Inconsistent dialogue responses and how to recover from them

Mian Zhang[♦,*], **Lifeng Jin**[♡], **Linfeng Song**[♡], **Haitao Mi**[♡] and **Dong Yu**[♡]
[♦]Virginia Tech [♡]Tencent AI Lab, USA
mianz@vt.edu, lifengjin@global.tencent.com

Abstract

One critical issue for chat systems is to stay consistent about preferences, opinions, beliefs and facts of itself, which has been shown a difficult problem. In this work, we study methods to assess and bolster utterance consistency of chat systems. A dataset is first developed for studying the inconsistencies, where inconsistent dialogue responses, explanations of the inconsistencies, and recovery utterances are authored by annotators. This covers the life span of inconsistencies, namely introduction, understanding, and resolution. Building on this, we introduce a set of tasks centered on dialogue consistency, specifically focused on its detection and resolution. Our experimental findings indicate that our dataset significantly helps the progress in identifying and resolving conversational inconsistencies, and current popular large language models like ChatGPT which are good at resolving inconsistencies however still struggle with detection.[1]

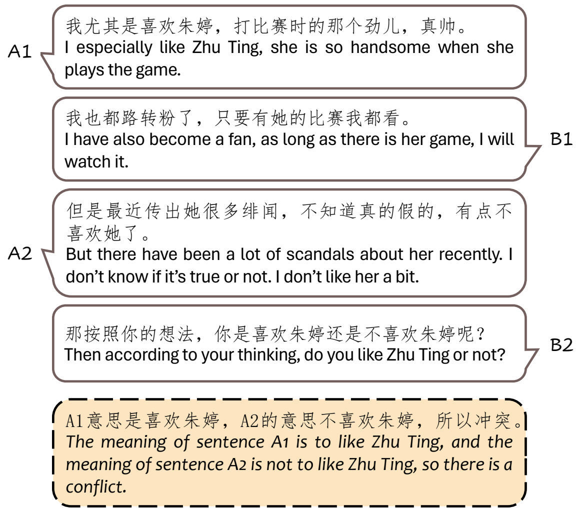

Figure 1: An instance in **CIDER** dataset. {A, B}x denotes the x-th utterance of one of the two speakers (A or B). An inconsistent utterance (A2 in this case), an explanation of the inconsistency (the dotted box), and a clarification response (B2 in this case) are written for each dialogue.

1 Introduction

For years, inconsistencies in human-to-chatbot conversations have been evident (Dziri et al., 2019; Qin et al., 2021; Ji et al., 2023), even in the era of large language models (Mündler et al., 2023). We categorize these inconsistencies as either extrinsic or intrinsic. *Extrinsic* inconsistencies (Rashkin et al., 2021; Santhanam et al., 2021) arise when there's a discrepancy between a statement and an external source of information, such as a knowledge base. On the other hand, *intrinsic* inconsistencies (Dziri et al., 2019; Nie et al., 2021; Zheng et al., 2022) occur within the dialogue itself. These can manifest in two ways: through an intra-utterance contradiction (Zheng et al., 2022), where a single sentence contains conflicting information, or a history contradiction (Nie et al., 2021), where a current statement conflicts with a previous one. Our study particularly addresses history contradictions, a persistent challenge in conversational models due to the nature of language modeling: models could forget what they said due to intervening context (Roller et al., 2021).

Researchers have been actively exploring how to resolve inconsistencies between utterances generated by conversational models in recent years. Li et al. (2020); Rashkin et al. (2021) has made progress in this domain by enhancing the training of these models, incorporating additional features and objectives to bolster self-consistency. Furthermore, Lee et al. (2022); Su and Collier (2022) introduced innovative decoding algorithms aimed at fostering greater coherence in utterances. These preemptive approaches are designed to mitigate conversational inconsistencies by reducing the likelihood of generating responses that contradict pre-

*Work done as an intern at Tencent AI Lab.

[1]The dataset and codebase are released at https://github.com/mianzhang/CIDER.

Findings of the Association for Computational Linguistics: EACL 2024, pages 220–230
March 17-22, 2024

vious dialogue. However, these approaches cannot resolve the inconsistencies that do occur, possibly from the user or from model errors. Therefore it's equally important to robustly address inconsistencies that do arise. Various remedial techniques have shown promise in other tasks, from grammar error correction (Wu et al., 2023) and moderating inappropriate dialogue content (Zhang et al., 2023), to generating clarifying questions in information retrieval (Zamani et al., 2020a) and conversational question answering (Guo et al., 2021). However, there seems to be a significant gap in the research when it comes to directly addressing inconsistencies that do arise between utterances.

In this work, we first propose a human-authored dataset with 27,180 dialogues to study the inconsistencies between utterances. At a high level, the dataset, called **CIDER**, covers the whole life span of in**C**onsistencies, encompassing their **I**ntroduction, un**DE**rstanding, and **R**esolution. Specifically, for each dialogue, annotators are first asked to write an utterance with inconsistent content regarding one utterance in the history to continue the conversation (A2 in Figure 1), and then explain why the two utterances are inconsistent with natural language (the dotted box in Figure 1), and finally provide a clarification response to continue the dialogue to resolve the inconsistency[2] (B2 in Figure 1). Given its large collection of inconsistent utterances paired with clarifying responses, **CIDER** stands out as a valuable resource for researching strategies to mitigate conversational inconsistencies.

