room table and chairs**.

Figure 5: A qualitative example where the pragmatic speaker (the last model) avoids missing information by simulating the interpretation of the human listener.

performance of these models on model-generated instructions.

8 Conclusion

This work introduces a framework for analyzing task-oriented cognitive capabilities of instruction-generation language models. We show that insights from the analysis are helpful in directing development on these models. Our results highlight the necessity of constructing better ToM models for improving these models. We argue that learning accurate ToM listener models is met with novel, distinct challenges. We hope that our findings will motivate the community to focus more on evaluating task-oriented cognitive capabilities and to create datasets, training methods, and evaluation procedures for enhancing the pragmatic capability of language models.

Limitations

Our work is predicated on hypothetical models of human cognition. These models are still under development by cognitive scientists and need to be validated in more realistic domains. Our method assumes access to a simulation of the environment, which may be costly to construct in some domains.

In general, instruction generation agents pose substantial risk to humans. Previous studies have shown that humans can become overly reliant on AI instructions and commit disastrous mistakes

(Robinette et al., 2016). It is thus important for practitioners to comprehend the constraints of our experimental setting. Our experiments take place in a coarse simulator of real-world indoor environments, which restricts the action and perception of the human listeners. Due to the expensive cost and the large number of agent variants, our human evaluation remains limited in terms of population scale and diversity, and the comprehensiveness of the questionnaires. As each instruction is only evaluated by a single human, we have not investigated the variance of the interpretation of the same instruction among different humans. In addition, human evaluators may "guess" a path even if a part of the instruction is misleading or impossible to follow. Hence, the path-similarity metrics may not reflect faithfully the quality of the instructions. Nevertheless, results shown in Table 4 of § A.5 indicates that instructions generated by our agents are almost as easy to interpret as those generated by humans. But again, these results are still subject to the constraints of our annotator population. To deploy our method, practitioners should carefully re-evaluate its safety and effectiveness in conditions that closely emulate the deployment conditions.

Acknowledgments

We thank Jieyu Zhao, Trista Cao and our reviewers for providing helpful feedback to improve the manuscript. We are also grateful to Patrick Shafto and Ted Sumers for great discussions and references to mathematical models of human cognition.

References

Anne H Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, et al. 1991. The hcrc map task corpus. *Language and speech*, 34(4):351–366.

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. 2018a. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. 2018b. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683.

Jacob Andreas and Dan Klein. 2016. Reasoning about pragmatics with neural listeners and speakers. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Austin, Texas. Association for Computational Linguistics.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Yuwei Bao, Sayan Ghosh, and Joyce Chai. 2022. Learning to mediate disparities towards pragmatic communication. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2829–2842, Dublin, Ireland. Association for Computational Linguistics.

Simon Baron-Cohen, Alan M Leslie, and Uta Frith. 1985. Does the autistic child have a "theory of mind"? *Cognition*, 21(1):37–46.

Emily M Bender and Alexander Koller. 2020. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 5185–5198.

Paul A Bloom and Ira Fischler. 1980. Completion norms for 329 sentence contexts. *Memory & cognition*, 8(6):631–642.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Donna Byron, Alexander Koller, Kristina Striegnitz, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2009. Report on the First NLG Challenge on Generating Instructions in Virtual Environments (GIVE). In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 165–173, Athens, Greece. Association for Computational Linguistics.

Josep Call and Michael Tomasello. 2011. Does the chimpanzee have a theory of mind? 30 years later. *Human Nature and Self Design*, pages 83–96.

Ivan Enrici, Bruno G Bara, and Mauro Adenzato. 2019. Theory of mind, pragmatics and the brain: Converging evidence for the role of intention processing as a core feature ofhuman communication. *Pragmatics & Cognition*, 26(1):5–38.

FAIR. 2022. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*.

Michael C Frank and Noah D Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998–998.

