Exploring Conversational Adaptability: Assessing the Proficiency of Large Language Models in Dynamic Alignment with Updated User Intent

Yu-Chuan Chen, Hen-Hsen Huang

Institute of Information Science, Academia Sinica, Taipei, Taiwan
cychen@iis.sinica.edu.tw, hhhuang@iis.sinica.edu.tw

Abstract

This paper presents a practical problem in dialogue systems: the capability to adapt to changing user intentions and resolve inconsistencies in conversation histories. It is crucial in scenarios like train ticket booking, where travel plans often change dynamically. Notwithstanding the advancements in NLP and large language models (LLMs), these systems struggle with real-time information updates during conversations. We introduce a specialized dataset to evaluate LLM-based chatbots on such conversational adaptability by asking a broad range of open-domain questions, focusing on scenarios where users modify their requests mid-conversation. Additionally, as LLMs are susceptible to generating superfluous sentences, we propose a novel, Chain-of-Thought-free evaluation framework to distill the user intention from their responses. Through extensive investigations on four LLMs, we observe that these contemporary LLMs are not well-aligned with the latest user intent in long-term conversations; they often fail to capture the nuances of natural conversations in a zero-shot setting. Interestingly, the results demonstrate that GPT-4, widely recognized as having the most advanced reasoning capabilities to date, is bested by GPT-3.5 in this task. This work aims to improve the practicality of LLM-based chatbots, bridging the gap between the current capabilities of dialogue systems and the fluidity of human interactions.

1 Introduction

In the dynamic flow of a conversation, it is common for speakers to shift their intentions and revise their previously spoken words. Take, for instance, the scenario of customers booking train tickets for travel. Naturally, their initial travel plans are often subject to change during the booking process, influenced by factors like ticket availability or weather conditions. In response to these changes, the booking agent, responsible for understanding and processing the flux of user intent, must promptly update its knowledge and adapt its responses to align with their latest requirements.

As dialogue systems continue to evolve, the number of online customer service interactions handled by NLP models has proliferated. Yet, the ability of these models, including state-of-the-art large language models (LLMs), to accurately and efficiently update information during a conversation remains a significant challenge. The difficulty stems from the need for an LLM to understand the nuances of human communication and dynamically adjust its understanding as the conversation progresses along with the emergence of new information (see Figure 1).

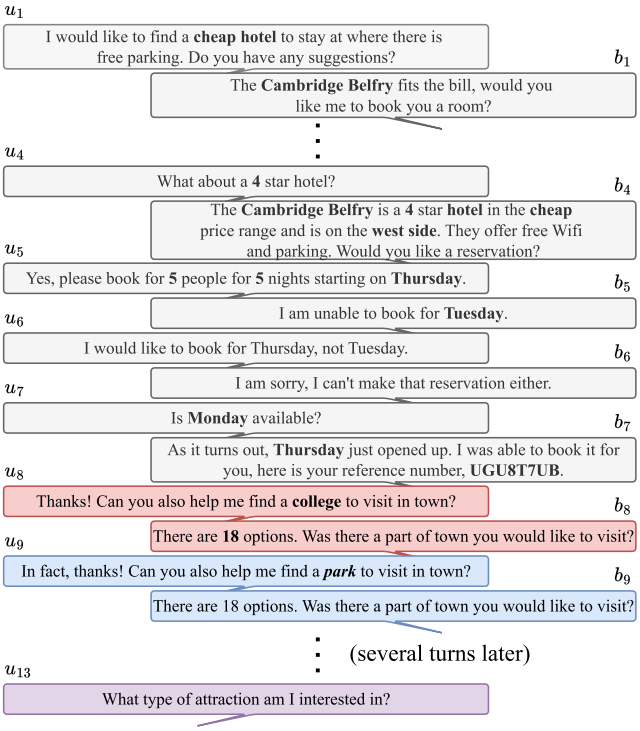

Figure 1: An example of our dataset: The customer made a wrong request in u_8 but corrected the attraction to "park" in u_9. Bold texts are the user intents labeled in MultiWOZ 2.2. Our dataset contains only one **valid** (defined in §3) intent in the update turn (u_9 and b_9; see §3), as the phrase "In fact" in u_9 should *implicitly* nullify the previous intent in u_8 (the *explicit* case is in §6.1). Note that the utterance u_9 is **only for exposition**, as we also test natural ones like "In fact, park." and "Oh, I'm sorry. Should have been park, not college." in this paper (see §6.1). Hence, LLMs should catch the update and reply only "park" rather than "college and park" in b_{13}. Our updated intent does not have "museum and park" (multi-intent) or "others except college" (obscure).

The crux of this issue lies in the model's ability to discern and align itself with the latest user intent, effectively disregarding or re-contextualizing the outdated information from the earlier conversation. Such a problem intensifies as conversation histories grow longer and changes become more frequent or subtle. The chatbot must continuously track the entire conversation, identify the shifts back and forth, and reconcile any discrepancies in the information flow. It requires advanced capabilities in contextual understanding, memory management, and dynamic response generation, pushing the boundaries of current NLP technologies.