Utilizing the **CIDER** dataset, we conduct comprehensive experiments and analyses to study dialogue inconsistencies. Our findings underscore that **CIDER** can facilitate the development of robust inconsistency checkers compared to models trained on comparable public datasets. In addition, our research indicates that classic models like T5 and BART face challenges in adeptly resolving inconsistencies by providing clarifying responses. When assessing the proficiency of large language models (LLMs) in identifying and resolving conversational inconsistencies, we discerned two key points: 1) LLMs, when employed as inconsistency checkers, still leave much to be desired in terms of performance. 2) In contrast, as resolvers of inconsistency, LLMs exhibit a higher success rate compared to the fully supervised BART resolver.

[2]The dialogues and annotation in the dataset are in Chinese. We also offer an English version translated by ChatGPT to facilitate research.

2 Related work

Consistency checking. Natural Language Inference (NLI) (Hu et al., 2020; Saha et al., 2020) is a task closely related to our work, where a provided hypothesis is evaluated for its logical consistency with a given premise, with both presented in natural language. Within the context of dialogues, Welleck et al. (2019) framed the consistency checking in dialogue as NLI and annotated binary consistency labels between dialogue-persona or persona-persona sentence pairs from the Persona-Chat dataset (Zhang et al., 2018). Dziri et al. (2019) employed NLI models to assess topic coherence between a current response and the preceding dialogue history. Meanwhile, Shuster et al. (2022) delved into the issue of role confusion, where dialogue systems might inadvertently adopt the identity of the other party involved, and proposed a reranker trained with human judgments of identity consistency. The most relevant works are from (Nie et al., 2021) and (Zheng et al., 2022), where they created datasets providing supervision for contradiction detection between conversational sentences. Our work distinguishes itself by providing more extensive annotations, including explanations and clarification responses.

Consistency resolving in dialogue. To enhance the self-consistency of conversational models, Rashkin et al. (2021) employed controllable features, steering models towards generating more consistent responses. Lee et al. (2022) introduced factual-nucleus sampling and factuality-enhanced continued training to augment the reliability of language models during both decoding and training phases. Shuster et al. (2022) encouraged the conversational models to maintain an identity with the help of a role-playing accuracy classifier. Li et al. (2020) explored unlikelihood training (Welleck et al., 2020) to curb inconsistencies in dialogue. However, given computational constraints, contemporary conversational models tend to rely predominantly on recent dialogue history when formulating responses. This predisposes them to produce content that may contradict earlier parts of the dialogue, especially distant sections. Generating clarification questions has emerged as a strategy to address communication breakdowns in dialogues, such as resolving ambiguities in a query during conversational information retrieval (Zamani et al., 2020b) or clarifying ambiguous user questions in conversational question answering (Guo et al., 2021) scenarios. In this research, we propose an approach to recover from conversational inconsistencies by generating clarification questions, with the support of the proposed dataset.

Large language models. Recent advancements in AI have been dominated by the rise of large language models, notably ChatGPT (Ouyang et al., 2022), GPT-4 (OpenAI, 2023) and others. They has shown that by scaling up language models, they can be equipped to tackle intricate tasks, such as question answering, machine translation, and numerical reasoning. In this study, leveraging the extensive annotations of our proposed dataset, **CIDER**, we assess these models' proficiency in detecting and addressing conversational inconsistencies.

3 Data collection

The candidate conversations for annotation are sampled from two open-source conversation datasets: LCCC and NaturalConv. LCCC (Wang et al., 2020) is a large collection of short conversations from the Chinese social media platform Weibo. NaturalConv (Wang et al., 2021) is an annotator-written dataset containing conversations around news items on topics like film and sports. They are different in content and style. LCCC conversations tend to be short in number of turns, and more in the style of daily chitchat. NaturalConv conversations, in contrast, are two to five times longer and contain more serious discussions about events in sports, films, and other areas. 20,000 and 10,000 conversations are sampled from the LCCC and NaturalConv respectively for annotation. When sampling, conversations that are shorter than 4 turns or contain utterances shorter than 5 words are filtered out.

The sampled conversations are generally consistent, therefore the goal of data annotation is to create an alternative conversation that contains inconsistent utterances. To achieve this, we truncate the original conversation to create a common conversation context. For LCCC, the last utterance is truncated for inconsistent continuation writing; for NaturalConv, a random turn between 8 and $l - 4$[3] and the following turns are chosen for truncation, where l is the length of the conversation.

Finally, a specified source turn is sampled from the last turn or the turn before the last. This source turn is designated to be the source of the inconsistency where the following inconsistent continuation needs to form inconsistency with the utterance from the same speaker in this turn.

[3]The last turns of NaturalConv tend to be goodbyes, therefore we choose to truncate before such utterances.

	LCCC			NaturalConv		
	Train	Dev	Test	Train	Dev	Test
# of Convs	14,126	1,883	1,797	7,537	917	920
Ave. Cont. Len.	29.3	28.9	28.9	40.4	40.9	40.5
Ave. Exp. Len.	40.9	40.5	41.0	50.4	50.3	50.3
Ave. Res. Len.	16.2	16.1	16.1	20.3	20.1	20.0

Table 1: Some basic statistics of the annotated datasets. Ave. Cont. Len. is the average continuation length in number of Chinese characters; Ave. Exp. Len. is the average explanation length; Ave. Res. Len. is the average resolution question length. They correspond to the outcome from the three annotation tasks.

