Understanding Common Ground Misalignment in Goal-Oriented Dialog: A Case-Study with Ubuntu Chat Logs

Rupak Sarkar[1], **Neha Srikanth**[1], **Taylor Pellegrin**[2], **Rachel Rudinger**[1], **Claire Bonial**[3], and **Philip Resnik**[1]

[1]University of Maryland, College Park
[2]Oak Ridge Associated Universities
[3]Army Research Lab
{rupak,nehasrik}@umd.edu

Abstract

While it is commonly accepted that maintaining common ground plays a role in conversational success, little prior research exists connecting conversational grounding to success in task-oriented conversations. We study failures of grounding in the Ubuntu IRC dataset, where participants use text-only communication to resolve technical issues. We find that disruptions in conversational flow often stem from a misalignment in common ground, driven by a divergence in beliefs and assumptions held by participants. These disruptions, which we call conversational friction, significantly correlate with task success. While LLMs can identify overt cases of conversational friction, they struggle with subtler and more context-dependent instances that require pragmatic or domain-specific reasoning.

1 Introduction

Effective communication between humans in conversation hinges on a set of facts and beliefs relevant to the conversation, or the *conversational common ground* (Stalnaker, 1978, 2002; Clark and Brennan, 1991), that is shared between participants. They must collaboratively maintain and update this common ground for the conversation to progress successfully. This dynamic, ongoing management is essential: a misalignment or misunderstanding can disrupt the communicative flow, potentially leading to confusion or conflict.

Typically, much of this maintenance is implicit: listeners acknowledge their understanding through verbal and non-verbal cues, making research on common ground and its role in conversational success challenging. When participants successfully complete a goal-oriented conversation without visible disruption or misunderstanding, it is unclear what information constitutes their common ground. Many studies sidestep this by constraining the conversational setting to physically grounded tasks, such as building objects in Minecraft-like worlds (Narayan-Chen et al., 2019; Bara et al., 2021), providing environments where researchers can infer participants' common ground through their actions.

Turn	Speaker	Utterance
4	B	or you can use tab completion. Type cd rt [tab]
5	B	Are you at the terminal?
6	B	If you got that far, the cd command should be easy. lol
7	A	That command, right?
8	B	I am trying to help. What is the error you are getting with cd command?
9	A	What do I type into the terminal for the cd command?

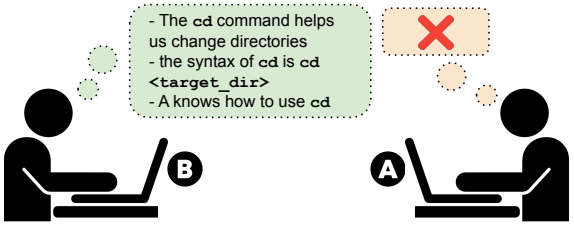

Figure 1: An annotated instance of conversational friction. Though it is challenging to access propositions in a speakers' perception of common ground, certain propositions in B's version of common ground are revealed (green thought bubble) when there is a misalignment between the two participants. B assumes A knows about the cd command, which is proven false by A in Turn 9.

We address this challenge in a different way—by focusing on *miscommunications* as a window into the shared beliefs of conversational participants. Consider the conversation in Figure 1. At the outset, the common ground includes beliefs such as *"A is an Ubuntu user"* and *"A is accessing a Linux terminal"*, etc. Following Turn 4, B believes that *"the syntax of* cd *is* cd ⟨target_dir⟩" is now part of the conversational common ground. Turn 9 reveals that this assumption was incorrect through an *observable* interruption precluding A and B from proceeding towards A's main conversational goal.[1]

[1]Grosz and Sidner (1986) would distinguish this goal as the *discourse purpose*.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3200–3215

We use the term **conversational friction** to describe such an instance of disruption in communicative flow, caused by a misalignment in speaker beliefs about what is present in the common ground.[2] Frictions reveal the importance of maintaining common ground, as they require re-negotiation (Clark and Wilkes-Gibbs, 1986) of content: instead of making progress, participants need a "conversational detour" to align their interpretations of previously shared content.

This work explores two key questions. First, (**RQ1**) to what extent is achieving a participant's goal—or *success*—associated with the presence or absence of conversational friction? And (**RQ2**), can large language models (LLMs) identify and explain sources of friction in human conversations? We seek to shed light on the relationship between conversational friction, which serves as evidence of a misalignment in common ground, and the success of participants in achieving a shared goal.

To achieve this, we annotate real-world conversations involving Ubuntu users attempting to fix an issue or bug.[3] We annotate 200 conversations from the Ubuntu Dialog Corpus (Kummerfeld et al., 2019), a corpus of conversations among users solving issues when using the Ubuntu operating system.[4] Each conversation is annotated for the presence of conversational friction (supporting RQ2) and the degree of task success (supporting RQ1) (§3.1) to analyze the importance of maintaining common ground (§4). Then, we explore the ability of LLMs to identify friction and compare their explanations with human explanations (§5).

Not only are LLMs increasingly relied upon as conversational partners (Minaee et al., 2024), they are also used as mediators (Tan et al., 2024) or to generate conversational summaries (Ramprasad et al., 2024). As such, it is important to know if they track the common ground, an essential component of smooth communication. Our analyses of friction and repair reveal that **friction often arises from misalignment in common ground**, particularly when participants hold diverging assumptions about the task or possess varying levels of domain expertise. Furthermore, we find that while models are able to detect overt signals of friction, **they struggle to identify subtler and more context-dependent instances of misalignment** that require deeper pragmatic or domain-specific reasoning[5].

	Kummerfeld et al. (2019)	2-person conversations	Ubuntu-CG	Analysis Subset
#Conversations	496469	282027	200	70
Average Length	7.16	5.84	39.75	51.78

Table 1: Overview of our dataset. We use 200 dyadic conversations sampled from Kummerfeld et al. (2019) totaling 7590 turns for friction detection, and a subset of 70 for grounding act annotation (§3.3)

2 Background

The *conversational common ground* (CG) is a body of statements treated as mutual knowledge among participants (Stalnaker, 1978). It guides both how speakers choose their utterances and how they want them to be interpreted (Stalnaker, 2002)[6]. Subsequently, Clark and Brennan (1991) define common ground as a collection of mutual knowledge, beliefs, and assumptions that humans maintain collaboratively through the process of *grounding*.[7]

In early computational work studying CG, Traum and Allen (1992) proposes breaking down a conversation into Discourse Units, where humans collaboratively build common ground through speech acts such as `RequestRepair`, a speech act through which the speaker urges their conversational partner to ground an utterance.[8]

While it is acknowledged that maintaining CG is of some importance to conversational success (Traum, 1995), there has been little empirical work that explicitly ties participant effort in maintaining it to the success of an end goal. In this study, we look at the importance of grounding in the success of naturally-occurring goal-oriented conversations. **Specifically, we focus on conversational friction as evidence of the loss and re-negotiation of common ground.**

[2]Hereafter we use the terms "friction" and "conversational friction" interchangeably.