Despite a myriad of tasks proposed to improve the coherence and consistency of chatbots, little attention has been paid to this unique and urgent issue—as LLMs continue to scale up impressively, it becomes prohibitive to integrate this task during training. Without a tailored dataset, it is impossible to gauge the LLMs' capability, let alone train them using RLHF (Ouyang et al. 2022) or DPO (Rafailov et al. 2023).

To this end, we briefly describe how to generate our dynamic dialogue state tracking (DynDST) dataset. We leverage the MultiWOZ 2.2 dataset (Zang et al. 2020) by identifying all the slots or text span in the dialogue, then we randomly choose one user utterance and alter one of its entities. As shown in Figure 1, college is selected in u_8, which, along with the subsequent bot response b_8, is considered a false turn. Next, we duplicate the false turn and make necessary changes to obtain the valid update turn (u_9 and b_9). Finally, we gather the question (u_{13}) inquiring whether the incorrect intent in u_8 has been overwritten.

The MultiWOZ dataset is a valuable resource for simulating human interactions in the dialogue state tracking (DST) task (Budzianowski et al. 2018; Eric et al. 2020; Zang et al. 2020). However, to evaluate an LLM's response accurately is challenging since numerous annotation errors persist throughout these datasets (Han et al. 2021; Ye, Manotumruksa, and Yilmaz 2022). We propose a multi-step evaluation framework to extract the user intent from an LLM response without Chain-of-Thought (CoT) or prompt engineering (Kojima et al. 2022). Our pipeline framework is CoT-free, flexible to capture typos, and can apply to proprietary and open-source LLMs: GPT (OpenAI 2023), Gemini (Team et al. 2023), Vicuna (Zheng et al. 2023), and Llama-2 (Touvron et al. 2023). The main contributions are:

- We introduce a practical and challenging task of chatbots to align with users' latest intents in real-world scenarios, where both updated and incorrect contexts coexist mid-conversation. We create a specialized 8k dataset and conduct extensive experiments on four LLMs to benchmark this advanced contextual understanding task.[1]
- We propose a pipeline, prompt-engineering free evaluation method to remove extraneous information from an LLM's response after an inquiry. Our method remains effective across LLMs even if they can only be inferenced through APIs. It addresses one of the notorious issues that LLMs are prone to generate superfluous sentences in response to open-domain questions and resolves the problem of the damaged gold labels in MultiWOZ 2.2.

[1]https://github.com/hhhuang/DynDST

2 Related Work

This paper lies at the intersection of knowledge editing (KE) and dialogue contradiction detection (DCD). We draw comparisons between those datasets and our DynDST (see Table 1). In Table 1, we count the number of data (# Data) as follows: In DECODE, we select the contradiction data in the dev and test set (i.e., verified by three annotators). In CareCall$_{mem}$, we choose the publicly released *English* version to match the language of others. In DIALFACT, we collect data labeled REFUTED in the validation and test set. In CDCONV, we report # Negative in their paper.

Knowledge Editing (KE) Dataset Meng et al. (2023) use the COUNTERFACT dataset (Meng et al. 2022, derived from Elazar et al. 2021) to evaluate their MEMIT framework, where the data is of the form $(\text{subject}, \text{relation}, \text{object})$ like (s=Michael Jordan, r=played_sport, o=basketball). Mitchell et al. (2022) create the Wikitext generation dataset to evaluate their MEND method on GPT-style models, where the data is a snippet of a paragraph. However, the text does not always directly contradict the other, so it may need further post-processing to ensure this, which is similar to the source paper of zsRE dataset (De Cao, Aziz, and Titov 2021).

Dialogue Contradiction Detection (DCD) Dataset Welleck et al. (2019) construct the DNLI dataset by using three approaches (*entity swap*, *relation swap*, and *numeric*), which is similar to our approach to generate DynDST (we swap the entity). Zheng et al. (2022) point out that it contains two isolated sentences, which is insufficient for capturing the contextual information in dialogue. Nie et al. (2021) create their DECODE dataset that one speaker deliberately contradicts what they said earlier in the conversation; they conduct *unstructured* and *structured* approaches for fine-tuning three models (Devlin et al. 2019; Liu et al. 2019; Clark et al. 2020). Zheng et al. (2022) state that most of the contradictions in Nie et al. (2021) fall into the category of *History Contradiction*, so they propose the CDCONV dataset and include two typical contradictions issues from the chatbot in their dataset: *Intra-sentence Contradiction* and *Role Confusion*. Nevertheless, the CDCONV dataset is not suitable for assessing LLMs' long-term capability.