4 Annotation guidelines

The annotation process has been divided into three different tasks: inconsistent continuation, inconsistency explanation, and inconsistency resolution, which are required to be performed to each candidate conversation by one annotator when given a candidate conversation and a specified source turn.

Inconsistent continuation. The annotator first tries to create a natural continuation of the conversation by providing a possible utterance to the candidate conversation, but forms an inconsistency with the specified source utterance (A2 in Figure 1 is the continuation, and A1 is the source.) The annotators are instructed to write the utterance with contradictory viewpoints, reasoning, and argumentation, instead of providing simple negation to the source utterance. For example, for the specified utterance *I went to the supermarket yesterday.*, the continuation meeting the annotation requirement is *I have been staying home for the past four days, not really wanting to go anywhere*, instead of *I didn't go to the supermarket yesterday.*

Inconsistency explanation. After writing the continuation of the candidate conversation, the annotator is instructed to write down the rationale behind the created inconsistency (the dashed box in Figure 1). They are asked to follow this template when writing the rationale: *The specified utterance means X, but the continuation utterance means Y, which is in contradiction with X.*, where the utterance meanings should be explicit. In the example above, the explanation one may write is *The specified utterance indicates that I went out of my home*

	Pair-Check						*Diag-Check*					
	Train		Valid		Test		Train		Valid		Test	
	#Pos	#Neg	#Pos	#Neg	#Pos	#Neg	#Pos	#Neg	#Pos	#Neg	#Pos	#Neg
STANCE	1816	3959	195	446	346	644	1816	3959	195	446	346	644
OCNLI	14837	30601	1639	3409	900	2100	14837	30601	1639	3409	900	2100
CDConv	2623	4373	880	1452	848	1484	2623	4373	880	1452	848	1484
CIDER	21663	53012	2800	6692	2717	6569	21663	21663	2800	2800	2717	2717

Table 2: Dataset statistics for checking tasks. Pos/Neg corresponds to label *inconsistent*/*consistent*.

yesterday, but the continuation utterance means that I didn't go out for many days including yesterday, which is in contradiction with the previous statement.

Inconsistency resolution. Finally, the annotator provides another utterance to expose and question the inconsistency from a different party than the continuation party (B2 in Figure 1). The annotator is asked to write the resolution question naturally with the main purpose being clarifying the situation instead of complaining. They are also asked to try varying how the clarification question is raised, because the most intuitive way is asking by providing a binary choice. The resolution question for the example above is *So were you home yesterday or did you go to the supermarket?*

Twelve examples collected from the two data sources and annotated by the authors were provided to the annotators along with the guidelines, which cover a number of common mistakes that the authors discovered in the trial annotation. The annotation project lasted two months, with six annotators[4] participating in the project from a commercial annotation provider, who was chosen amongst three providers based on the performance in the trial annotation task. The items for annotation were segmented into batches, each with 3000 conversations. The annotated items are checked first by quality assurance specialists from the annotation provider by batch, and then spot-checked by the authors with the acceptance rate setting at 95%.[5] Candidate conversations which are not possible to form inconsistencies, such as conversations containing mostly utterances of simple greeting or agreeing, are dropped in the annotation process.

5 Data overview

After annotation, 17,806 conversations from LCCC and 9,374 conversations from NaturalConv have valid annotation. They are further split into train, dev and test sets, shown in Table 1. The average continuation and explanation lengths from LCCC conversations are substantially shorter than from NaturalConv, indicating the simple nature of social media conversations. The resolution question lengths are closer than the other lengths, showing that resolution questions tend to be less influenced by context and style.

6 Consistency checking

In this section, we experimentally verify whether the proposed **CIDER** could help the detection of inconsistency in conversation via two task settings: (1) checking the consistency between two sentences (*Pair-Check*); (2) checking the consistency between an utterance and its preceding context (*Diag-Check*). The (inconsistency) checker is initialized as RoBERTa-base (Liu et al., 2019) with a linear binary classification head on the top. The input of the encoder for *Pair-Check* is formatted as "`[CLS]` {*sentence 1*} `[SEP]` {*sentence 2*} `[SEP]`" while for *Diag-Check*, "`[CLS]` {*context*} `[SEP]` {*utterance*} `[SEP]`", where the `[CLS]` and `[SEP]` are special tokens.

Baselines. We compare **CIDER** with several related datasets:

- CDConv (Zheng et al., 2022): a dataset with 12K dialogues for conversational contradiction detection. Compared to **CIDER**, CDConv covers another two types of contradiction: intra-sentence contradiction and role confusion. Each dialogue of CDConv contains two turns of utterances between a user and a bot

[4] The chosen provider created a qualification test based on the annotation guidelines for selecting annotators. The annotators with the highest agreement with the authors were then chosen as annotators. They then went through an online training session with the authors to align with the understanding of guidelines from the authors. They were paid twice the local average monthly salary for their contributions.

[5] The spot-check rate is 10%.