Daniel Fried, Jacob Andreas, and Dan Klein. 2018a. Unified pragmatic models for generating and following instructions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1951–1963, New Orleans, Louisiana. Association for Computational Linguistics.

Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. 2018b. Speaker-follower models for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 31.

Robert Goeddel and Edwin Olson. 2012. Dart: A particle-based method for generating easy-to-follow directions. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1213–1219. IEEE.

Jason M Gold, Richard F Murray, Patrick J Bennett, and Allison B Sekuler. 2000. Deriving behavioural receptive fields for visually completed contours. *Current Biology*, 10(11):663–666.

Noah D Goodman and Michael C Frank. 2016. Pragmatic language interpretation as probabilistic inference. *Trends in cognitive sciences*, 20(11):818–829.

Alison Gopnik and Janet W Astington. 1988. Children's understanding of representational change and its relation to the understanding of false belief and the appearance-reality distinction. *Child development*, pages 26–37.

Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Janosch Haber, Tim Baumgärtner, Ece Takmaz, Lieke Gelderloos, Elia Bruni, and Raquel Fernández. 2019. The PhotoBook dataset: Building common ground through visually-grounded dialogue. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1895–1910, Florence, Italy. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Mark K Ho, Michael Littman, James MacGlashan, Fiery Cushman, and Joseph L Austerweil. 2016. Showing versus doing: Teaching by demonstration. *Advances in neural information processing systems*, 29.

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. 2021. Vln bert: A recurrent vision-and-language bert for navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1643–1653.

Jennifer Hu, Sammy Floyd, Olessia Jouravlev, Evelina Fedorenko, and Edward Gibson. 2023. A fine-grained comparison of pragmatic language understanding in humans and language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. 2019. Stay on the path: Instruction fidelity in vision-and-language navigation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1862–1872, Florence, Italy. Association for Computational Linguistics.

Aishwarya Kamath, Peter Anderson, Su Wang, Jing Yu Koh, Alexander Ku, Austin Waters, Yinfei Yang, Jason Baldridge, and Zarana Parekh. 2023. A new path: Scaling vision-and-language navigation with synthetic instructions and imitation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10813–10823.

Alexander Koller, Kristina Striegnitz, Andrew Gargett, Donna Byron, Justine Cassell, Robert Dale, Johanna D Moore, and Jon Oberlander. 2010. Report on the second nlg challenge on generating instructions in virtual environments (give-2). In *Proceedings of the 6th international natural language generation conference*. The Association for Computer Linguistics.

Michal Kosinski. 2023. Theory of mind may have spontaneously emerged in large language models. *arXiv preprint arXiv:2302.02083*.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. 2020. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4392–4412, Online. Association for Computational Linguistics.

Royi Lachmy, Valentina Pyatkin, Avshalom Manevich, and Reut Tsarfaty. 2022. Draw Me a Flower: Processing and Grounding Abstraction in Natural Language. *Transactions of the Association for Computational Linguistics*, 10:1341–1356.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Angeliki Lazaridou, Anna Potapenko, and Olivier Tieleman. 2020. Multi-agent communication meets natural language: Synergies between functional and structural language learning. In *Proceedings of the 58th*

Annual Meeting of the Association for Computational Linguistics, pages 7663–7674, Online. Association for Computational Linguistics.

Matthew Le, Y-Lan Boureau, and Maximilian Nickel. 2019. Revisiting the evaluation of theory of mind through question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5872–5877.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Jessy Lin, Daniel Fried, Dan Klein, and Anca Dragan. 2022. Inferring rewards from language in context. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8546–8560, Dublin, Ireland. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Gabriel Ilharco Magalhaes, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. 2019. General evaluation for instruction conditioned navigation using dynamic time warping. In *NeurIPS Visually Grounded Interaction and Language (ViGIL) Workshop*.