3 Definition

Fact (Intent) The term fact is the text to be edited in a conversation throughout this paper, which is the intent (slot value) in the DST task (see the bold texts in Figure 1). Seeing that the MultiWOZ datasets solely focus on tracking personal status (e.g., booking a train), these facts or intents do not pertain to factual knowledge. We follow the form of fact in Meng et al. (2023), which is a tuple τ comprising subject, relation, and object. Given a fact x, we define another new fact x' is *valid* (i.e., semantically different) if

$$\tau(x') \neq \tau(x) \tag{1}$$

Specifically, the subject and relation of $\tau(x')$ and $\tau(x)$ are the same. For instance, the fact of u_8 in Figure 1 is $\tau(u_8) =$ (s=I, r=askAttractionType, o=college), while $\tau(u_9) =$ (s=I, r=askAttractionType, o=park). Note that the term valid also

Dataset	Format	Lang	F	¬ F	Valid	LT	(m, M)	# Data	Source
zsRE	S	en	✓	✗	✗	–	–	–	Levy et al. (2017)
FEVER	S	en	✓	✗	✓	–	–	–	Thorne et al. (2018)
Dialogue NLI (DNLI)	S	en	✗	✓	✓	–	–	–	Welleck et al. (2019)
COUNTERFACT	S	en	✓	✗	✓	–	–	–	Meng et al. (2022)
TruthfulQA	S	en	✓	✗	✓	–	–	–	Lin, Hilton, and Evans (2022)
Wikitext generation	P	en	✓	✗	✓	–	–	–	Mitchell et al. (2022)
DECODE	C	en	✗	✓	✓	✗	(4.4, 4.5)	4,121	Nie et al. (2021)
CareCall$_{mem}$	C	ko	✗	✓	✓	✓	(12.0, 11.5)†	3,581†	Bae et al. (2022)
DIALFACT	C	en	✓	✗	✓	✗	(2.7, 2.5)	7,298	Gupta et al. (2022)
CDCONV	C	zh	✗	✓	✓	✗	(2.0, 2.0)	4,351	Zheng et al. (2022)
DynDST (Ours)	C	en	✗	✓	✓	✓	(7.9, 8.0)	8,001	

† We report the English version

Table 1: An overview of KE and DCD datasets. There are three input formats: S (sentence), P (paragraph), or C (chat). Lang stands for language. F/¬ F column displays if the dataset contains factual/non-factual knowledge to be edited. Note that KE researchers only focus on factual datasets. Valid is defined in Section 3. LT stands for long-term. We regard the dataset as long-term so long as its average length is at least 5 turns (i.e., 10 utterances). The underlined checkmark (✓) denotes the source data that partially satisfies the property. We also report the mean (m) and median (M) number of turns in chat datasets.

encompasses the meaning of *specific*, for we exclude slot values that are (1) multi-intent ("museum and park") or (2) obscure ("others except college") during dataset generation. We use the terms fact, intent, and slot value interchangeably.

Conversation A conversation or dialogue with n turns is denoted as $(u_1, b_1, ..., u_n, b_n)$, where u_i and b_i is the user and bot utterance in the i-th turn, respectively. We focus on whether b_{n+1} aligns with the updated fact in a **multi-turn** conversation when a user inquire an open-domain question related to such fact in u_{n+1}, given a valid fact introduced within the user utterances $\{u_1, u_2, ..., u_n\}$. Let u_j be a valid fact, where $j \in [2, n]$, its previous user utterance u_{j-1} has an invalid (incorrect) fact. Hence, $\tau(u_j) \neq \tau(u_{j-1})$. In this paper, we also ensure that $\tau(b_j) \neq \tau(u_{j-1})$.[2] Naturally, a conversation can be classified into four categories (i.e., disjoint turns) in this task: *false, update, test,* and *previous turn*. In this task, it is straightforward in a multi-turn conversation since there exists a turn that has the incorrect context where the user wants to correct (u_{j-1}, b_{j-1}); there is another turn that the user updates the incorrect context (u_j, b_j); there is a turn where we want to evaluate an LLM's knowledge in this task (u_{n+1}, b_{n+1}); and there are other turns unrelated to the user update. As a result, we define: (1) The *false turn* contains a false intent. (2) The *update turn* has a user utterance that corrects the previous false intent. (3) The *test turn* is the question we aim to assess whether the LLM aligns with the latest user intent in the update turn. (4) The rest of the turns fall into the *previous turn*. For example, in Figure 1, u_8 and b_8 are in the false turn; u_9 and b_9 are in the update turn; the test turn contains u_{13}; and the rest of the utterances fall into the previous turn ($\{u_1, ..., b_7\} \cup \{u_{10}, ..., b_{12}\}$).

4 Dataset Generation

We first filter out data that does not have any labeled text span in user utterances in the MultiWOZ 2.2 training set. Setting the random seed to 0, we randomly select one user utterance for each data and obtain the first slot's value to edit. To obtain another valid entity, we gather the universal set of values from all training data, and then we randomly select one specific entity that is not in the current data. Mathematically speaking, let $\mathcal{D} = \{d_1, d_2, ...\}$ be the training set, $\mathcal{U}(\mathcal{D}) = \bigcup_i \mathcal{U}(d_i)$ be the set union of all slots' values in $\mathcal{D}$. For each d_i and its associated slot s_i with value v_i to be edited, another valid value v' is picked from $\mathcal{U}(\mathcal{D}) \setminus \mathcal{U}(d_i)$, where v' has the same slot name as s_i and $v' \neq v_i$.