[3]We will release code and data upon publication.

[4]Ubuntu (`https://ubuntu.com/desktop`) is one of the most popular free and open-source Linux-based operating systems in the world.

[5]Code and data can be found in `https://github.com/styx97/cg-misalignment`

[6]Even before Stalnaker, Paul Grice mentioned propositions having *common ground status* in his William James lectures (Stalnaker, 2002). For a thorough discussion of common ground in linguistics, see Geurts (2024).

[7]We focus only on discourse-theoretic grounding and do not delve into symbol grounding (Harnad, 1990), as exemplified in mapping a linguistic concept to a visual scene (see Cohen et al. (2024) for a survey of methodologies for robotic language grounding); however, we embrace the conceptual relationship between both types of grounding, as described in Chandu et al. (2021).

[8]See Table 1 of Traum and Allen (1992) for a full list.

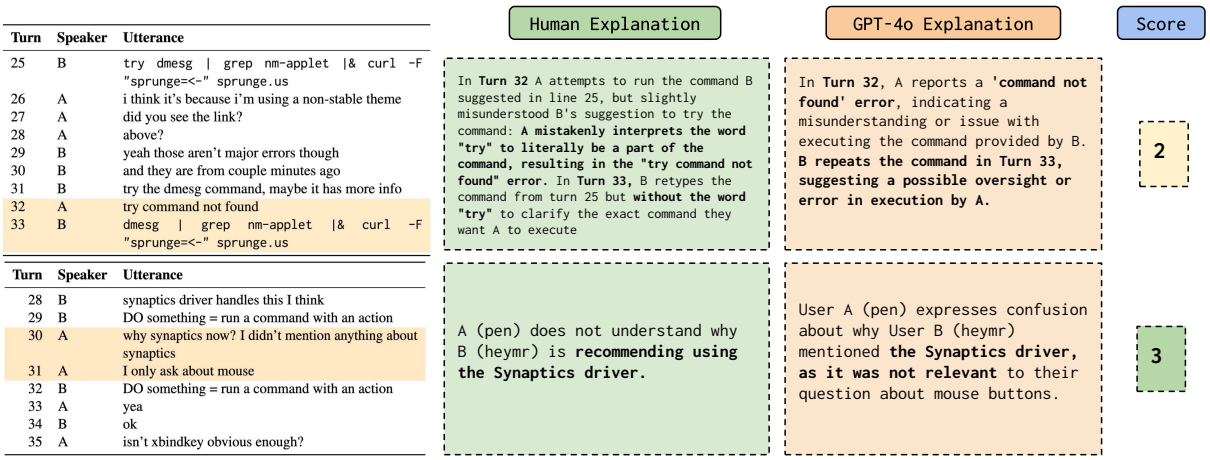

Figure 2: Comparing GPT4o and human explanations for the cause of friction. GPT4o explanations align with humans when friction is explicit (row 2). In a more implicit case of friction (row 1), GPT4o fails to capture the true reason for friction—A misreading "try" as part of a terminal command (Turn 25), revealed in the error message "try command not found" in Turn 32.

Typical conversations in our dataset (e.g., in Table 3) involve two participants (an `asker` and a `helper`) collaboratively attempting to solve a Linux bug over a text channel. This consists of several steps—the `asker` must describe their issue (often with insufficient knowledge of Linux), and the `helper` must understand their goal to propose a solution. This makes the setting well-positioned for studying friction and grounding.

Dataset: Ubuntu-CG. The Ubuntu Dialog Corpus satisfies several criteria for our study; (1) conversations are *naturally* goal-oriented (e.g., resolving an error in Ubuntu), incentivizing participants to communicate effectively; (2) participants have to establish CG from scratch; (3) conversations are text-only; and (4) are multi-turn, ranging from three turns to over 100, giving users ample time to build and utilize CG. §7 discusses other datasets we considered.

The Ubuntu Dialog Corpus (Lowe et al., 2015) contains conversations scraped from the `#Ubuntu` IRC channel, where users primarily discuss features, issues, and bugs related to the Ubuntu operating system. This requires disentangling conversations from a single message stream. Kummerfeld et al. (2019) found that the disentanglement strategy originally used had a high error rate, and released a cleaner version. We use a sample of 200 two-person conversations from this cleaner corpus, upsampling longer conversations to study diverse behavior (Table 1). We refer to this subset as Ubuntu-CG (Common Ground).

Success	Mean Length (Std.)	Friction (%Present)	Mean #Friction (when Present)
1 (No Progress)	31.90 (24.99)	57.60 (30/52)	2.43
2 (Some Progress)	43.86 (25.73)	55.05 (49/89)	2.06
3 (Success)	40.45 (28.84)	50.84 (30/59)	2.13

Table 2: An overview of Ubuntu-CG, annotated for friction and task success. Conversations where participants make *some* progress towards their task contain lower occurrences of friction (Column 4).

3 Conversational Friction

We now focus on detecting and understanding causes of conversational friction in Ubuntu-CG. Users with varying levels of expertise or familiarity with Linux and English try to collaboratively fix an issue with Ubuntu over text.[9] This setting naturally lends itself to frequent occurrence of conversational friction. But how often is friction resolved in subsequent grounding, and does it have a demonstrable effect on the success of a conversation? To answer these questions, we collect a dataset of instances of conversational friction.

Task. Given a conversation consisting of M turns, the task is to identify a **list of turn intervals** $\{I_1 = m_{x_1}...m_{y_1}, I_2 = m_{x_2}...m_{y_2}, ...\}$ exhibiting conversational friction, or instances of disruption in communicative flow caused by a misalignment in speaker beliefs *about what is present in the CG*, along **with an explanation** of why each interval exhibits friction. A strong indicator of conversa-

[9] Some of the users are (self-professed) non-native speakers of English.

tional friction is when a participant asks the other participant to *repair* their conversation. However, *implicit* cases of friction require identifying when a user is struggling to keep up with the conversation. Note that not all followup questions indicate friction. For example, clarification questions that ask for information not assumed to be in the CG are *not* cases of conversational friction.

3.1 Annotating Conversational Friction

Three computer science undergraduates familiar with Linux annotate conversations by (1) identifying turn intervals which exhibit friction along with explanations and (2) judging the success of overall conversation on a three-point scale with respect to the conversational goal. Annotators were paid $18/hr to annotate 200 conversations totaling 7950 turns, taking over 80 hours to complete. Since conversations date back to over a decade ago, they often contain antiquated terms or references that annotators were unfamiliar with. To mitigate this, we provide explanations generated by `gpt-4o` (OpenAI, 2024) of technical terms in dialog turns. For example, the model-generated elaboration in Table 9 (Row 1) in Appendix explains that *"dapper"* and *"feisty"* refer to Ubuntu versions 6.06 and 7.04. We make these elaborations available to various models in our computational experiments as well.

