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

Anonymous ACL submission

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. We also find that although Large Language Models (LLMs) can identify overt cases of conversational friction, they struggle with subtler and more context-dependent instances requiring 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 in order 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 constituted 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?

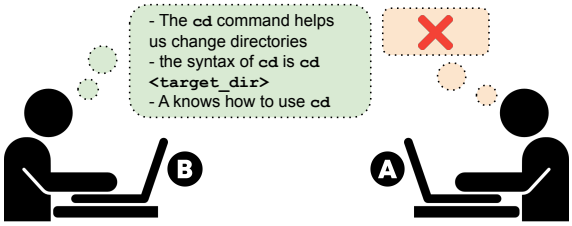

Figure 1: An instance of conversational friction. Though it is challenging to access propositions in a speakers' perception of common ground, certain propositions in B's version of common ground are revealed (green thought bubble) when there is a misalignment between the two participants. B assumes A 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. It later emerges in Turn 9 that this assumption was incorrect via an *observable* interruption precluding A and B from proceeding towards the main

conversational goal of A.[1]

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.

We explore 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 study real-world conversations involving Ubuntu users attempting to fix an issue or bug which share important properties with other real-world conversations. 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.[3] Each conversation is annotated for the presence of conversational friction and the degree of task success (§3.1) to analyze the importance of maintaining common ground (§4). Then, we explore the ability of LLMs to predict instances of conversational friction and compare their explanations with human explanations (§5).

Not only are LLMs are 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, a 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.

	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* 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)[4]. Subsequently, Clark and Brennan (1991) define common ground as a collection of mutual knowledge, beliefs, and assumptions that humans build and maintain collaboratively through the process of *grounding*.[5]

In early computational work studying the common ground, Traum and Allen (1992) propose 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 a presented utterance.[6]

While it is acknowledged that maintaining common ground is of some importance to conversational success (Traum, 1995), there has been little empirical work that explicitly ties participant effort in maintaining common ground to the success towards an end goal. In this study, we look at the importance of grounding in the success of naturally-

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

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

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

[4] 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).

[5] 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).

[6] See Table 1 of Traum and Allen (1992) for an exhaustive list.

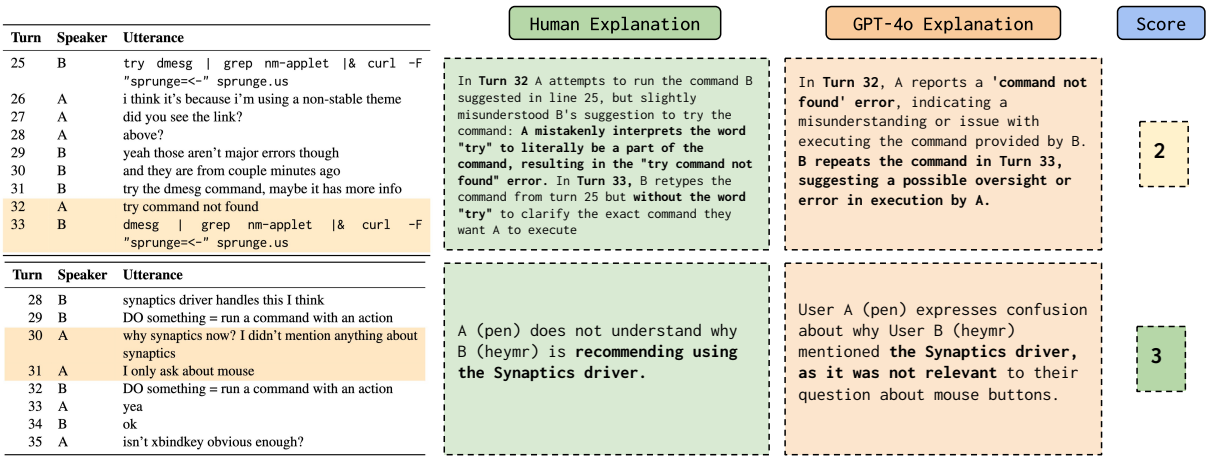

Figure 2: Comparing GPT4o and human explanations for the cause of friction. GPT4o explanations align with humans when friction is explicit (row 2). In a more implicit case of friction (row 1), GPT4o fails to capture the true reason for friction—A misreading "try" as part of a terminal command (Turn 25),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

occurring goal-oriented conversations. **Specifically, we focus on conversational friction as evidence of the loss and re-negotiation of common ground.**

In a typical conversation in our dataset (such as the one in Table 3), two participants (the `asker` and the `helper`) try to collaboratively solve a Linux bug over a text channel. This consists of several communicative steps—the `asker` must describe the issue they are facing in Ubuntu (often with insufficient knowledge of Linux), and the `helper` must understand their goal to propose a solution. This conversational setting is well-positioned for studying friction and grounding.