After each data contains five valid intents, we duplicate the selected user utterance and the subsequent bot utterance and replace the original slot with new values. To further ensure the update turn does *not* contain the old intent, we (1) prepend five correction phrases to the user utterance (*Actually*, *In fact*, *In reality*, *As a matter of fact*, and *To tell the truth*; also see Section 6.1) and (2) check whether the old intent is still in the bot utterance, and we use the string replacement if necessary.[3] Finally, we insert the update turn after the false turn and generate a question related to such

[2] If not doing so, then a user may ask "Could you recommend a Japanese restaurant?" but the bot still responds "There's a Mexican restaurant located in...." in the update turn, which is undesirable.

[3] We do not apply string replacement for the rest of the existing future turn because we hardly observe the user and chatbot mention the original intent in the following dialogue *context*, in which we also believe MultiWOZ 2.2 ought to avoid such repetitions because of the DST's objective: tracking and memorizing the user intent. We often find the entity is either mentioned once or referred to using implicit terms like "that" or "same."

changes in the dialogue (one is in Figure 1; the rest are in the Appendix). The size of our DynDST dataset is 8,001.

5 Pipeline Evaluation Method for LLMs

Despite an increasing number of works leveraging LLMs to evaluate model outputs via prompt engineering or CoT, these approaches are often slow and lack explainability. On top of that, hand-crafted prompts do not generalize well across LLMs and are difficult to reproduce (costly per inference). Hence, we propose our multi-step exact match framework.

5.1 Annotation Errors and Canonical Form

If a slot entity has a "canonical" form, we can fix those typos and other non-trivial labels (e.g., synonyms) by leveraging LLMs trained with rich linguistic knowledge. For example, if the slot value has a typo (say, "Leister"), we can ask GPT-4 to do the answer mapping task (the possible values set is {Leicester, Ely, ...} in *train-departure*). Since we select valid intents from the universal set, these typos and non-trivial entities can be converted beforehand. Note that the slot value has its canonical form only if its name has possible values in MultiWOZ 2.2's *schema*. We provide the following template for GPT-4 to map a slot value to its canonical form:

> "Given slot name, please map the current answer into its corresponding label set from your knowledge. You must output the label only. Do not write explanation. \n Slot name: `[SN]` \n Label set: `[LS]` \n Current answer: `[CA]` \n Label of current answer: \n"

In this template, `[CA]` is filled with the slot value (which may be a typo or a synonym not presented in possible values in *schema*), `[SN]` is filled with its name, and `[LS]` is filled with its possible values. For we had pre-processed and removed all invalid intents (see Section 3), this answer mapping task is as simple as classifying "espensive" (a typo) or "high-priced" (a synonym) into {expensive, cheap, moderate}. Thus, we do not have to train a classification model from scratch, which may be essential if the connection between labels and their canonical forms is opaque.

5.2 The Stop Words Set of an LLM

We pre-generate each model's stop words set by obtaining the outputs in one of our baselines, then we tokenize all of them and remove any token that appears in the entire values set ($\mathcal{U}(\mathcal{D})$ in Section 4). After that, we regard tokens with frequency ≥ 100 as redundant (the cutoff is pre-defined). Note that even if we can generate each model's stop words set automatically, we still manually screen the result in case some tokens related to the gold labels are included. This and the NLTK stop words set are merged to form the final one for each LLM's output (see Step 4 in Section 5.3).

In this paper, each LLM and its families share the *same* stop words set, regardless of different experiments and configurations. For instance, GPT-3.5 and GPT-4 share the same stop words set, $\mathcal{S}_{GPT}$. All stop words set ($\mathcal{S}_{GPT}$, $\mathcal{S}_{Gemini}$, $\mathcal{S}_{Vicuna}$, and $\mathcal{S}_{Llama2}$) are generated and analyzed by 5 runs of Exp. 1 (40,005 outputs) in the default setting (see Section 6.2). $\mathcal{S}_{GPT}$, $\mathcal{S}_{Vicuna}$, and $\mathcal{S}_{Llama2}$ are constructed by GPT-3.5, Vicuna (13B), and Llama-2 (13B), respectively.

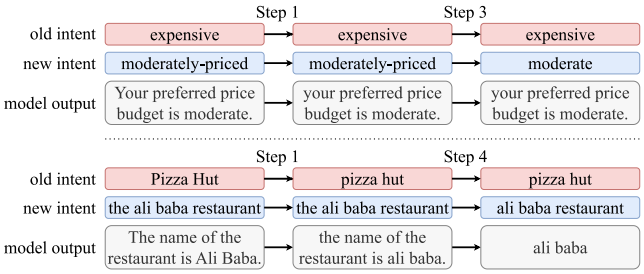

Figure 2: Two illustrative examples of our exact match pipeline. The new intent in the upper example has possible values in the MultiWoZ 2.2's schema ("moderately-priced" is converted to "moderate"), whereas the lower one does not.