Conversational Success. In addition to friction, annotators assess how successful participants were in solving the issue by scoring the conversation on a three-point scale. A score of 1 denotes that the conversation was not helpful to the `asker` at all, and no progress was made; a score of 2 denotes some progress towards solving or diagnosing the issue, and a score of 3 indicates that the issue was solved. In cases where experienced `helpers` propose alternate solutions, success is measured by progress towards this *new* goal. Table 2 shows the overall statistics, and instructions for friction and success annotation can be found in the Appendix A.6. We obtain an agreement of $\alpha = 0.58$ on success annotation as measured by Krippendorff's Alpha (Castro, 2017).

3.2 Measuring Friction Agreement

Measuring inter-rater agreement on friction detection is not straightforward, since we must account for agreement both in identifying an instance of friction *and* the turn interval in which it occurs. To simplify this measurement, we compute

Turn	Speaker	Utterance	Grounding Act
0	A (`asker`)	i have recently installed nvidia driver (working), but upon restart i get an error message: "failed to initialize nvidia kernel module" - anyone have any tips? :)	
1	B (`helper`)	manf. drivers?	
2	A (`asker`)	sorry im not familiar with manf. drivers. i installed NVIDIA-Linux-x86-195.36.24-pkg1.run :)	`RequestRepair`
3	B (`helper`)	yes i meant from nvidia site :)	`Repair`

Table 3: A typical conversation in our dataset, containing instances of `RequestRepair` and `Repair` acts.

overlap metrics for each pair of annotators, as in Markowska et al. (2023). Agreement between an annotator pair is reported as the average of a modified version F1 score to measure interval overlap. Specifically, for two annotators A_1 and A_2, we **average** two F1 scores—one treating annotations from A_1 as ground truth and those from A_2 as predictions and vice versa.[10] We compute agreement in two different settings:

Friction Found. In this *relaxed* setting, an interval is "found" if *any* turn within that friction window is part of *any* predicted interval. This setting does not require one-to-one mapping between predicted and gold friction instances. Here, predicting one dialog turn within a gold friction interval is equivalent to predicting all turns correctly.

Friction Overlap. We consider a second setting that rewards the *degree* of overlap with the gold interval. We first match each instance of friction with the predicted instance with the highest overlap, ensuring a one-to-one mapping between a predicted and gold friction interval. For each *matched* interval, we compute the Jaccard similarity between the two intervals, resulting in higher scores for predictions that better align with human-annotated instances of friction and penalizing predicting multiple short or overtly long instances.[11] A perfect score indicates an exact overlap between predicted and gold intervals. We use these same two settings to compute model performance (Table 6). Table 4 shows agreement between pairs of annotators. Given their high agreement, A1 and A2 annotated 80% of the final dataset, while A3 annotated 20%.

3.3 Annotating for Grounding Acts

Our annotations reveal that successful conversations contain less friction (Table 2). However,

[10] Our operationalization of F1 makes it asymmetric, hence $F1(A_1, A_2)$ is not guaranteed to be equivalent to $F1(A_2, A_1)$.

[11] This is similar in spirit to methods discussed in Ortmann (2022), adapted for our task.

	A_1	A_2	A_3
A_1	–	65.91 / 25.86	48.0 / 18.21
A_2	–	–	43.88 / 13.58
A_3	–	–	–

Table 4: Inter-rater agreement of detecting conversational frictions in Ubuntu-CG. Each cell contains the average of F1 scores between two annotators in two settings described in § 3.2 (Friction Found/Span Overlap).

Degree of Progress	#Convs	Instances (Repair/ReqRepair)	Unaddressed ReqRepair (%)
2 or 3	49	102 (83/75)	22.67
No Progress (1)	21	50 (38/36)	30.56

Table 5: Summary of success and grounding acts in our analysis subset of 70 conversations. In conversations with no progress, more requests for repairs go anaddressed.

when friction is present, can participants collaboratively rebuild CG to complete tasks successfully? To better understand and model loss and repair of CG, we identify particular *speech acts* associated with repair in friction intervals, as in Levelt (1983); Heeman and Allen (1994); Bohus and Rudnicky (2008); Bonial et al. (2022). For each friction interval I in a conversation, we identify specific turns within I expressing two grounding acts: `RequestRepair` and `Repair` (Traum and Allen, 1992). `RequestRepair` indicates whether a participant, spotting friction, *explicitly* requests conversational repair from their partner. `Repair` indicates whether friction was *addressed* by either participant with a clarification (Table 3).

Identifying these acts not only helps us determine whether participants recovered from friction, but also helps us to study in greater detail whether models can detect friction. For example, this framework allows us to measure whether models detect friction only when in the presence of explicit requests or if they can identify *implicit* cases of common ground misalignment. This is important, as using LLMs as conversational partners or as mediators in human-human conversations depends on their ability to detect *implicit* cases of friction.

We sample 70 conversations containing 152 instances of friction to study the effects of grounding on task success. 21 conversations received a success score of 1 (No Progress), 26 received a score of 2 (Some Progress), and 23 received a score of 3 (Success). Since conversations with friction tend to be longer, this sample has a higher average length than our dataset overall. Two authors annotated each friction instance in this subset for the presence or absence of `RequestRepair` and `Repair`acts, obtaining inter-rater scores of 0.69 on `RequestRepair` , and 0.63 on `Repair` , measured using Cohen's Kappa (Cohen, 1960).

4 Analysis of Grounding in Ubuntu-CG

We study the relationship between the presence of conversational friction in goal-driven conversation and its success in Ubuntu-CG, and present our principal findings from the data below.

Successful conversations contain less friction. In Ubuntu-CG, 61% percent of conversations contained friction. In contrast, of conversations where the `helper` succeeded in solving the `asker`'s issue (receiving a score of 3), only 54.5% contained friction (Table 2). Conversations where participants make some progress or succeed contain less friction on average as compared to conversations where they did not make *any* progress, as the former exhibits some amount of grounding effort by the participants (Table 2, Column 4). This is further supported by the proportion of unaddressed repair efforts (Table 5, Column 4).

Friction is more likely in longer conversations. While conversation length shows no clear pattern with task success (Table 2), the mean length of a conversation containing friction is 49 (median 55), as compared to an average of 29 (median 22) of those without friction. Compared to the overall mean length of the dataset 40.56 (median 33), it is plausible that conversational friction and repair through the process of grounding contribute to the increased number of turns it takes to complete the conversation.

4.1 Role of Grounding Acts in Task Success

Conversations where participants could not make *any* progress towards diagnosing a particular issue (success score of 1) are characteristically different from conversations receiving a score of 2 or 3. In a retrospective study, we analyze the presence of grounding acts (`RequestRepair` and `Repair`) in conversations that received a score of 1 (No Progress) as compared to conversations receiving a score of 2 or 3 (Some Progress or Success).