	STANCE Test			*OCNLI Test*			*CDConv Test (Turn)*			**CIDER** *Test (Turn)*		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
$\mathrm{C}^{Turn}_{\mathrm{STANCE}}$	**72.8**	**60.4**	$\underline{\mathbf{66.0}}_{\Uparrow 14.3}$	37.7	19.4	*25.7*	38.1	21.3	*27.4*	37.5	14.4	*20.8*
$\mathrm{C}^{Turn}_{\mathrm{OCNLI}}$	31.6	36.1	*33.7*	**72.9**	74.9	$\underline{\mathbf{73.9}}_{\Uparrow 10.2}$	51.3	37.3	*43.2*	35.7	37.4	$\mathit{36.5}_{\uparrow 1.4}$
$\mathrm{C}^{Turn}_{\mathrm{CDConv}}$	41.8	8.1	*13.6*	40.9	15.0	*22.0*	**56.3**	**72.9**	$\underline{\mathbf{63.5}}_{\Uparrow 14.7}$	29.8	42.8	*35.1*
$\mathrm{C}^{Turn}_{\mathtt{CIDER}}$	61.0	44.8	$\mathit{51.7}_{\uparrow 18.0}$	30.7	**76.2**	$\mathit{63.7}_{\uparrow 38.0}$	37.7	69.3	$\mathit{48.8}_{\uparrow 5.6}$	**76.2**	**69.3**	$\underline{\mathbf{72.6}}_{\Uparrow 36.1}$

(a) Performance of *Pair-Check* checkers.

	STANCE Test			*OCNLI Test*			*CDConv Test (Diag)*			**CIDER** *Test (Diag)*		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
$\mathrm{C}^{Turn}_{\mathrm{STANCE}}$	**72.8**	**60.4**	$\underline{\mathbf{66.0}}_{\Uparrow 20.4}$	37.7	19.4	*25.7*	25.9	4.5	*7.6*	48.4	21.8	*30.0*
$\mathrm{C}^{Turn}_{\mathrm{OCNLI}}$	31.6	36.1	*33.7*	**72.9**	**74.9**	$\underline{\mathbf{73.9}}_{\Uparrow 40.8}$	46.6	37.6	$\mathit{41.6}_{\uparrow 18.6}$	52.5	42.7	$\mathit{47.1}_{\uparrow 17.1}$
$\mathrm{C}^{Diag}_{\mathrm{CDConv}}$	54.5	8.7	*15.0*	31.5	16.2	*21.4*	**62.5**	**60.8**	$\mathbf{61.7}_{\Uparrow 20.1}$	61.3	8.3	*14.6*
$\mathrm{C}^{Diag}_{\mathtt{CIDER}}$	38.8	55.2	$\mathit{45.6}_{\uparrow 11.9}$	33.7	32.4	$\mathit{33.1}_{\uparrow 7.4}$	52.7	14.7	*23.0*	**89.4**	**91.6**	$\mathbf{90.5}_{\Uparrow 43.4}$

(b) Performance of *Diag-Check* checkers.

Table 3: Performance of the checking tasks. The checker trained on dataset Y for task *X-Check* is denoted as $\mathrm{C}^{X}_{\mathrm{Y}}$. The best result in each column is in bold. The best F1 score on each dataset is underscored and the points by which it exceeds the second best are shown by $\Uparrow$. The transferring F1 scores on each dataset are in italics and the points by which they exceed the second best transferring score are shown by $\uparrow$. The performance of $\mathrm{C}^{Turn}_{\mathrm{STANCE}}$ and $\mathrm{C}^{Turn}_{\mathrm{OCNLI}}$ on *STANCE Test* and *OCNLI Test* in Table 3b is copied from Table 3a.

	Merge						*Pretrain*					
	Pair-Check			*Diag-Check*			*Pair-Check*			*Diag-Check*		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
$\mathrm{C}_{\mathtt{CIDER}}$	76.2	69.3	72.6	**89.4**	91.6	90.5	76.2	69.3	72.6	89.4	91.6	90.5
+CDConv	**76.7**	72.5	$74.6_{\Uparrow 2.0}$	**90.7**	91.9	$\mathbf{91.3}_{\Uparrow 0.8}$	76.4	**71.1**	$73.7_{\Uparrow 1.1}$	88.4	91.4	$89.9_{\Downarrow 0.6}$
+OCNLI	70.1	77.4	$73.6_{\Uparrow 1.0}$	89.8	92.1	$90.9_{\Uparrow 0.4}$	**77.4**	70.7	$\mathbf{73.9}_{\Uparrow 1.3}$	88.6	**93.1**	$\mathbf{90.8}_{\Uparrow 0.3}$
+STANCE	72.4	**77.9**	$\mathbf{75.1}_{\Uparrow 2.5}$	88.2	**92.9**	$90.5_{\Uparrow 0.0}$	76.2	70.3	$73.2_{\Uparrow 0.6}$	87.3	92.7	$89.9_{\Downarrow 0.6}$

Table 4: Performance of checkers leveraging extra data on the test set of **CIDER**. The best are in bold. The relative increasing ($\Uparrow$) and decreasing ($\Downarrow$) points are calculated based on the performance of $\mathrm{C}_{\mathtt{CIDER}}$.

and annotation of *consistent* or *inconsistent* between the replies of the bot.