Kyle Mahowald, Anna A Ivanova, Idan A Blank, Nancy Kanwisher, Joshua B Tenenbaum, and Evelina Fedorenko. 2023. Dissociating language and thought in large language models: a cognitive perspective. *arXiv preprint arXiv:2301.06627*.

Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. 2020. Improving vision-and-language navigation with image-text pairs from the web. In *European Conference on Computer Vision*, pages 259–274. Springer.

Pascal Mamassian, Michael Landy, and Laurence T Maloney. 2002. Bayesian modelling of visual perception. *Probabilistic models of the brain*, 13:36.

Aida Nematzadeh, Kaylee Burns, Erin Grant, Alison Gopnik, and Tom Griffiths. 2018. Evaluating theory of mind in question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2392–2400, Brussels, Belgium. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

David Premack and Guy Woodruff. 1978. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4):515–526.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Paul Robinette, Wenchen Li, Robert Allen, Ayanna M Howard, and Alan R Wagner. 2016. Overtrust of robots in emergency evacuation scenarios. In *2016 11th ACM/IEEE international conference on human-robot interaction (HRI)*, pages 101–108. IEEE.

Homero Roman Roman, Yonatan Bisk, Jesse Thomason, Asli Celikyilmaz, and Jianfeng Gao. 2020. RMM: A recursive mental model for dialogue navigation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1732–1745, Online. Association for Computational Linguistics.

Paula Rubio-Fernandez. 2021. Pragmatic markers: the missing link between language and theory of mind. *Synthese*, 199(1):1125–1158.

Adam N Sanborn and Nick Chater. 2016. Bayesian brains without probabilities. *Trends in cognitive sciences*, 20(12):883–893.

Adam N Sanborn, Thomas L Griffiths, and Daniel J Navarro. 2010. Rational approximations to rational models: alternative algorithms for category learning. *Psychological review*, 117(4):1144.

Maarten Sap, Ronan Le Bras, Daniel Fried, and Yejin Choi. 2022. Neural theory-of-mind? on the limits of social intelligence in large LMs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3762–3780, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Thom Scott-Phillips. 2014. *Speaking our minds: Why human communication is different, and how language evolved to make it special*. Bloomsbury Publishing.

Patrick Shafto, Noah D Goodman, and Thomas L Griffiths. 2014. A rational account of pedagogical reasoning: Teaching by, and learning from, examples. *Cognitive psychology*, 71:55–89.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. 2022. How much can clip benefit vision-and-language tasks? In *Proceedings of the International Conference on Learning Representations*.

Herbert A Simon. 1957. Models of man; social and rational. *wiley*.

Kristina Striegnitz, Alexandre AJ Denis, Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Mariët Theune. 2011. Report on the second second challenge on generating instructions in virtual environments (give-2.5). In *13th European workshop on natural language generation*.

Theodore Sumers, Robert D Hawkins, Mark K Ho, Thomas L Griffiths, and Dylan Hadfield-Menell. 2022. How to talk so ai will learn: Instructions, descriptions, and autonomy. In *Advances in Neural Information Processing Systems*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Hao Tan, Licheng Yu, and Mohit Bansal. 2019. Learning to navigate unseen environments: Back translation with environmental dropout. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2610–2621, Minneapolis, Minnesota. Association for Computational Linguistics.

Michael Tomasello. 2019. Becoming human. In *Becoming Human*. Harvard University Press.

Anna Trosborg. 2010. *Pragmatics across languages and cultures*, volume 7. De Gruyter Mouton.

Takuma Udagawa and Akiko Aizawa. 2019. A natural language corpus of common grounding under continuous and partially-observable context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7120–7127.

Tomer Ullman. 2023. Large language models fail on trivial alterations to theory-of-mind tasks. *arXiv preprint arXiv:2302.08399*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Edward Vul, Noah Goodman, Thomas L Griffiths, and Joshua B Tenenbaum. 2014. One and done? optimal decisions from very few samples. *Cognitive science*, 38(4):599–637.