We focus on the proportion of `RequestRepair` acts that were not addressed. This captures instances of friction where, despite one participant spotting a potential mismatch in common ground, their efforts are not reciprocated by their conversational partner. Notably, conversations with no progress exhibited a higher proportion of these unacknowledged `RequestRepair` acts (Column 4 in Table 5). This further shows that achieving a communicative goal requires *both* participants to engage in grounding.

5 Can LLMs Identify Conversational Friction?

Identifying friction in ongoing conversations is a first step towards analyzing the content of the CG. After establishing simple finetuned baselines on the task of conversational friction detection, we go on to explore whether larger LLMs can identify and explain instances of conversational friction in Ubuntu-CG.

5.1 Experimental Setup

Encoder-Only Baseline. Before moving on to prompting, we first explore a baseline setting in which we finetune a small encoder-based model `distilroberta-base` (Sanh et al., 2019) on five randomly-split folds of Ubuntu-CG (Appendix A.2). Given an excerpt of a conversation consisting of a target turn t and a context window of k turns before and after t, the model is trained to predict whether t is part of an annotated instance of friction. We report results on context windows of three and five, and notice no significant improvement in using a higher k. Finetuning a lighter-weight model allows us to understand the extent to which friction is identifiable from surface features.

Decoder-Only Models. Given a *full* conversation, we prompt several larger decoder-only LLMs to output *a list of turn intervals* exhibiting friction (Prompt A.1 in Appendix).[12] For each predicted turn interval, the model must provide a brief explanation for the cause of friction. We also include a setting where we provide models with the elaborations of technical terminology as outlined in §3.1 ("w Elab"). Note that in this setting, an LLM predicts possible friction intervals in a single pass, in contrast to the encoder baseline, where models make predictions on every single turn separately, taking neighboring turns as context.

Evaluation Metrics. We evaluate models in the **Friction Found** and **Friction Overlap** settings 3.2. While **Friction Found** allows models like `Llama-3.1-8b-Instruct` (Touvron et al., 2023) to obtain high recall scores by over-predicting friction intervals, **Friction Overlap** penalizes this behavior. For all experimental settings, we set temperature to 0.01.

5.2 Results

The F1 scores of our baseline encoder-only models (Table 6) in two settings (context window $k = 3$ and $k = 5$) give us an estimate of the degree to which friction is identifiable from shallower, local features as opposed to more complex, contextual, and implicit cases of friction like subtle clarification questions or other pragmatic phenomena. Error analysis reveals that instances predicted correctly by our baselines often contained explicit markers of friction, such as a user expressing frustration or dissatisfaction. While these baselines were competitive with several prompting-based methods, it is difficult to interpret the results beyond error analysis since we do not have access to explanations. However, it does indicate that our dataset also contains surface features such as explicit expressions of frustration or anger that are learnable by a small encoder-only model.

Under both evaluation settings, `gpt-4o` without any further technical elaborations obtained the highest F1 score. We use this setting for all further error analysis and ablations. All models over-predict friction intervals (see column #Predictions in Table 6).

The effect of `gpt-4o` Elaborations. Explaining technical terms with `gpt-4o` helped our human annotators better understand the flow of information in a conversation. However, in the relaxed evaluation setting of **(Friction Found)**, adding elaborations does not improve prediction scores of models. For `Llama-3.1-8b-Instruct` and `gpt-4o-mini`, adding elaborations improves recall and hence the overall F1 score. This may be due to elaborations "sharpening" the predicted intervals.

Ablations. Human annotators do not always agree on the location of friction and repair-related

[12] We experimented with several prompting strategies such as adding random exemplars, self-consistency, and chain-of-thought reasoning, but found that they did not beat the F1 scores obtained simply by asking the model to detect friction windows along with brief explanations of why a dialog window represents friction.

Model	Friction Found			Friction Overlap			#Predictions
	Precision	Recall	F1	Precision	Recall	F1	
gpt-4o	31.50	43.69	**34.01**	13.50	18.74	**14.61**	495
gpt-4o w/ Elab.	31.63	37.46	32.22	13.54	16.59	14.00	435
gpt-4o-mini	**32.75**	27.86	28.01	**13.67**	12.32	12.10	316
gpt-4o-mini w/ Elab.	28.54	28.67	26.51	13.63	14.11	12.81	392
Llama-3.1-8b-Instruct	16.72	47.28	22.53	6.87	18.72	9.14	**1282**
Llama-3.1-8b-Instruct w/ Elab.	15.98	46.33	21.73	7.11	**20.02**	9.58	1253
Llama-3.1-70b-Instruct	21.70	**48.09**	27.97	8.93	20.26	11.59	857
Llama-3.1-70b-Instruct w/ Elab.	16.72	39.83	22.06	7.35	16.76	9.52	959
distilroberta-base ($k = 3$, finetuned)	19.89	48.07	27.99	7.70	18.36	10.80	-
distilroberta-base ($k = 5$, finetuned)	18.32	46.43	26.16	8.42	20.97	11.97	-

Table 6: Precision, Recall, and F1 scores of different models on detecting friction. #Predictions refer to the total number of instances of conversational friction found by each model. For reference, annotators identified **238** instances in total. gpt-4o *without* Elaboration of technical terms (Sec 3.1) performed best across all models.

Model	Success Prediction (Spearman's ρ)	Binary Friction Presence (Cohen's κ)
gpt-4o	**0.776**	**0.380**
gpt-4o w/ Elab.	0.743	0.310
gpt-4o-mini	0.699	0.205
gpt-4o-mini w/ Elab.	0.634	0.205
Llama-3.1-8b-Instruct	0.261	0.193
Llama-3.1-8b-Instruct w/ Elab.	0.235	-0.249
Llama-3.1-70b-Instruct	0.702	0.290
Llama-3.1-70b-Instruct w/ Elab.	0.630	0.223

Table 7: Spearman's ρ and Cohen's κ for the related tasks of predicting success friction presence. Models align more with humans on the success of a conversation.

grounding acts. To understand whether models can make binary judgments as to whether or not friction is present without identifying their location, we prompt models to predict the presence of friction *without* pinpointing specific dialog turns. This allows us to assess the model's ability to predict friction as a broader phenomenon. We also evaluate the capability of models to predict the *success* of the task undertaken in the conversation on a three-point scale, as in §3.1.

We evaluate the binary prediction task with Cohen's κ, framing it as inter-rater agreement between models and humans. Models' over-prediction of friction intervals persists in the conversation level as well (Table 7). Predictions on task success, on the other hand, is highly correlated with annotator ratings of success.

6 Error Analysis

We now investigate the successes and failures of gpt-4o, the strongest performing model at this task.

Undetected frictions are deeper in conversations. As a conversation proceeds, detecting friction requires a deeper understanding of preceding turns. To explore whether the *position* of friction impacts model accuracy, we stratify our results by conversational depth and calculate the relative depth of each instance of friction as the ratio of the first turn of the friction interval to the conversation length multiplied by 100. The mean relative depth of a detected instance of friction (35.19) is significantly smaller than the mean relative depth of a detected instance (49.62), according to an independent t-test ($p < 0.01$). This indicates that models struggle with taking a longer context into account while determining whether participants' versions of common ground are misaligned.