- STANCE[6]: a dataset for stance classification of articles of debating topics from online forums, where sentence pairs against each other are marked as *inconsistent* and otherwise *consistent*.
- OCNLI (Hu et al., 2020): a large-scale natural language inference (NLI) dataset, consisting of about 56,000 annotated sentence pairs. We regard sentence pairs with *contradiction* label as *inconsistent* and others as *consistent*.

Implementation details. For **CIDER**, when creating *consistent* training instances of *Pair-Check*, we regard all the utterances in the context of the same speaker without *inconsistent* label as being consistent with the current response; and when creating the training instances of *Diag-Check*, we drop current response with inconsistency and regard the previous response as being consistent with the context. Table 2 shows the statistics of the datasets for these two checking tasks.

We adopt AdamW (Loshchilov and Hutter, 2019) to optimize models for 50 epochs with a learning rate of 1e-6 and a batch size of 16. We evaluate the model on the validation set at each epoch and keep the one with the best performance with an early stop patience of 3. All the results are averaged over three runs. Our experiments are run on two Nvidia V100 GPUs.

Results for *Pair-Check*. The performance of checkers trained on different datasets for *Pair-*

[6] www.fudan-disc.com/sharedtask/AIDebater21/tracks.html

Check is demonstrated in Table 3a. For each checker, we show its performance on all the test sets of the evaluating datasets.

There is a substantial distribution difference between the datasets with the checker trained on one dataset performing the best on the corresponding test set. $C^{Turn}_{\texttt{CIDER}}$ has the largest exceeding F1 points over the second best, 36.1, indicating that the checker trained on other datasets is not good at detecting the consistency in the test set of **`CIDER`** and the training set of **`CIDER`** could provide useful supervision for it. Moreover, we compare the 0-shot transfer ability of checkers across the datasets. Results show that $C^{Turn}_{\texttt{CIDER}}$ has the best transfer results on all the other three datasets, surpassing the second best by 18.0, 38.0, and 5.6 F1 points, respectively, demonstrating $C^{Turn}_{\texttt{CIDER}}$ covering many similar linguistic phenomena in other datasets. On the whole, **`CIDER` provides robust supervision to check whether a pair of sentences are consistent, regardless of they are in a dialogue or not**.

Results for *Diag-Check*. The performance of the checkers trained on different datasets for *Diag-Check* is demonstrated in Table 3b. The results of C^{Diag}_{CDConv} and $C^{Diag}_{\texttt{CIDER}}$ indicates again the distribution difference between **`CIDER`** and CDConv also being significant for *Diag-Check* task: **`CIDER`** do not cover role confusion and intra-sentence contradiction these two types of inconsistency while being much larger than CDConv. In addition, $C^{Diag}_{\texttt{CIDER}}$ outperforms C^{Diag}_{CDConv} on *STANCE Test* by 30.6 F1 points and on *OCNLI Test* by 11.7 F1 points, which demonstrates better transferring ability of $C^{Diag}_{\texttt{CIDER}}$ to non-conversational scenarios. Therefore, along with the transferring results in Table 3a, **`CIDER` offers more transferable patterns for checking consistency, and may be complementary to CDConv in the conversational scenarios**. We also notice that C^{Turn}_{OCNLI} is superior to $C^{Diag}_{\texttt{CIDER}}$ on *CDConv Test (Diag)* and to C^{Diag}_{CDConv} on ***`CIDER`** Test (Diag)*, showing that the knowledge of inconsistency between sentences in OCNLI is also useful for the inconsistency checking in dialogue.

Role of extra data. We are interested in whether other datasets could improve the performance of $C_{\texttt{CIDER}}$. We leverage the training data of STANCE, OCNLI, and CDConv via two ways: 1) directly merging one of them into the training data of **`CIDER`** (*Merge*); 2) pretraining the checker on one of them before training on **`CIDER`** (*Pretrain*).

The results are presented in Table 4. It's evident that **incorporating additional data generally enhances the overall performance of $C_{\texttt{CIDER}}$**, The only exception is that only pretraining on OCNLI could improve the checker for *Diag-Check* task, which indicates better supervision signal from OCNLI for checking the inconsistency of an utterance. Compared with pretraining on extra data, directly merging them is superior, which could be ascribed to the phenomenon of catastrophic forgetting (Kirkpatrick et al., 2017) of pretrained models. Moreover, *Pair-Check* generally benefits from the extra datasets more than *Diag-Check* because most of the extra datasets are intrinsically designed for checking of sentence pairs and in large quality so models could learn generalized patterns from them.

LLMs as consistency checker. We investigated the potential of large language models (LLMs) to function as robust consistency checkers. We pre-examine five human-crafted prompts for each task using a small-scale test set (50 instances) and select the best. The prompts applied for the checking tasks are illustrated in Figure 2. The evaluating LLMs are ChatGPT and GPT4[7]. As shown in Table 5, LLM-based checkers significantly lag behind the fully supervised $C_{\texttt{CIDER}}$, indicating that there is still much room for improvement. Moreover, the higher performance of GPT4 over ChatGPT underscores that larger LLMs possess a better capability to detect inconsistencies.