Pei Wang, Junqi Wang, Pushpi Paranamana, and Patrick Shafto. 2020. A mathematical theory of cooperative communication. *Advances in Neural Information Processing Systems*, 33:17582–17593.

Su Wang, Ceslee Montgomery, Jordi Orbay, Vighnesh Birodkar, Aleksandra Faust, Izzeddin Gur, Natasha Jaques, Austin Waters, Jason Baldridge, and Peter Anderson. 2021. Less is more: Generating grounded navigation instructions from landmarks. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15407–15417.

Heinz Wimmer and Josef Perner. 1983. Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception. *Cognition*, 13(1):103–128.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6720–6731.

Ming Zhao, Peter Anderson, Vihan Jain, Su Wang, Alexander Ku, Jason Baldridge, and Eugene Ie. 2021. On the evaluation of vision-and-language navigation instructions. *European Chapter of the Association for Computational Linguistics*, pages 1302–1316.

Hao Zhu, Graham Neubig, and Yonatan Bisk. 2021. Few-shot language coordination by modeling theory of mind. In *International Conference on Machine Learning*, pages 12901–12911. PMLR.

Hyperparam	GPT-2	Transformer
Learning rate	10^{-4}	10^{-4}
Batch size	4	32
Optimizer	AdamW	AdamW
Num. of training iterations	2×10^5	16×10^4
Max. action steps	15	35
Max. instruction length	100	80
Image feature size	2048	512
Orientation feature size	128	128
Embedding dropout	0.1	0.3
Hidden size	768	512
Num. of hidden layers	1	1
Hidden-layer dropout rate	0.0	0.6
Num. of encoder layers	-	2
Num. of decoder layers	12	2
Transformer dropout rate	0.1	0.3
Beam size	5	1

Table 3: Hyperparameters for training the GPT-2 EncDec-Transformer speakers.

A Appendices

A.1 The Room-to-Room dataset

The R2R dataset (Anderson et al., 2018b) was originally created for training instruction-following agents. Each data point was collected by asking a crowd-worker to write a verbal description of a path in an environment. In the end, each path was annotated with three instructions. Each instruction contains 29 words on average. The dataset is split into a training set (61 environments, 4,675 paths), a seen validation set (340 paths) whose paths are sampled in the training environments, and an unseen validation set (11 environments unseen during training, 783 paths). We do not use the unseen test split because it does not provide ground-truth paths of the descriptions. We use the dataset consistent to their MIT License.

A.2 Implementation of Speaker Models

We train the speakers with a standard maximum-likelihood objective using the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of 10^{-4}.

The speaker models take a sequence of visual observations and actions from the trajectory $\boldsymbol{e}^\star$ as input and output a text instruction $\boldsymbol{u}$. The model is trained to estimate conditional probability $S_\theta(\boldsymbol{u}|\boldsymbol{e}^\star)$. We use grid search to select the model and training hyperparameters, and the best-found values are listed in Table 3.

Input. The input trajectory $\boldsymbol{e}^\star$ is a sequence of panoramic views and actions. Each panoramic view at time step t is represented by 36 vectors $\{o_{t,i}\}_{i=1}^{36}$, each of which is a visual feature vector extracted from a pre-trained vision model concatenated with orientation features describing the agent's current gaze direction. The image features of the GPT-2 model are extracted from a ResNet-152 model (He et al., 2016), whereas those of the encoder-decoder models are from a CLIP model (Radford et al., 2021). Each ground truth action $a_t^\star$, which moves the agent to an adjacent location, is represented by image features from the gaze direction of the agent when looking towards that adjacent location, and orientation features capturing the direction of the adjacent location relative to the agent's current gaze direction.

Output. The output of a speaker model is a language instruction describing the input trajectory. At test time, the GPT-2 model employs beam search, and the encoder-decoder models generate instructions via greedy decoding (Shen et al., 2022).