Implicit cases of friction are harder to detect. Models, particularly gpt-4o, are more likely to correctly identify friction when an explicit request for conversational repair is present. Specifically, 77.22% of detected frictions involved an explicit RequestRepair, compared to 64.81% of frictions that went undetected ($p < 0.05$). This highlights the tendency of models to rely on overt cues that signal a common ground misalignment.

Consider the conversation in Table 8. A's utterance *"how about nmap"* (Turn 22) is not introducing nmap as an option, but following up on B's earlier suggestion in Turn 21 by asking *how* nmap can be used to solve the issue. B reveals that they did not understand this interpretation through their response in Turn 23 (*"yeah, i said nmap,"*), prompting A to issue a Repair act. We hypothesize that this unconventional way of issuing a Repair (through a question) without an explicit RequestRepair results in an undetected conversational friction.

Turn	Speaker	Utterance
16	B (helper)	btw, you do need to restart the ssh server for it to work on the new ip(s)
17	A (asker)	sudo service ssh restart?
18	B (helper)	yeah
19	A (asker)	is the service ssh or anything else?
20	B (helper)	yep thats the service
21	B (helper)	and you can check if its listening with nmap
22	A (asker)	how about nmap?
23	B (helper)	yeah, i said nmap
24	A (asker)	I mean how do I use nmap to find that out?

Table 8: A conversation showing an undetected case of friction, where a Repair act is expressed through a question (Turn 24). B misinterprets A's question in Turn 22 as a suggestion, while, as revealed in Turn 24, A was simply following up on B's early suggestion of using nmap from Turn 21.

Comparing Model and Human Explanations. Collecting model explanations along with friction interval predictions allows us to evaluate whether they accurately capture the cause of friction. In most cases, this amounts to correctly pointing out the cause of misalignment in participants' respective versions of the CG. To study this, we ask two non-author Linux experts to annotate the similarity between gold friction explanations and model-generated explanations for 64 instances of friction on a three-point scale, marking them as either (1) dissimilar, (2) somewhat similar, or (3) equivalent. Agreement between the two annotators using Spearman correlation is $\rho = 0.61$, with $p < 0.01$.

Most model explanations accurately captured the cause of friction, with 57.81% instances annotated as equivalent and 34.37% as somewhat similar. Only 7.8% explanations did not point out the cause of friction at all. Echoing our earlier findings, models struggle to pinpoint the cause of friction *even when they identified the window correctly*. For example, in Figure 2, Row 1, A mistakenly assumes that B's suggestion for a command (Turn 25) includes the keyword "try", leading to the error "try command not found" (Turn 32) after which B repeats the dmesg command removing "try" at the beginning. LLMs must be able to pinpoint causes of friction to issue repairs that address the friction directly, a crucial ability in settings where LLMs are used for dispute resolution (Tan et al., 2024).

7 Related Work

The speech-act based approach in Traum and Allen (1992) has been used to study cooperative grounding acts in the Meetup (Ilinykh et al., 2019) and Spot the Difference (Lopes et al., 2018) datasets (Mohapatra et al., 2024). While conversations in such scenarios also require grounding, both datasets involve conversational participants interacting in a *physical* setting. Because of the additional modality, the mutually shared basis of their CG (e.g. an object both or one participant can see) is not available to the reader, making it difficult to capture what causes friction from text alone.

Markowska et al. (2023) track speaker versions of the CG through speaker "beliefs" expressed in conversations in the LDC Callhome (Canavan et al., 1997) corpus. However, participants are not as incentivized to build and maintain a rich CG since the conversations are not goal-driven, and are between close friends or family. Khebour et al. (2024) annotate a task-oriented corpus for multi-modal features and dialogue moves to model shared beliefs and questions under discussion. The authors train LSTM-based classifiers of dialogue moves relevant to tracking CG, finding that utterances may or may not be aligned with other modalities such as gesture. This highlights the challenge of tracking common ground in physically situated dialogue; our dataset simplifies the focus to text alone.

Shaikh et al. (2024) use grounding acts to compare the degree of grounding by LLMs in human-LLM conversations, finding that LLMs perform less computational grounding. Our work complements these directions by focusing on whether LLMs can *detect* when and how participants in a conversation might lose track of CG. In more recent work, Shaikh et al. (2025) focuses on this divergence across several grounding acts, and introduces an annotated dataset of LLMs failing to ground in human-LLM conversations.

Also related to our work is the concept of positive friction as explored in İnan et al. (2025). Here, the authors view "friction" not as a misalignment in common ground that detracts from the goal of the conversation, but as a series of communicative "movements" such as pausing, revealing speaker assumptions, etc., that facilitate long-term success over short-term progress in a conversation.

8 Conclusion and Future Work

In this case study, we have conducted what is to our knowledge the first investigation of friction and repair of CG for task-oriented dialogue in a real-world, text-only setting. Our qualitative and quantitative results reveal that friction in goal-oriented

dialog is inevitable, and it takes effort from *both* participants to repair the CG to make progress towards a task. Keeping track of CG over text is no easy feat—it requires participants to be vigilant about implicit cues in text that might signal a potential misalignment. While some `helpers` in our dataset anticipated and prevented potential friction or issued `Repair` acts once friction did happen, LLMs such as `gpt-4o` struggled with detecting and explaining cases of friction in the absence of explicit evidence.

As LLMs are deployed in settings such as education (Wang et al., 2024), future work might explore improving their ability to understand *implicit* ruptures in CG—an LLM tasked with analyzing conversations between a student and teacher should be able to detect the loss of common ground for better learning outcomes. Another future direction involves explicitly *modeling* the CG, consisting of propositions that are part of participants' underlying mental state. Recent work has demonstrated that LLMs draw plausible inferences about such propositions in non-conversational settings (Hoyle et al., 2023). Conceptually, thought bubbles such as those illustrated in Figure 1 could be populated automatically, operationalizing the detection of CG misalignments by similarity-based comparison and contrast of participants' individual belief spaces.

Limitations

Our study takes an important step towards quantifying the role of grounding in goal-oriented dialog and studying LLM capabilities of detecting friction. Unlike studies that simulate conversations between participants in artificial settings to gain access to their mental states and the common ground, we do not have access to conversational participants' common ground or mental states beyond what is expressed in the text conversation. In addition, we do not have access to the degree of self-effort that goes into solving an issue alongside a conversation—the `asker` might simultaneously have been searching the internet for answers while engaged in conversation.