Pair-Check	Whether the following two sentences are semantically related and have semantic inconsistencies, please answer "yes" or "no". sentence 1: {sentence 1} sentence 2: {sentence 2}
Diag-Check	Please answer "yes" or "no" if the speaker of the last sentence in the following dialogue contradicts himself, and give an explanation. {dialogue}

Figure 2: Prompts of checking tasks.

7 Consistency resolution

Inconsistent responses of a conversational model could be detected by a consistency checker in advance, avoiding being exposed to users. However, inconsistent responses from a user can not be ignored by chat systems. The existence of inconsistent content may confuse the conversational model

[7]We use the versions gpt-3.5-turbo-0613 and gpt-4-0613 across our experiments.

	Pair-Check			*Diag-Check*		
	Pre.	Rec.	F1	Pre.	Rec.	F1
$C_{\texttt{CIDER}}$	**76.2**	69.3	**72.6**	**89.4**	**91.6**	**90.5**
ChatGPT	42.0	**79.0**	54.8	57.2	84.9	68.4
GPT4	49.9	76.2	60.3	68.8	82.1	74.8

Table 5: Performance of LLMs on checking tasks.

and induce undesired responses. Resolving the occurred inconsistency is necessary to maintain a smooth dialogue flow with clear semantics. The proposed **CIDER** dataset contributes to resolving the occurred inconsistency in a dialogue with *clarification responses*, which is a valuable source to train an inconsistency resolution model.

We choose the base version of two representative conditional generative models to initialize the resolver: BART (Lewis et al., 2020) and T5 (Raffel et al., 2020). They both follow an encoder-decoder structure and generate clarification responses in a sequence-to-sequence fashion: the conversational text with inconsistency is fed into the encoder and the clarification response is generated auto-aggressively by the decoder. Like the checking experiments in section 6, we consider two task settings: (1) generating a clarification response for a pair of inconsistent utterances (*Pair-Resolve*); (2) generating a clarification response for a dialogue, of which the current response is inconsistent to the preceding context (*Diag-Resolve*). The input of the encoder for *Pair-Resolve* is formatted as "`[CLS]` {*utterance 1*} `[SEP]` {*utterance 2*} `[SEP]`" while for *Diag-Resolve*, "`[CLS]` {*context*} `[SEP]` {*response*} `[SEP]`".

Implementation details. We use the same optimization configuration of checkers to train the resolvers, except that a learning rate of 3e-4 is used for T5. BART and T5 are loaded with pretrained parameters from Zhao et al. (2019) and Shao et al. (2021), respectively. In decoding, we adopt Nucleus Sampling (Holtzman et al., 2020) with top-0.90 probability mass across the experiments.

Evaluation. We use BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), including ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-L (R-L), to measure the similarity between the generated text and the ground truth.

Results. According to rows #1 and #2 in Table 6, BART shows better performance in both *Pair-Resolve* and *Diag-Resolve* tasks than T5, indicating the pretrained parameters of BART are more suitable to inconsistency resolving. Meanwhile, the points of *Pair-Resolve* are higher than those of *Diag-Resolve*, which could be ascribed to *Diag-Resolve* being a more difficult task than *Pair-Resolve* because recognizing inconsistent contents between conversational context and a response is harder than between a pair of sentences. We also try appending *explanations* to the input of the encoder to aid the generation process. Specifically, the input becomes "`[CLS]` {*utterance 1*} `[SEP]` {*utterance 2*} `[SEP]` {*explanation*} `[SEP]`" for *Pair-Resolve* and "`[CLS]` {*context*} `[SEP]` {*response*} `[SEP]` {*explanation*} `[SEP]`" for *Diag-Resolve*. The models with *explanation* are denoted as $\text{T5}_{\text{oracle}}$ and $\text{BART}_{\text{oracle}}$, whose performances are shown at rows #3 and #4 in Table 6. We could see that $\text{T5}_{\text{oracle}}$ and $\text{BART}_{\text{oracle}}$ surpass T5 and BART by a significant margin, showing that with *explanations* informing what inconsistency the input delivers, the models are able to produce clarification responses more semantically similar to the ground truth. Moreover, $\text{BART}_{\text{oracle}}$ performs better than $\text{T5}_{\text{oracle}}$ across all the metrics, demonstrating BART is better at exploiting *explanations* to resolve semantic inconsistency.

Analysis. We go through 200 randomly selected instances (100 from *Pair-Resolve* and 100 from *Diag-Resolve*) of the best-performing BART resolver to 1) check whether the generated responses successfully clarify the inconsistent content and 2) explore the possible reasons that the clarification fails. The numbers of successful instances are presented in Table 7. We could see **BART faces challenges in inconsistency resolution** and there is still large room for improvement. The higher success count for *Pair-Resolve* compared to *Diag-Resolve* indicates again that resolving inconsistencies between a response and its context poses greater challenges. We summarise the main types of failed clarification as follows:

1. The resolver misses inconsistent content and just picks irrelevant semantic units to form a clarifying response. For instance, the user first says *I want to buy a cup of coffee because I'm so sleepy.* and then *Great, let's try Chinese tea!*. The resolver responds with *Are you on earth sleepy or not?* This error type is common in *Diag-Resolve* because long context contains irrelevant information that interferes with locating inconsistent content.