Training Objective. We train the speakers with maximum-likelihood objective:

$$\max_{\theta} \sum_{(\boldsymbol{u}^\star,\boldsymbol{e}^\star)\in\mathcal{D}_{\text{train}}} \sum_{t=1}^{|\boldsymbol{u}^\star|} \log S_\theta(\boldsymbol{u}_t^\star \mid \boldsymbol{e}^\star, \boldsymbol{u}_{<t}^\star) \quad (15)$$

where θ is the speaker model parameters, $\boldsymbol{u}_t^\star$ is t-th word of the ground-truth instruction, and $\boldsymbol{u}_{<t}^\star$ is the first $t-1$ words of the instruction.

We select the best model based on the unseen-validation BLEU score (Papineni et al., 2002) of the model-generated instructions with the respect to the ground-truth instructions.

Tools. We use SacreBLEU 2.2.1 to compute BLEU scores. For preprocessing and implementing the speaker models, we use Pytorch 1.7.1, NLTK 3.6.7, SentencePiece 0.1.97, and Huggingface Transformers 4.5.1.

Computation. The GPT-2 model has 124.4 million parameters, and was trained for 24 hours on single NVIDIA GEFORCE RTX 2080 Ti. The EncDec-LSTM model has 7.5 million parameters, taking 24 hours to train on single NVIDIA RTX A6000. The EncDec-Transformer model has 56.6 million parameters, trained on single NVIDIA RTX A6000 for 48 hours.

Speaker	SR ↑	SPL ↑	NDTW ↑	SDTW ↑	Path Len ↓	Interpretability ↑
	Performance Metrics					
Without ToM listener						
Finetuned GPT-2	36.0	27.8	37.7	24.5	20.9	2.9
EncDec-LSTM	49.3	37.6	45.3	33.8	17.4	3.3
EncDec-Transformer	54.7	43.8	49.4	40.4	15.8	3.4
With 10 VLN↻BERT as ToM listener						
Finetuned GPT-2	46.7	30.9	43.4	28.1	21.2	3.0
EncDec-LSTM	54.7	46.0	56.4	41.9	14.0	3.1
EncDec-Transformer	52.0	44.0	54.2	41.6	17.7	3.2
Humans (R2R dataset)	76.0	67.6	71.0	64.8	14.2	3.6

Table 4: Humans evaluation results on instructions generated by the speaker models. The similarity metrics are defined in §6. *Path Len* measures the average length of the generated trajectories. *Interpretability* indicates how easy or difficult to follow the instructions according to human evaluators (without knowing the ground-truth trajectory).

A.3 Fine-tuning GPT-2 Speaker Model

To represent the trajectory features as a sequence of feature vectors to feed into the GPT-2 model, we first average the view features $\bar{o}_t$ for each time step:

$$\bar{o}_t = \frac{1}{36}\sum_{i=1}^{36} o_{t,i} \tag{16}$$

We compute the input features $\boldsymbol{e}_t^\star$ by concatenating the panoramic view features and ground truth action features:

$$\boldsymbol{e}_t^\star = [\bar{o}_t; a_t^\star] \tag{17}$$

The sequence of feature vectors $\boldsymbol{e}^\star$ representing a trajectory is calculated as follows

$$\boldsymbol{e}^\star = [\tanh(\boldsymbol{e}_1^\star W); \cdots; \tanh(\boldsymbol{e}_T^\star W)] \tag{18}$$

where W is parameters of a linear layer.

For the instruction $\boldsymbol{u}^\star$, we perform an embedding look-up of its words. Then, we first prompt the model with $\boldsymbol{e}^\star$ and then train it to generate $\boldsymbol{u}^\star$ as a suffix.

A.4 Training Encoder-Decoder Speaker Models

Our EncDec-LSTM model follows the implementation of the speaker in Shen et al. (2022). We implement the EncDec-Transformer model by replacing the LSTM layers of the speaker model described in Tan et al. (2019) with Transformer layers (Vaswani et al., 2017).