5.3 Multi-Step Exact Match Framework

After an LLM's stop words set is available and every typos and synonyms have their canonical forms, we describe how we combine the exact match (EM), ROGUE-1 (R-1), and ROGUE-L (R-L) to implement the multi-step exact match:

1. Lowercase old intent (`old`), new intent (`new`), and model output (`output`).
2. Perform EM.
3. Convert `new` to its canonical form, then perform EM. (Objective: catch typos and synonyms in gold labels.)
4. Remove punctuation and stop words in `old`, `new`, and `output`. (Objective: remove redundant words so they will not interfere with the evaluation of R-1 and R-L.)
5. Perform EM.
6. Compute R-1 score of `new` (and `old`) with `output`. Label `output` as `new` only if (i) `new`'s F1 $>$ `old`'s F1 and (ii) either `new`'s precision or recall $>$ max{0.5, `old`'s precision or recall}. (Objective: catch long answers such as the address of *taxi-destination*.)
7. Compute the metric of edit distance (ED), longest common subsequence (LCSeq, aka R-L), and longest common substring (LCStr) of `new` (and `old`) with `output`. Label `output` as `new` only if (i) `new`'s ED $<$ `old`'s ED and (ii) `new`'s R-L $>$ `old`'s R-L and (iii) `new`'s LCStr $> \frac{1}{2}|\texttt{new}|$, where $|x|$ is the length of x. (Objective: should catch the typos finally.)

These steps may seem complex, but they are intuitive (see Figure 2): First, we use EM in Step 2. If EM can determine whether the model output contains either the new intent or the old, then we immediately return the result. Next, we attempt to convert the new intent to its canonical form and use EM again in Step 3. If it does not have possible values, we skip this step and proceed to Step 4, where we remove stop words as much as possible lest they interfere with our strict R-1 and R-L evaluation. In Step 4, we only remove the NLTK stop words in the old and new intent, whereas the model's output is further trimmed by its own stop words set. After this, we apply EM in Step 5 and check if removing these words is sufficient. If not, we resort to using *strict* R-1 and R-L in Steps 6 and 7, which is necessary because many slots, such as *restaurant-food* and *hotel-name*, are not categorical. In Steps 6 and 7, deciding whether the model output contains the old intent follows the same criteria. We do not use a stemmer when computing R-1, and it demonstrates

the rigor of our evaluation method. After Step 7, we classify the model's output as "N.A." if our evaluation cannot decide whether it has only the new intent or the old one.

6 Experimental Setup

Model and Evaluation Metric The configurations of LLMs are: GPT (gpt-4-0125-preview and gpt-3.5-turbo-0125), Gemini (gemini-1.0-pro-001), Vicuna (vicuna-33b-v1.3, vicuna-13b-v1.5-16k, and vicuna-7b-v1.5-16k), and Llama-2 (Llama-2-13b-chat-hf and Llama-2-7b-chat-hf). Half precision (FP16) is used in Vicuna and Llama-2 due to the limitation of computing resources. We set the temperature to 0 to maximize the reproducibility. Using our exact match pipeline, we report the accuracy (denoted as "Align"), which is used in KE tasks. "No Align" means the output only has an old intent. We ran each experiment five times to stabilize the results (Wang et al. 2023) using 4 RTX 3090 GPUs.

6.1 Multi-Turn Dynamic Alignment Framework

There are eight variants in aligning the latest user intent in the update and test turn: {implicit, explicit} negation of old intent in the update turn $\times$ {long, short} input in the update turn $\times$ {with, without} options in the test turn.

Two Types of Templates in Update Turn (Implicit or Explicit) Two types of user utterances in the update turn are whether it contains the negation of the old intent (*explicit*) or not (*implicit*). We test both scenarios by using the following ten templates (`[X]` and `[Y]` are the slots for the old and new user utterances, respectively). See Figure 3 for illustration.

1. Actually, `[Y]`
2. In fact, `[Y]`
3. In reality, `[Y]`
4. As a matter of fact, `[Y]`
5. To tell the truth, `[Y]`
6. I'm sorry to bring this up, but I mistakenly gave you `[X]`. In fact, `[Y]`
7. Oh, I'm sorry. Should have been `[Y]`, not `[X]`
8. Something is wrong with my previous statement. You can correct it by replacing `[X]` with `[Y]`
9. Wrong. It's not `[X]`, but `[Y]`
10. There's a problem with my previous statement. There's a mistake on `[X]`. It should be `[Y]`

Two Types of Inputs in Update Turn (Long or Short) We consider the naturalness of human conversations, so we propose another method when filling out the above templates. Specifically, it is natural (and concise) to express *only the change* rather than the entire utterance in our template. For example, in Figure 1, the user may say "Oh, I'm sorry. Should have been park, not college." or "In reality, park."

Two Types of Questions in Test Turn (with or without Options) We explore whether appending all possible values to the question helps these LLMs update a user's intent. For instance, the question with options provided in Figure 1 will be "What type of attraction am I interested in? (Options: architecture, boat, cinema, college, entertainment, museum, multiple sports, nightclub, park, swimming pool, theatre)."