	Model	Pair-Resolve				Diag-Resolve			
		BLEU	R-1	R-2	R-L	BLEU	R-1	R-2	R-L
#1	T5	26.9	55.3	33.0	52.2	14.8	43.0	20.6	40.4
#2	BART	$28.2_{\Uparrow 1.3}$	$57.2_{\Uparrow 1.9}$	$34.8_{\Uparrow 1.8}$	$53.7_{\Uparrow 1.5}$	$14.9_{\Uparrow 0.1}$	$43.7_{\Uparrow 0.7}$	$21.7_{\Uparrow 1.1}$	$41.0_{\Uparrow 0.6}$
#3	$T5_{oracle}$	46.2	71.5	53.0	68.3	46.7	71.7	53.2	68.3
#4	$BART_{oracle}$	$49.4_{\Uparrow 3.2}$	$74.4_{\Uparrow 2.9}$	$56.2_{\Uparrow 3.2}$	$70.7_{\Uparrow 2.4}$	$47.4_{\Uparrow 0.7}$	$72.4_{\Uparrow 0.7}$	$53.9_{\Uparrow 0.7}$	$68.7_{\Uparrow 0.4}$
#5	ChatGPT	14.3	45.2	22.2	41.4	5.3	29.8	9.9	26.9
#6	GPT4	10.8	42.7	20.2	38.0	4.1	28.0	9.8	24.2

Table 6: Performance of resolvers on the test set of **CIDER**. The relative increasing ($\Uparrow$) points of BART ($BART_{oracle}$) are calculated based on the performance of T5 ($T5_{oracle}$).

Model	#Succ. / #Total Pair-Resolve	Diag-Resolve
BART	56 / 100	36 / 100
$BART_{oracle}$	91 / 100	82 / 100
ChatGPT	76 / 100	64 / 100
GPT4	92 / 100	79 / 100

Table 7: The number of successfully resolved instances.

2. The resolver includes the inconsistent content in the response but fails to form a fluent, contextual coherent response. For example, the user first says *Are you free? I want you to do me a favor.* and then *I am busy now.* and the resolver replies with *Can you do a favor at all?*. In this case, the resolver misunderstands who is the subject of the action, thus providing a response incoherent to the context.

LLMs as consistency resolver. We examine the consistency resolution ability of LLMs by asking LLMs to form a clarification response for the two resolving tasks via the prompts shown in Figure 3 (one in-context example is included in the prompts to ensure a fixed output format).

We report automatic evaluation results in rows #5 and #6 of Table 6. On the selected instances in subsection **Analysis**, we conduct the same human evaluation of the generated clarification response of the LLMs and show the results in Table 7. Results indicate that: **while ChatGPT and GPT4, both cutting-edge LLMs, score lower in BLEU and ROUGE compared to T5 and BART, they excel in addressing inconsistencies in dialogue history**, whose performance rivals that of the oracle resolvers. The lower BLEU and ROUGE scores of LLMs can be attributed to their tendency to produce more varied and extensive sentences. To illustrate, consider the reference clarification sentence: *Do you really want to eat hot pot or barbecue?*. BART's response is, *Do you really want to eat hot pot or not?*, whereas GPT4 offers, *So, are you more attracted to hot pot, or does barbecue appeal to you more?*.

Pair-Resolve

You will be given two contradictory sentences from a person, and you need to reply to him and ask him what he really thinks. Like the following example:
{sentence 1}
{sentence 2}
{reply}
{sentence 1}
{sentence 2}
What is the reply?

Diag-Resolve

You will be given a dialogue between A and B, in which the current speaker says something contradictory, and you need to generate a reply from another person to ask him what he really thinks. Like the following example:
{dialogue1}
{reply}
{dialogue2}
What is the reply?

Figure 3: Prompts of resolving tasks.

8 Conclusion

We present **CIDER**, a comprehensive dialogue dataset comprising 27,180 annotated dialogues to investigate conversational inconsistencies. The annotations of **CIDER** cover the whole life span of inconsistencies: the human-authored utterances with inconsistent content demonstrate the introduction of inconsistencies; the explanations help understand the inconsistencies; and the clarification responses exemplify how to resolve the inconsistencies. Through rigorous experiments and analysis, we show that **CIDER** significantly advance the detection and resolution of conversational inconsistencies, and large language models, ChatGPT and GPT4, exhibit commendable performance in resolving these conversational inconsistencies but struggle with identifying them.

Limitation

Our work has following limitations:

- Our proposed dataset emphasizes contradictions between utterances. For a truly effective system that detects or resolves inconsistencies, it is essential to incorporate resources that address other types of inconsistencies, such as intra-utterance or extrinsic discrepancies.

- We've currently evaluated the ability of LLMs to function as independent resolvers under specific prompts to generate clarification questions. The potential for these models to autonomously identify and clarify inconsistencies remains an intriguing avenue for future exploration. Moreover, while our evaluation of LLMs relies on the optimal prompts chosen from several human-crafted options, a more rigorous approach to prompt engineering could potentially yield superior outcomes.

Ethical consideration

Our dataset, along with the LCCC (Wang et al., 2020) and NaturalConv (Wang et al., 2021) sources, have been cleaned to ensure no breaches of privacy (further details are available in their respective papers). All annotation guidelines (as detailed in Section 4) have received approval from the ethics review committee. We are confident that **CIDER** will play a pivotal role in crafting more human-friendly conversational models.