Another limitation of our work stems from the fact that we limit our analyses of LLMs to their role as an observer of human-human conversation, and not as a participant. Given that LLMs perform differently in linguistic tasks (such as responding to a query) as opposed to metalinguistic tasks (spotting a mismatch in common ground), it is possible that a model *response* addresses a mismatch in common ground indirectly as a conversational partner while failing to identify the cause of friction as an observer (Hu and Frank, 2024). However, we believe that understanding LLM behavior of tracking common ground is an essential prerequisite to many other downstream research questions, such as cases where an LLM is used as a conversational facilitator (Argyle et al., 2023).

Although the conversations take place purely through text, participants sometimes shared links to blog posts and tutorials, many of which now no longer work. In rare cases, it might be possible that the cause (or resolution) of a friction instance is rooted in such a link. We also do not have access to their screens or other metadata about the user that might have been instrumental in resolving friction.

Acknowledgments

We thank our anonymous reviewers for their comments and suggestions throughout the reviewing process. We also thank Atrey Desai, Nyle Masood, Ayush Kumar, Maya Srikanth, and Pratheek Hegde for providing valuable annotations. This work was supported in part by the US National Science Foundation award 2124270. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

References

L.P. Argyle, C.A. Bail, E.C. Busby, J.R. Gubler, T. Howe, C. Rytting, T. Sorensen, and D. Wingate. 2023. Leveraging ai for democratic discourse: Chat interventions can improve online political conversations at scale. *Proceedings of the National Academy of Sciences of the United States of America*, 120(41):e2311627120.

Cristian-Paul Bara, Sky CH-Wang, and Joyce Chai. 2021. MindCraft: Theory of mind modeling for situated dialogue in collaborative tasks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1112–1125, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dan Bohus and Alexander I Rudnicky. 2008. Sorry, i didn't catch that! In *Recent trends in discourse and dialogue*, pages 123–154. Springer.

Claire Bonial, Taylor Hudson, Anthony L Baker, Stephanie M Lukin, and David Traum. 2022. Making sense of stop. In *AREA II Workshop 2022*.

Alexandra Canavan, David Graff, and George Zipperlen. 1997. Callhome american english speech ldc97s42. Web Download.

Santiago Castro. 2017. Fast Krippendorff: Fast computation of Krippendorff's alpha agreement measure. https://github.com/pln-fing-udelar/fast-krippendorff.

Khyathi Raghavi Chandu, Yonatan Bisk, and Alan W Black. 2021. Grounding'grounding'in nlp. *arXiv preprint arXiv:2106.02192*.

Herbert H. Clark and Susan E. Brennan. 1991. Grounding in communication. In Lauren B. Resnick, John M. Levine, and Stephanie D. Teasley, editors, *Perspectives on Socially Shared Cognition*, pages 127–149. APA Books, Washington, DC.

Herbert H. Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22(1):1–39.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Vanya Cohen, Jason Xinyu Liu, Raymond Mooney, Stefanie Tellex, and David Watkins. 2024. A survey of robotic language grounding: Tradeoffs between symbols and embeddings. *arXiv preprint arXiv:2405.13245*.

Bart Geurts. 2024. Common Ground in Pragmatics. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*, Winter 2024 edition. Metaphysics Research Lab, Stanford University.

Barbara J Grosz and Candace L Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational linguistics*, 12(3):175–204.

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

Peter Heeman and James Allen. 1994. Detecting and correcting speech repairs. *arXiv preprint cmp-lg/9406006*.

Alexander Hoyle, Rupak Sarkar, Pranav Goel, and Philip Resnik. 2023. Natural language decompositions of implicit content enable better text representations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13188–13214, Singapore. Association for Computational Linguistics.

Jennifer Hu and Michael C. Frank. 2024. Auxiliary task demands mask the capabilities of smaller language models. *Preprint*, arXiv:2404.02418.

Nikolai Ilinykh, Sina Zarrieß, and David Schlangen. 2019. Meetup! a corpus of joint activity dialogues in a visual environment. *arXiv preprint arXiv:1907.05084*.

Ibrahim Khalil Khebour, Kenneth Lai, Mariah Bradford, Yifan Zhu, Richard A. Brutti, Christopher Tam, Jingxuan Tu, Benjamin A. Ibarra, Nathaniel Blanchard, Nikhil Krishnaswamy, and James Pustejovsky. 2024. Common ground tracking in multimodal dialogue. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3587–3602, Torino, Italia. ELRA and ICCL.

Jonathan K. Kummerfeld, Sai R. Gouravajhala, Joseph J. Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros C Polymenakos, and Walter Lasecki. 2019. A large-scale corpus for conversation disentanglement. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3846–3856, Florence, Italy. Association for Computational Linguistics.

Willem JM Levelt. 1983. Monitoring and self-repair in speech. *Cognition*, 14(1):41–104.

José Lopes, Nils Hemmingsson, and Oliver Åstrand. 2018. The spot the difference corpus: a multi-modal corpus of spontaneous task oriented spoken interactions. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294, Prague, Czech Republic. Association for Computational Linguistics.

Magdalena Markowska, Mohammad Taghizadeh, Adil Soubki, Seyed Mirroshandel, and Owen Rambow. 2023. Finding common ground: Annotating and predicting common ground in spoken conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8221–8233, Singapore. Association for Computational Linguistics.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *Preprint*, arXiv:2402.06196.

Biswesh Mohapatra, Seemab Hassan, Laurent Romary, and Justine Cassell. 2024. Conversational grounding: Annotation and analysis of grounding acts and grounding units. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3967–3977, Torino, Italia. ELRA and ICCL.

Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. 2019. Collaborative dialogue in Minecraft. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*,

pages 5405–5415, Florence, Italy. Association for Computational Linguistics.

OpenAI. 2024. Hello gpt-4o: A new model for openai's future. Accessed: 2024-10-13.

Katrin Ortmann. 2022. Fine-grained error analysis and fair evaluation of labeled spans. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1400–1407, Marseille, France. European Language Resources Association.

Sanjana Ramprasad, Elisa Ferracane, and Zachary C. Lipton. 2024. Analyzing llm behavior in dialogue summarization: Unveiling circumstantial hallucination trends. *Preprint*, arXiv:2406.03487.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Omar Shaikh, Kristina Gligoric, Ashna Khetan, Matthias Gerstgrasser, Diyi Yang, and Dan Jurafsky. 2024. Grounding gaps in language model generations. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6279–6296, Mexico City, Mexico. Association for Computational Linguistics.

Omar Shaikh, Hussein Mozannar, Gagan Bansal, Adam Fourney, and Eric Horvitz. 2025. Navigating rifts in human-llm grounding: Study and benchmark. *Preprint*, arXiv:2503.13975.

Robert Stalnaker. 1978. Assertion. *Syntax and Semantics*, 9:315–332.

Robert Stalnaker. 2002. Common ground. *Linguistics and Philosophy*, 25(5/6):701–721. Accessed: 2019-02-15 07:19 UTC.