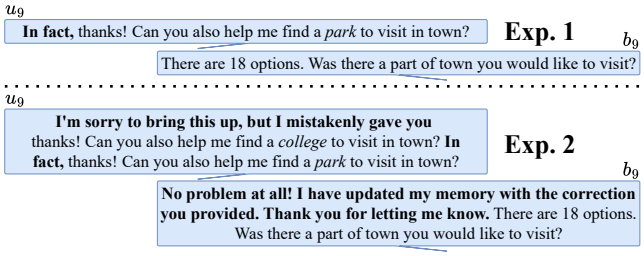

Figure 3: An illustrative difference between Exp. 1 and Exp. 2. Exp. 2 is an explicit update, as the user directly negates the original intent; we also prepend a "mock update" sentence in the bot utterance (bold text in b_9). **This figure is only for exposition** (default setting), as we also test other natural utterances like "I'm sorry to bring this up, but I mistakenly gave you college. In fact, park" (see Section 6.1).

6.2 Experiments and Ablation Analysis

Below are the four experiments (without update, implicit update, explicit update, adversarial attack of implicit update):

1. **Exp. 0**: We test the original MultiWOZ 2.2 data with the question appended to it (i.e., there is no update turn).
2. **Exp. 1**: We test the implicit update in the DynDST dataset (template index 1 to 5), as visualized in Figure 3.
3. **Exp. 2**: It is the explicit case of update (template index 6 to 10). We additionally insert the pre-defined sequences into the bot utterance, which acts as a *mock update*: "No problem at all! I have updated my memory with the correction you provided. Thank you for letting me know."
4. **Exp. 3**: This additional experiment is the adversarial attack of Exp. 1, as we swap the false turn and the update.

Align should be high in Exp. 1 and 2, whereas No Align should be high in Exp. 0 and 3. We experiment Exp. 1, Exp. 2, and Exp. 3 with the four variants or settings, which can be viewed as the ablation analysis (see Section 6.1): {long, short} input in the update turn $\times$ {with, without} options in the test turn. We define the **default** setting as "long input in the update turn" and "with options in the test turn."

7 Results and Discussion

We tabulate complete experiments of GPT-3.5 in Table 2. Table 3 is the result of GPT-4. The complete table results of the other LLMs (Gemini, Vicuna 7B, Vicuna 13B, Vicuna 33B, Llama-2 7B, and Llama-2 13B) are in the Appendix due to the page limit. For we define the default setting as "long input in the update turn" and "with options in the test turn" in Section 6.2, the other three ablation experiments are the removal of (a) options in the test turn, (b) long utterance in the update turn, and (c) both. We also report the upper bound performance of multiple CoTs. For example, in 3 runs of Exp. 2, we pick the top 3 templates (in the update turn) in Align, and we consider the LLM algin in this data if any of the three templates triggers it to output the latest user intent.

We present four aspects to analyze the complete experiments of an LLM in Table 2 (GPT-3.5 as an example):

GPT-3.5 (0125)	Align (Maj)			No Align (Maj)			N.A. (↓)			Upper Bound (↑)		
# run	1	3	5	1	3	5	1	3	5	1	3	5
Exp. 0 (MultiWOZ 2.2)	0.0	0.0	0.0	86.7	88.3	**88.8**	13.3	11.7	11.2	86.7	88.3	**88.8**
Exp. 1 (baseline, default)	66.6	72.4	**72.6**	20.7	19.7	20.6	12.7	7.9	6.8	66.6	80.5	**83.6**
(a) w/o options	62.6	67.7	69.4	20.3	20.9	21.8	17.1	11.4	8.8	62.6	76.3	79.7
(b) w/o long	46.7	51.0	50.5	39.7	39.8	42.5	13.6	9.2	7.0	46.7	67.7	73.6
(c) w/o both	42.3	48.2	47.1	39.6	39.9	43.5	18.1	11.9	9.4	42.3	61.7	67.1
Exp. 2 (Our, default)	73.7	79.9	**80.1**	13.8	13.2	14.6	12.5	6.9	5.3	73.7	86.2	**88.6**
(a) w/o options	70.4	75.1	78.6	16.3	15.4	14.4	13.3	9.5	7.0	70.4	82.4	86.3
(b) w/o long	71.0	75.8	76.9	16.0	15.9	17.1	13.0	8.3	6.0	71.0	84.0	88.2
(c) w/o both	68.8	74.3	76.5	13.8	15.7	16.1	17.4	10.0	7.4	68.8	81.7	86.3
Exp. 3 (Attack, default)	5.4	6.7	7.2	80.4	83.8	**84.8**	14.2	9.5	8.0	80.4	85.2	**86.1**
(a) w/o options	6.0	7.0	7.6	77.8	81.8	82.9	16.2	11.2	9.5	77.8	83.6	85.0
(b) w/o long	7.9	8.7	8.9	77.3	81.7	83.0	14.8	9.6	8.1	77.3	84.1	85.5
(c) w/o both	8.7	9.5	10.1	74.7	78.9	79.9	16.6	11.6	10.0	74.7	81.9	83.8

Table 2: Percentage of Align/No Align on DynDST dataset. Maj stands for majority voting. The ablation analyses are defined in Section 6.2. In Exp. 1 and Exp. 2, Align should be high; on the other hand, No Align should be high in Exp. 0 and Exp. 3.