References

Nouha Dziri, Ehsan Kamalloo, Kory Mathewson, and Osmar Zaiane. 2019. Evaluating coherence in dialogue systems using entailment. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3806–3812, Minneapolis, Minnesota. Association for Computational Linguistics.

Meiqi Guo, Mingda Zhang, Siva Reddy, and Malihe Alikhani. 2021. Abg-CoQA: Clarifying ambiguity in conversational question answering.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Hai Hu, Kyle Richardson, Liang Xu, Lu Li, Sandra Kübler, and Lawrence Moss. 2020. OCNLI: Original Chinese Natural Language Inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3512–3526, Online. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2022. Factuality enhanced language models for open-ended text generation. *Advances in Neural Information Processing Systems*, 35:34586–34599.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. 2020. Don't say that! making inconsistent dialogue unlikely with unlikelihood training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4715–4728, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2023. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *ArXiv preprint*, abs/2305.15852.

Yixin Nie, Mary Williamson, Mohit Bansal, Douwe Kiela, and Jason Weston. 2021. I like fish, especially dolphins: Addressing contradictions in dialogue modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1699–1713, Online. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report. *ArXiv preprint*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

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

Libo Qin, Tianbao Xie, Shijue Huang, Qiguang Chen, Xiao Xu, and Wanxiang Che. 2021. Don't be contradicted with anything! CI-ToD: Towards benchmarking consistency for task-oriented dialogue system. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2357–2367, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Hannah Rashkin, David Reitter, Gaurav Singh Tomar, and Dipanjan Das. 2021. Increasing faithfulness in knowledge-grounded dialogue with controllable features. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 704–718, Online. Association for Computational Linguistics.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Swarnadeep Saha, Yixin Nie, and Mohit Bansal. 2020. ConjNLI: Natural language inference over conjunctive sentences. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8240–8252, Online. Association for Computational Linguistics.

Sashank Santhanam, Behnam Hedayatnia, Spandana Gella, Aishwarya Padmakumar, Seokhwan Kim, Yang Liu, and Dilek Hakkani-Tur. 2021. Rome was built in 1776: A case study on factual correctness in knowledge-grounded response generation. *ArXiv preprint*, abs/2110.05456.

Yunfan Shao, Zhichao Geng, Yitao Liu, Junqi Dai, Fei Yang, Li Zhe, Hujun Bao, and Xipeng Qiu. 2021. Cpt: A pre-trained unbalanced transformer for both chinese language understanding and generation. *ArXiv preprint*, abs/2109.05729.

Kurt Shuster, Jack Urbanek, Arthur Szlam, and Jason Weston. 2022. Am I me or you? state-of-the-art dialogue models cannot maintain an identity. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2367–2387, Seattle, United States. Association for Computational Linguistics.

Yixuan Su and Nigel Collier. 2022. Contrastive search is what you need for neural text generation. *ArXiv preprint*, abs/2210.14140.

Xiaoyang Wang, Chen Li, Jianqiao Zhao, and Dong Yu. 2021. Naturalconv: A chinese dialogue dataset towards multi-turn topic-driven conversation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14006–14014. AAAI Press.

Yida Wang, Pei Ke, Yinhe Zheng, Kaili Huang, Yong Jiang, Xiaoyan Zhu, and Minlie Huang. 2020. A large-scale chinese short-text conversation dataset. In *Natural Language Processing and Chinese Computing: 9th CCF International Conference, NLPCC 2020, Zhengzhou, China, October 14–18, 2020, Proceedings, Part I 9*, pages 91–103. Springer.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, Florence, Italy. Association for Computational Linguistics.

Haoran Wu, Wenxuan Wang, Yuxuan Wan, Wenxiang Jiao, and Michael Lyu. 2023. Chatgpt or grammarly? evaluating chatgpt on grammatical error correction benchmark. *ArXiv preprint*, abs/2303.13648.

Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. 2020a. Generating clarifying questions for information retrieval. In *Proceedings of The Web Conference 2020*, WWW '20, pages 418–428, New York, NY, USA. Association for Computing Machinery.

Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. 2020b. Generating clarifying questions for information retrieval. In *Proceedings of The Web Conference 2020*, WWW '20, pages 418–428, New York, NY, USA. Association for Computing Machinery.

Mian Zhang, Lifeng Jin, Linfeng Song, Haitao Mi, Wenliang Chen, and Dong Yu. 2023. Safeconv: Explaining and correcting conversational unsafe behavior. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22–35.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Zhe Zhao, Hui Chen, Jinbin Zhang, Xin Zhao, Tao Liu, Wei Lu, Xi Chen, Haotang Deng, Qi Ju, and Xiaoyong Du. 2019. UER: An open-source toolkit for pre-training models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 241–246, Hong Kong, China. Association for Computational Linguistics.

Chujie Zheng, Jinfeng Zhou, Yinhe Zheng, Libiao Peng, Zhen Guo, Wenquan Wu, Zheng-Yu Niu, Hua Wu, and Minlie Huang. 2022. CDConv: A benchmark for contradiction detection in Chinese conversations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 18–29, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.