Jinzhe Tan, Hannes Westermann, Nikhil Reddy Pottanigari, Jaromír Šavelka, Sébastien Meeùs, Mia Godet, and Karim Benyekhlef. 2024. Robots in the middle: Evaluating llms in dispute resolution. *Preprint*, arXiv:2410.07053.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

David R. Traum and James F. Allen. 1992. A "speech acts" approach to grounding in conversation. In *The Second International Conference on Spoken Language Processing, ICSLP 1992, Banff, Alberta, Canada, October 13-16, 1992*, pages 137–140. ISCA.

David Rood Traum. 1995. *A Computational Theory of Grounding in Natural Language Conversation*. Ph.D. thesis, University of the USA. UMI Order No. GAX95-23171.

Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S. Yu, and Qingsong Wen. 2024. Large language models for education: A survey and outlook. *Preprint*, arXiv:2403.18105.

Aaron Steven White, Pushpendre Rastogi, Kevin Duh, and Benjamin Van Durme. 2017. Inference is everything: Recasting semantic resources into a unified evaluation framework. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 996–1005, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Mert İnan, Anthony Sicilia, Suvodip Dey, Vardhan Dongre, Tejas Srinivasan, Jesse Thomason, Gökhan Tür, Dilek Hakkani-Tür, and Malihe Alikhani. 2025. Better slow than sorry: Introducing positive friction for reliable dialogue systems. *Preprint*, arXiv:2501.17348.

A Appendix

A.1 Further Annotation Details

In our dataset of 200 conversations, we divide annotations into 10 batches, each containing 20 conversations. Three of those batches (60 conversations) were annotated by all annotators, going towards computing the inter-rater agreement scores seen in Table 4. The rest of the conversations were annotated by a single annotator. For the conversations annotated by all three annotators, we picked annotations from either A1 or A2, since they had stronger inter-rater agreement. Overall, out of the 10 batches, 4 batches were annotated by A1, 4 were annotated by A2, and 2 were annotated by A3.

The model scores and the inter-rater agreement values are calculated using the same metrics, friction-found and friction-overlap 3.2. They are comparable with the caveat that results in Table 6 are unidirectional (models against gold data), and those in Table 4 are made two-sided (by averaging A_i against A_j and A_j against A_i). We note that while annotator agreement numbers are modest, the best inter-rater agreement between our annotators (A1 and A2) is significantly higher than any of our considered models. These two particular annotators (A1 and A2) annotated 80% of our dataset.

A.2 Model Finetuning Details

For our finetuning experiments, we create five random train-test splits of our dataset, with 30% of conversations in Ubuntu-CG in each split going towards the "test" set. For each of our context windows ($k = 3$ or $k = 5$ turns), we train a `distilroberta-base` model with 82M parameters for 15 epochs with a learning rate of $4e-5$. As evaluation data, we pick a single fold and isolate a part of its training set as our development data. Since this is a class-imbalanced dataset (turns not containing friction greatly outnumber turns containing friction 10 to 1), we artificially reduce the number of negative samples in our *training* data, however resampling was not done on the test data. We also experimented with larger encoder-only models, but they all performed much poorer than `distilroberta` under both full-parameter and classification head-only training.

Relationship with NLI. The setup we choose for our finetuned baselines may resemble that of classical textual entailment, where, given a premise sentence, a model must determine whether a hypothesis sentence is entailed, contradicted, or neutral in relation to the premise. Friction detection in this baseline setting *could* be *recast* as an NLI-style task (White et al., 2017). We do not experiment with existing finetuned NLI models since, as is, our task in the baseline setting does not cleanly or directly map to the task of textual entailment in its current form, and hence, would produce unreliable labels.

A.3 Computational Details

The `Llama-3.1-70b-Instruct` models were used with 4bit quantization to fit on two A6000 GPUs.

A.4 Prompts

We outline all prompts used in the paper below. In the interest of presentation, they are broken into modules. For example, Prompt A.1 and Prompt A.2 would combine to form a single prompt for friction detection, and Prompt A.3 is plugged in the middle to make use of `gpt-4o`-generated explanations.

Prompt A.1: Friction Detection Prompt

Prompt: ### TASK DESCRIPTION: Detecting "Conversational Friction" in Online Conversations.

Given a conversation between two participants in an online chat forum, label one or more turns in the conversation where there is evidence of friction between the two participants, that is, where they don't seem to fully understand each other or seem to not be on the same page. This friction could be due to a mismatch between their goals, due to a false assumption one participant made about the other leading to a misunderstanding, and so on. These may result from a mismatch in the common ground between the two participants.

A strong indicator of conversational friction could be a participant asking the other participant to revisit or clarify previously shared content in the conversation, in a process known as conversational repair. However, in many cases there may not be an explicit Repair Request issued by a participant but from context it can be reasoned that a participant is struggling to keep up with the conversation. In some cases, it becomes apparent that a participant was requesting conversational repair in a turn only after reading through subsequent turns. In that case, go back and annotate that turn as friction.

Note that possible friction can occur in a single turn (in which case, mark that specific turn), or through a series of turns (in which case, mark the window of turns that all together add up to a repair request). In each of these cases, you should mark the turn(s) where the friction is most apparent. Also write a brief explanation of why you think that turn is an instance of conversational friction as defined above.

Prompt A.2: Input/Output Format

Prompt: ### INPUT:
Conversation: {convo_text}

Now, follow the output format below to annotate the conversation.

OUTPUT FORMAT:

First output the turns showing conversational friction in a dictionary. If there is more than one instance of friction, list them in the order they appear in the conversation. If there's no friction in the conversation, set "friction_present" to false and don't provide any other fields.
Follow the output format below to annotate the conversation.

```
{{
"friction_present": [Choose true or false], # if false, stop here
"friction1": [X, Y], # the start and end turns of the first instance of friction
"explanation1": "Brief explanation for friction1",
"friction2": [X, Y], # If there is more than one instance of friction
"explanation2": "Brief explanation for friction2"
....
}}
```

Prompt A.3: Adding Explanations

Prompt: ### EXPLANATIONS

To clarify the many technical terms used in the conversations, you are also provided an explanation of terms used in a particular turn at the end of the turn. This explanation is provided in the format: Turn X Explanation: <Explanation of the terms used in Turn X>. In general, the format of the conversation is as follows:

[Turn 0] User A: <Message about current current issue with linux>
Turn 0 Explanation: <Contextual explanation of the technical terms used in the conversation>
[Turn 1] User B: <Response to Turn 0>
Turn 1 Explanation: <Contextual explanation of the technical terms used in the conversation>
...
...

NOTE: In addition to the conversation, optionally use the explanations provided to better understand what's going on in the conversation. Discard the explanations if you feel they are not necessary.