A.5 Human Evaluation Interface and Data Collection

We pay the evaluator $5.20 per task which takes about 25 minutes, and the payment is decided by state minimum wage. For each task, we ask the evaluator to follow six instruction-following sessions. One of the six sessions, which appears in all tasks, is a quality-control test featuring an easy-to-follow human-written instruction. We only approve an evaluator if they navigate successfully to the goal destination in this test. Following Zhao et al. (2021), we instruct the judges to not explore the environments unnecessarily and not wander back and forth unless they are lost. We record the trajectories created by the human and use them to compute the performance metrics.

Figure 6 shows the interface for our human evaluation to collect annotations, which we adapted from the PanGEA tool[4] consistent with their Apache License v2.0. After a human evaluator finishes following an instruction, we recorded the path they generate and compute similarity metrics with respect to the ground-truth path. After the instruction-following sessions, we ask each evaluator to assess the interpretability of the instructions by asking them how easy (or difficult) it was for them to follow the instruction. We provide four rating levels ranging from "*1: I couldn't follow any part of the instruction*" to "*4: very easy, the instructions gave*

[4] https://github.com/google-research/pangea

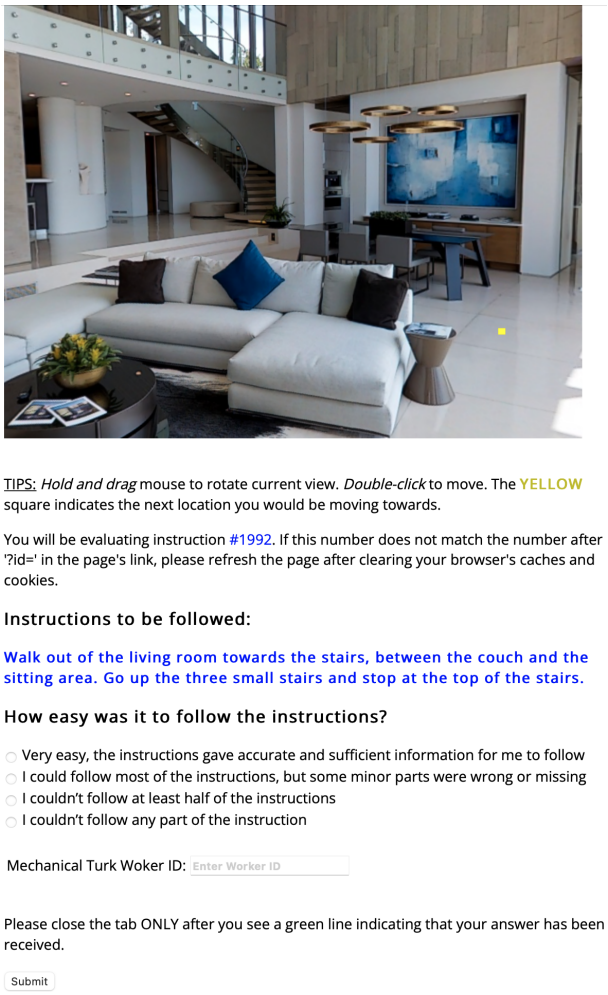

Figure 6: Human evaluation interface.

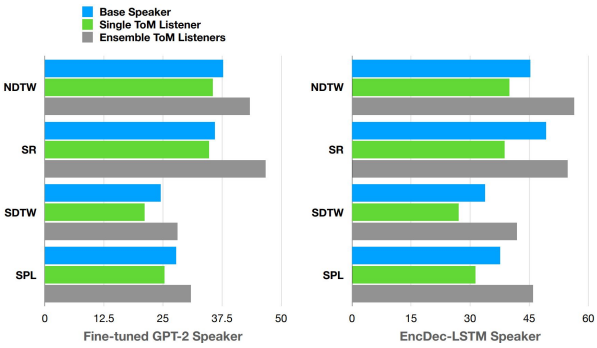

Figure 7: Comparison of using single and ensemble ToM listeners.

accurate and sufficient information for me to follow". The answer of the evaluators is converted to a score between one and four.