2.1 Dataset: Ubuntu-CG

The Ubuntu Dialog Corpus satisfies several essential criteria for our study; (1) conversations are *naturally* goal-oriented (e.g., resolving a bug or an error in Ubuntu), resulting in a significant incentive for participants to communicate effectively; (2) participants have to establish common ground from scratch; (3) conversations are only through the medium of text; and (4) are multi-turn, ranging from three turns to over one hundred, thereby giving users ample time to build and utilize common ground. Some of the other datasets we considered are discussed in Section 7.

The Ubuntu Dialog Corpus (Lowe et al., 2015) contains conversations scraped from the `#Ubuntu` IRC channel, where users discuss features, issues and bugs related to the Ubuntu operating system, among other things. Extracting conversations from the IRC channel requires disentangling conversations from a single stream of messages. While the original corpus used a simple disentanglement strategy, Kummerfeld et al. (2019) found that 80% of the conversations were missing messages or contained added messages. We use a sample of 200 two-person conversations from the cleaned corpus released by Kummerfeld et al. (2019) for our study, 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)	33.05 (25.84)	64.40 (38/59)	2.39
2 (Some Progress)	44.22 (25.49)	64.19 (52/81)	2.08
3 (Success)	40.3 (28.56)	53.33 (32/60)	2.09

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 Approach

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, only over text.[7] This setting naturally lends itself to frequent occurrence of conversational friction. But how often is friction resolved in subsequent grounding, and does it have

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

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.

	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).

a demonstrable effect on the success of a conversation? To answer these questions, annotators familiar with Linux mark intervals of dialogue turns where there are instances of friction, and mark the overall conversation on a three-point success scale.

3.1 Annotating for Conversational Friction

Three computer science undergraduates familiar with Linux were paid $18/hr to annotate 200 conversations totaling 7950 turns, taking over 80 hours to complete. Annotators mark dialog turns containing evidence of friction, adding an explanation justifying their decision. Since conversations date back to over a decade ago, they often contain antiquated terms or reference which 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) 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.

3.2 Measuring Agreement

Measuring inter-rater agreement in our setting is not straightforward, as we need to account for agreement both in identifying an instance of friction *and* the turn interval in which it occurs. To simplify this measurement, we compute overlap metrics for each pair of annotator, 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 compute two F1 scores—once treat annotations from A_1 as ground truth and those from A_2 as predictions and vice versa.[8] Agreement is then the average of these two F1 scores. We compute agreement in two different settings.

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

Friction Overlap. We introduce a second setting called "Friction Overlap" which rewards the *degree* of overlap with the gold interval. We first match each instance of friction with the predicted instance with the highest overlap. Unlike the previous setting, this ensures a one to one mapping between a predicted friction interval and a gold one. For each *matched* interval, we compute the Jaccard similarity between the two intervals. This setting assigns a higher score to predictions that better align with human-annotated instances of friction. A perfect score indicates that predicted intervals exactly overlapped with gold intervals. In practice, this penalizes predicting multiple short or overtly long instances.[9] We use these same two settings to compute model performance (Table 6). Table 4 shows agreement between pairs of annotators.

Task Success. In addition to friction, annotators assess how successful participants were in solving the issue at hand. Each conversation was rated on a three-point scale of task success. 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 overall statistics, and the instructions for friction and success annotation can be found in the Appendix A.3. We obtain an agreement of

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

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

$\alpha = 0.58$ on task success annotation as measured by Krippendorff's Alpha (Castro, 2017).

3.3 Annotating for Grounding Acts

Our annotations reveal that successful conversations contain less friction (Table 2). However, when friction is present, can participants collaboratively rebuild common ground to complete tasks successfully? We annotate turns corresponding to friction for two essential 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 shows a typical example of these two acts in play.

Identifying these acts not only helps us determine whether participants were able to recover from friction, but also enables us to study the ability of models to detect friction in greater detail. 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 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 of 70 conversations has a higher average length than our overall dataset. 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 association between the presence of conversational friction and the success of a goal-driven conversation in Ubuntu-CG. We present our principal findings from the data below.

Successful conversations contain less friction. In Ubuntu-CG, 61% percent of conversations contained instances of conversational friction. In contrast, in conversations where the `helper` succeeded in solving the `asker`'s issue (receiving a score of 3), only 53.33% contained friction (Table 2). Conversations where participants at least make some progress or succeed in the task contain less friction on average in contrast to conversations where they did not make *any* progress, as the former exhibits some amount of grounding efforts by the participants (Table 2, Column 4). This is further supported by the proportion of unaddressed repair efforts reported in Column 4 of Table 5.