Prompt A.4: Success Prediction

```
Prompt: ### TASK DESCRIPTION
You will be given a conversation between two participants A (usually the **user** seeking help) and B (usually the **helper**) who are trying to solve an issue in Ubuntu together on the #Ubuntu IRC channel. Your task is to determine how successful the conversation was towards resolving the issue of the user.

Mark how helpful the conversation was to whoever was asking for help on a scale of 1-3, where each number on the scale has the following meaning:

- 1 (NO PROGRESS): This indicates that the conversation was not helpful to A at all in resolving their issue, and they did not make any progress towards solving the problem.
- 2 (SOME PROGRESS): This indicates that the participants made some progress towards solving the problem. They might not have resolved the issue entirely, but they made progress in diagnosing the problem or solved a part of the problem.
- 3 (SUCCESS): This indicates that the participants solved the problem they initially set out to solve, or the problem that evolved in the course of the conversation.

The scores hold true even if they themselves realize the issue in the course of the conversation and proceed to solve it. It also holds true even if the conversation went off-topic, as long as the participants were able to solve the problem at hand.

NOTE: The problem that A starts the conversation with might not be the right problem to solve at all, and the helper (usually B) might suggest what the right issue to solve is. In that case, solving the re-defined problem will decide conversational success on this scale.

### INPUT

Conversation:

{convo_text}

### OUTPUT

First, provide the success score for the conversation on a scale of 1-3. Then, provide a brief explanation explaining the score in the format below:
{{
"success_score": [1/2/3] # 1 for NO PROGRESS, 2 for SOME PROGRESS, 3 for SUCCESS. Output score only
"explanation": "Brief explanation for the success score"
}}
```

Prompt A.5: Binary Friction Detection

```
Prompt: ### TASK DESCRIPTION: Detecting "Conversational Friction" in Online Conversations

Given a conversation between two participants in an online chat forum, output whether there is evidence of conversational friction between the two participants. Conversational friction occurs when participants in a conversation don't seem to fully understand each other or seem to not be on the same page. This friction could be due to a mismatch between their goals, due to a false assumption one participant made about the other, leading to a misunderstanding, and so on. These may result from a mismatch in the common ground between the two participants.

A strong indicator of conversational friction could be a participant asking the other participant to revisit or clarify previously shared content in the conversation, in a process known as conversational repair. However, in many cases, there may not be an explicit Repair Request issued by a participant, but from the context, it can be reasoned that a participant is struggling to keep up with the conversation.

NOTE: Friction is often signaled by the helpee asking a follow-up question. However, not all follow-up questions indicate that the speakers are not on the same page. For example, clarification questions that ask for information not assumed by either user to be in the common ground are not cases of conversational friction. Clarification questions that move the conversation forward without questioning the common ground are not cases of conversational friction. If there is **no conversational friction** make sure to indicate that in the output by setting "friction_present" to false.

### TASK:

Given a conversation, list whether conversational friction occurs or not.
```

A.5 Elaborations

Examples of elaborations can be found in Table 9.

A.6 Annotator Instructions

Before any annotation task, annotators had to fill out a consent form (Figure 3). To ensure we're measuring equivalent constructs, the annotator instructions was kept identical to Prompt A.1. A more detailed instruction document can be found in the supplementary material. The similarity scoring prompt is shown in Figure 4.

A.7 Breakdown of Table 5

Utterance	GPT Elaboration	Year
hi, i have ubuntu dapper and want to do a clean install of feisty using the live cd (I want to put feisty in my current ext3 partition and format ext3). When the installation process comes to the part about partitioning, (Erase hard disk, automatic, or manual), should I choose manual and if so, will there be a way to format ext3 and will it allow me to put feisty in my current ext3 partition without making a new	Ubuntu Dapper and Feisty are code names for older versions of the Ubuntu operating system, specifically 6.06 (Dapper Drake) and 7.04 (Feisty Fawn), respectively. A 'live CD' allows you to run Ubuntu directly from the CD without installing it on your hard drive. 'ext3' is a type of file system used in Linux for organizing and storing files on a partition.	2005
does passwords and encryption keys support hkps?	"HKPS" stands for HTTP Keyserver Protocol Secure. It is a secure version of the HTTP Keyserver Protocol (HKP) used to retrieve encryption keys from a keyserver over a secure, encrypted connection. In the context of Ubuntu or other operating systems, this might refer to the secure retrieval or management of encryption keys, potentially in relation to applications or services that require encryption.	2010

Table 9: Explanation of technical terms present in dialog turns explained by GPT4. These help our annotators understand terms such as "khps", "dapper", or "feisty".

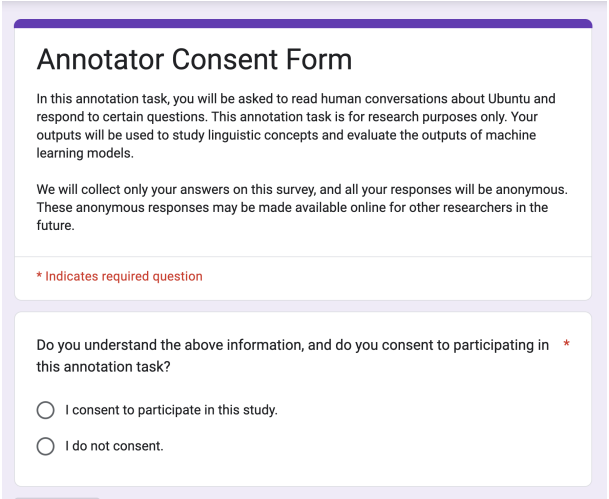

Figure 3: The consent form shown to annotators before each task.

Degree of Progress	#Convs	Instances (Repair/ReqRepair)	Unaddressed ReqRepair (%)
3	23	45 (35/33)	27.27
2	26	57 (48/42)	19.05
No Progress (1)	21	50 (38/36)	30.56

Table 10: Full breakdown of summary of success and grounding acts in our analysis subset of 70 conversations. In conversations with no progress, more requests for repairs go unaddressed.

Conversational Friction: In conversations, conversational friction denotes a disruption in communicative flow caused by a misalignment in the speakers' perceived versions of common ground, including their knowledge, beliefs or goals. You are given a conversation between two participants who are trying to solve an issue that revolves around the Ubuntu operating system along with instances of conversational friction tagged from two sources.

Scoring Similarities
Try to assign a similarity score between the two explanations, again on a three point scale, where

- 1 indicates that the two explanations are not similar at all - they are describing different reasons for the friction
- 2 indicates that the two explanations are somewhat similar, or are talking about related issues that may have caused conversational friction
- 3 indicates that the two explanations are similar, and are pointing to the same reason for conversational friction.

Assign a score from 1-3 on the basis of how similar the explanations are.

NOTES:

- While giving a score for similarity of explanations, also try to focus on **content** rather than **presentation**. Two explanations might point to the same reason for friction in different ways, in which case they still should receive a higher score.

- Given a window of friction, explanations might be scattered throughout turns or might be summarized next to a single turn - you should treat both of these cases similarly, considering explanations written for the **entire window**.

Figure 4: Instructions provided to the annotators for judging the similarity of `gpt-4o`and human-generated explanations for frictions. The annotators did not know the source of an explanation.