Table 4 shows the human evaluation results of the three speaker models we evaluated.

For the human evaluation survey, participants will be restricted to those fluent in English. There are no other restrictions for this study. Participants must be at least 18 years old. Before completing the survey, participants will be shown information about the task requirement: *You are in a building, and are provided with a short set of instructions to navigate to a target location. Please follow the instructions as closely as possible. Do NOT explore the building unnecessarily and do NOT wander back and forth unless you are lost. Please read ALL of the instructions before you start moving.*

We waive consent for this study for several reasons: 1) Minimal risk: The study collects minimal identifying information and there are no known risks for the subjects beyond everyday computer use. 2) Rights and welfare: All participants will be shown all information regarding task requirements before they complete our survey. They must consent to performing the task before they are shown the questions. 3) Practicality: Since the sessions are conducted online on a large scale, it would be infeasible to require all users to send a signed form. 4) Post participation information: We do not think there is any pertinent information that is not already shared with the participants before or during our experiments, so we do not feel it is necessary to provide any additional information after participation. PI information will be shared with the participants to enable them to obtain additional information about the study post completion.

For data anonymization, we removed the only identifying information, Amazon Mechanical Turk ID, after collecting the human annotation data. This information would also be removed for future dataset release. The dataset will be released under MIT license terms, which are compatible with those of the tools used to create it, and will be intended for research usage.

A.6 Single vs. Ensemble Listeners

As a preliminary experiment, we compare the effectiveness of a single and an ensemble of 10 VLN↻BERT agents when serving as the ToM model of a speaker. Results in Figure 7 show that the ensemble listener is significantly better than the single listener for two different speakers.

A.7 Qualitative Examples

In Figure 8, we show additional qualitative examples where having a ToM listener enables the speaker to generate a more accurate instruction.

Human: Walk along the patio towards the couch. **Stop next to the table that is in front of the couch**.
EncDec-Transformer: Walk straight down the walkway and **stop next to the first chair on the left**. *[Correct destination is next to the couch and table]*
EncDec-Transformer + ToM Listener (Ensemble of 10 EnvDrop-CLIP): Go straight down the walkway. Go straight and **pass the two chairs**. Stop near the landing with the pillars.

(a)

Human: Turn around and exit out the door in the right corner. Enter the next room and walk straight ahead towards the outdoor area. **Stop once you pass the columns and are in the middle facing all the chairs looking outside**.
EncDec-LSTM: Exit the bathroom and turn left. Walk past the bed and **wait by the two chairs**. *[Correct destination is next to the chairs in the outdoor area]*
EncDec-LSTM + ToM Listener (Ensemble of 10 EnvDrop-CLIP): Walk out of the bathroom and make a left. Walk through the bedroom and continue straight towards the red chair. **Stop at the chair before getting to the red front of the patio.**

(b)

Figure 8: Additional qualitative examples where the pragmatic speaker (the last model) avoids missing information by simulating the interpretation of the human listener.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations.

☑ A2. Did you discuss any potential risks of your work?
Section Limitations.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section Abstract, and Section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 6, Section 7, Section A.1, A.5.

☑ B1. Did you cite the creators of artifacts you used?
Section 1, Section 6, Section A.1.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section A.1, Section A.5.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section A.1, Section A.5, Section Limitations.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Section A.5.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 1, Section A.5

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section A.1, Section 6.

C ☑ Did you run computational experiments?

Section 6.2, Section A.2.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section A.2

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 6.2, Section A.2.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 7.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section A.2.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Section 6, Section A.5.

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Section A.5

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Section 6, Section A.5.

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Section A.5

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Section A.5.