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.

Friction is more likely in longer conversations. While no clear pattern emerges in how length of the conversation varies with task success (Table 2), the mean length of a conversation containing friction is 49 (median 50), while that of conversations without friction is 20.08 (median 12). Comparing these with the overall mean length of the dataset 40.56 (median 33), it is evident that conversational friction and subsequent repair through the process of grounding contributes to the increased number of turns it takes to complete the conversation.

4.1 Role of Grounding Acts in Task Success

Conversations that receive a success score of 1 (No Progress) are characteristically different from conversations receiving a score of 2 or 3. The former are cases where participants could not make any progress towards diagnosing a particular issue. 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 common ground. We evaluate whether models can identify and explain instances of conversational friction in Ubunutu-CG. We prompt several proprietary and open-source models to identify intervals of dialog turns where there's evidence of friction, and provide brief explanations for the cause of friction (results in Table 6). Models labeled "w Elab." are provided access to the elaborations of technical terminology as outlined in Section 3.1.

5.1 Experimental Setup

Evaluation Metrics. Models are prompted to predict the number of instances of friction in a conversation, along with corresponding dialog turns. We evaluate LLM output 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.

Models predict intervals of friction with Prompt A.1 on full conversations as input. For all experimental settings, we set temperature to 0.01. The `Llama-3.1-70b-Instruct` models were used with 4bit quantization to fit on two A6000 GPUs.[10] All prompts can be found in the Appendix.

5.2 Results

Under both evaluation settings, `gpt-4o` or `gpt-4o` with elaborations obtained the highest F1 score. `llama` models have the highest recall, which is balanced by their low precision. Under the stricter "span" setting, `gpt-4o` with elaboration had 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).

[10]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.

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 **(Friction Found)**, adding elaborations do not seem to improve prediction scores of models, though **Friction Overlap** yields stronger performance across all models except `Llama-3.1-70b-Instruct`. For `Llama-3.1-8b-Instruct` and `gpt-4o-mini`, adding elaborations improves both precision *and* recall. 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 grounding acts. To understand whether models can make a 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 §3.2.

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 persist 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 focus on better understanding the successes and failures of our best performing setting (`gpt-4o` with elaborations).

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. We 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

Model	Friction Found			Friction Overlap			#Predictions
	Precision	Recall	F1	Precision	Recall	F1	
gpt-4o	40.10	71.43	**51.36**	17.00	30.28	21.77	495
gpt-4o w/ Elab.	**41.83**	65.18	50.96	**18.03**	28.09	**21.96**	435
gpt-4o-mini	38.37	44.20	41.08	16.36	18.84	17.51	316
gpt-4o-mini w/ Elab.	32.72	47.77	38.84	16.65	24.30	19.76	392
Llama-3.1-8b-Instruct	17.94	81.70	29.42	7.08	32.25	11.62	1282
Llama-3.1-8b-Instruct w/ Elab.	19.08	**87.05**	31.30	7.58	**34.60**	12.44	1253
Llama-3.1-70b-Instruct	26.55	82.14	40.13	10.38	32.12	15.69	857
Llama-3.1-70b-Instruct w/ Elab.	21.35	75.00	33.23	8.81	30.94	13.71	959

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 **266** instances in total. gpt-4o with 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.

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 between 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.

with taking 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 with elaboration, are more likely to correctly identify an instance of 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 from participants that signal a common ground misalignment.

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

6.1 Explanations

Collecting model explanations along with detected windows of friction allows us to understand if these explanations accurately capture the cause of friction. In most cases, this amounts to correctly pointing out the cause of misalignment in participants' respective versions of common ground. To study this, we collect similarity scores on a 1–3 model-detected friction, where 1 indicates that the model explanations are not similar to the human explanations at all, 2 indicates they are somewhat similar, and 3 indicates that they point to the exact same cause for friction. Two non-author annotators who are experts in Linux annotate 64 samples on this three-point scale (Spearman's $\rho = 0.61$, with $p < 0.01$).

Most explanations accurately captured the cause of friction, receiving scores of 3 (57.81%) and 2 (34.37%). Only about 7.8% explanations did not point out the cause of friction at all. However, as we found previously, models struggle to pinpoint the cause of friction *even when they identified the window correctly* (Figure 2, Row 1). A mistakenly

assumes that B's suggestion for a command in Turn 25 includes the keyword "try", leading to the error "try command not found" in Turn 32. B then repeats the `dmesg` command removing the word try at the beginning. If LLMs cannot pinpoint the cause of friction, it is unlikely that they will be able to issue a repair that addresses the friction directly, a crucial ability in settings where LLMs are used for dispute resolution (Tan et al., 2024).