GPT-4 (0125)	Align (Maj)			No Align (Maj)			N.A. (↓)			Upper Bound (↑)		
# run	1	3	5	1	3	5	1	3	5	1	3	5
Exp. 0 (MultiWOZ 2.2)	0.0	0.0	0.0	90.3	92.3	**92.8**	9.7	7.7	7.2	90.3	92.3	**92.8**
Exp. 1 (baseline, default)	58.4	64.2	**66.5**	24.5	24.8	26.2	17.1	11.0	7.3	58.4	76.2	**80.3**
Exp. 2 (Our, default)	70.3	74.6	**76.0**	15.9	16.6	17.9	13.8	8.8	6.1	70.3	83.3	**86.0**
Exp. 3 (Attack, default)	6.0	6.9	7.2	79.8	84.6	**86.8**	14.2	8.5	6.0	79.8	86.7	**88.8**

Table 3: Percentage of Align/No Align on DynDST dataset. We do not conduct the ablation analysis in GPT-4 due to the cost.

- Within one experiment: Investigate each LLM's overall performance in four different settings and the "width" of Align between the best and the worst. In an ideal situation, the "width" should be small. For instance, the width of GPT-3.5 in Exp. 2 is 3.6 in 5 runs, which is comparably small; surprisingly, it becomes 25.5 in Exp. 1. Though the width of Llama-2 (7B) in Exp. 2 is also small from this perspective (3.9), its best performance in Align is only 49.3, which is below the random guess baseline.
- Compare Exp. 1 and Exp. 2: The differences are in the update turn (see Figure 3). In an ideal situation, there should be no difference in Align. For instance, if we compare side by side, the best settings of Exp. 1 and Exp. 2 are the same (which is the default), and the performance boost is 7.5% in Align. Moreover, we find that there is a significant improvement in (c) setting (near 30% boost).
- Compare Exp. 1 and Exp. 3: The difference is the order of the update turn and the false turn. In an ideal situation, if a chatbot only replies based on the most recent heuristic, there should be no difference between Align of Exp. 1 and No Align of Exp. 3. There is a 12.2% gap in GPT-3.5.
- Compare Exp. 0 and Exp. 3: The difference is if an update intent is *falsely* inserted before the original intent, which tests if an LLM is robustly trained. In an ideal situation, there should be no difference in No Align.

We also report all LLMs' best results of Exp. 1 and Exp. 2 in Table 4 and visualized in Figure 4. First and foremost, **the best performance of Exp. 2 consistently outperforms that of Exp. 1 across all LLMs**, indicating that our approach, combined with the explicit negation of a false fact in the user utterance and the injected sequences in the bot utterance, boosts these LLMs to align with the new user intent in long-term conversation, even if the wrong context is within a conversation. It is simple yet effective, and the improvement of Vicuna (33B) is astonishing: a 17.5% boost in Align, almost on par with GPT-4 (76.0). The average boost in GPT, Gemini, and Vicuna (except 33B) is 8.26%. As for Llama-2 7B and 13B, the boost is only 5.4% and 3.9%, respectively.

When running our dataset five times and making decisions through majority voting, GPT-3.5 tends to capture the user update by more than 70% in Exp. 1 and slightly above 80% in Exp. 2. Moreover, if we compare results side by side, Exp. 2 consistently outperforms Exp. 1 across all settings, which also shows that the *worst* setting of Exp. 2 (76.5) still outperforms the *best* of Exp. 1 (72.6) in GPT-3.5. Intriguingly, we find that while there is a common belief that GPT-3.5 is bested by GPT-4 in every task, GPT-3.5 significantly outperforms GPT-4 in this advanced contextual understanding task (see Figure 5 for hypothesis).

# run	Align (↑, Maj) 1	3	5	No Align (↓, Maj) 1	3	5	Best Set
Exp. 1 (baseline)							
GPT-4	58.4	64.2	66.5	24.5	24.8	26.2	(d)
GPT-3.5	66.6	72.4	**72.6**	20.7	19.7	20.6	(d)
Gemini	53.8	57.3	59.9	25.4	26.0	26.0	(d)
Vicuna (33B)	48.6	56.4	57.6	26.1	28.5	29.9	(a)
Vicuna (13B)	56.8	61.6	65.1	23.6	25.0	24.6	(d)
Vicuna (7B)	46.4	54.6	57.2	26.8	29.3	29.8	(d)
Llama2 (13B)	45.3	55.6	58.9	22.2	23.8	24.2	(a)
Llama2 (7B)	31.7	40.2	43.9	20.5	23.3	24.4	(a)
Exp. 2 (Our)							
GPT-4	70.3	74.6	76.0	15.9	16.6	17.9	(d)
GPT-3.5	73.7	79.9	**80.1**	13.8	13.2	14.6	(d)
Gemini	59.9	68.4	68.3	16.7	18.6	20.6	(b)
Vicuna (33B)	67.8	74.1	75.1	15.3	16.9	17.5	(c)
Vicuna (13B)	69.0	72.0	71.8	15.6	17.3	19.7	(c)
Vicuna (7B)	60.8	65.2	66.4	17.2	19.0	19.9	(c)
Llama2 (13B)	51.0	60.5	62.8	20.6	20.2	21.1	(c)
Llama2 (7B)	38.8	46.7	49.3	16.2	17.1	19.4	(c)