7 Related Work

More recently, the speech-act based approach outlined in Traum and Allen (1992) has been used to study cooperative grounding acts in the Meetup (cite) and Spot the Difference (cite) datasets (Mohapatra et al., 2024). While conversations in such scenarios also require grounding, both of these datasets involve conversational participants interacting in a *physical* setting such as looking at a picture or a 2D grid. Because of the additional modality, the mutually shared basis of their common ground (such as an object both or one participant can see) is not available to the reader and it's hard to capture what causes friction from text alone.

Markowska et al. (2023) try to track each speaker's version of common ground through speaker "beliefs" expressed in conversations in the LDC Callhome (Canavan et al., 1997) corpus. However, since the conversations in that corpus are between close friends or family and are not goal-driven, there's less incentive to build and maintain common ground—a mismatch in common ground might be quietly accommodated without conversational friction since there is no end goal. Moreover, although they try to keep track of propositions in the common ground, they are only first degree propositions revealed through text, friction is often caused by implicit acceptance, as we see in Fig. 2.

Khebour et al. (2024) annotate a task-oriented corpus for multi-modal features and dialogue moves in order to begin to model and enable prediction of shared beliefs and questions under discussion. Unlike our research, the authors use their annotated corpus to train LSTM-based classifiers of dialogue moves relevant to tracking the common ground. The authors find that, at times, utterances may or may not be aligned with other modalities such as gesture, posing the greatest challenge to both classification and prediction. This highlights the challenge of tracking common ground in physically situated dialogue; our dataset simplifies focus to text alone.

In recent work, Shaikh et al. (2024) use grounding acts to compare the degree of grounding performed by LLMs in human-LLM conversations and find that language models perform less computational grounding. Our work complements these directions, as we focus on whether LLMs can even *detect* when and how participants in a conversation might lose track of common ground.

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 common ground for task-oriented dialogue in a real-world, text-only setting. Our experimental results, both qualitative and quantitative reveal that friction in goal-oriented dialog is inevitable, and it takes effort from *both* participants to repair the common ground to make progress towards a task. We also find that keeping track of common ground 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 were adept at anticipating and preventing potential friction or issuing `Repair` acts once friction did happen, state-of-the-art LLMs such as `gpt-4o` struggled with detecting and explaining cases of friction in the absence of explicit evidence.

Since LLMs are settings such as assisting teachers and students in education (Wang et al., 2024), future work might look at evaluating and improving their ability to understand implicit ruptures in common ground—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 to pursue involves explicitly *modeling* common ground. Common ground consists in propositions that are part of conversational participants' conversationally relevant underlying mental state, and recent work has demonstrated that LLMs are capable of making plausible inferences about just such propositions in non-conversational settings (Hoyle et al., 2023). Conceptually, thought bubbles like the ones illustrated in Figure 1 could be populated automatically, leading to an operational way to detect common ground misalignments by similarity-based comparison and contrast of participants' individual belief spaces.

9 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.

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.

References

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.

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.

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.

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.

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.

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.

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.

A Appendix

A.1 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 us 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 was the conversation 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 subpart 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 followup question. However, not all followup 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 in the friction or not.
```

A.2 Elaborations

Examples of elaborations can be found in Table 9.

A.3 Annotator Instructions

Before any annotation task, annotators had to fill-up 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.

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".

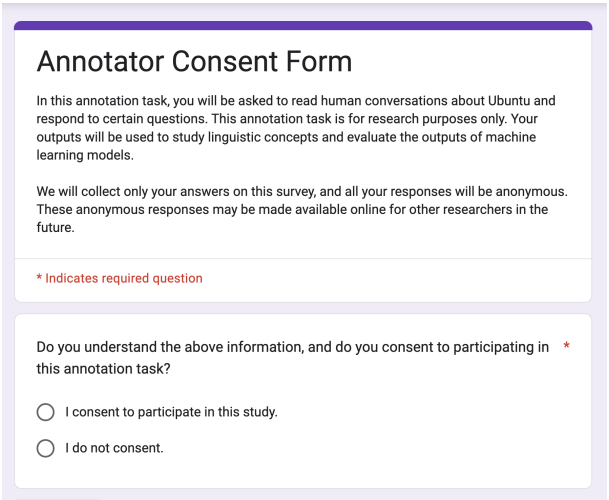

Figure 3: The consent form shown to annotators before each task.

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 have knowledge of the source of an explanation.