Table 4: We report the best-performing setting of LLMs in Exp. 1 and Exp. 2. Set stands for setting. Note that (d) is the default setting. N.A. column is excluded here, so the sum of Align and No Align is not 100. We also conduct the complete experiments for Gemini using the *single-turn* input format. Surprisingly, the best performance in Exp. 1 and Exp. 2 are 73.3 and 80.2, respectively, which are on par with or slightly better than the multi-turn results of GPT-3.5.

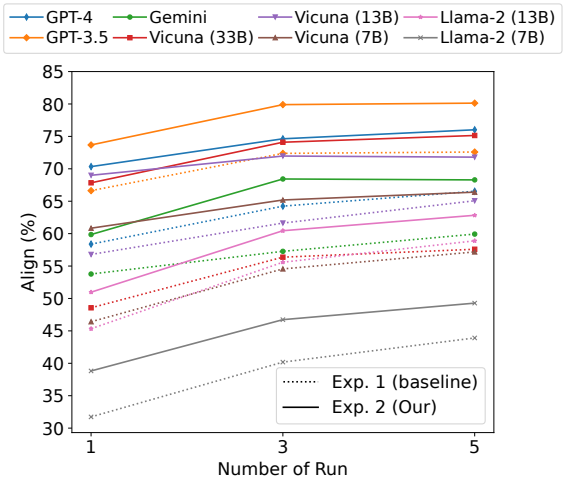

Figure 4: The best setting of Exp. 2 consistently outperforms that of Exp. 1 across all LLMs. GPT-4 is outrun by GPT-3.5 in both experiments. In Vicuna, 13B beats 33B in Exp. 1, while it is the other way around in Exp. 2. Vicuna (7B) is on par with Llama-2 (13B) in Exp. 1, outstripping it in Exp. 2.

In Vicuna, the overall performance increases as the model size increases in Exp. 2. However, this trend does not hold in Exp. 1, where the 13B model outperforms the 33B model. There are other interesting findings in Vicuna 7B and 13B LLMs: We observe that the best-performing setting in Exp. 1 for these models is the default setting. Conversely, in Exp. 2, the best-performing setting is (c), the most *natural* dialogue from a human perspective (only expresses the change and does not provide options in the test question). As for Llama-2, the best performance of Align in Exp. 1 is (a), whereas setting (c) performs best in Exp. 2.

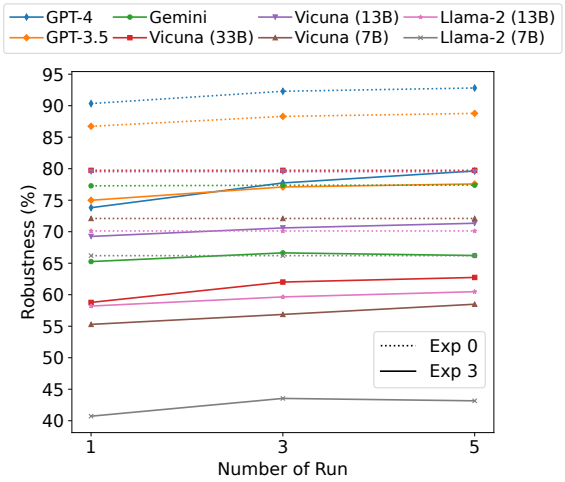

Figure 5: Robustness is defined as No Align − Align. We conjecture that GPT-4 is more robustly trained to alleviate malicious attacks and increase its safety. Nevertheless, this might hamper its ability to align with the latest user intent.

8 Conclusion

We tackle a key challenge limiting the practicality of contemporary LLMs for end users: aligning with the updated user intent when incorrect contexts persist mid-conversation. To gauge this conversational adaptability of LLMs on open-domain questions, we created an 8k dataset, DynDST. Built upon MultiWOZ 2.2, our dataset reflects the process of human interactions in real-world scenarios. Since LLMs tend to respond with lengthy sentences, we propose a multi-step framework for evaluation. Our exact match pipeline framework is robust across LLMs and does not rely on any CoT or prompt engineering. Through extensive experiments on four LLMs of varying sizes, we made several interesting observations: (1) A simple yet effective solution is to insert the negation of incorrect text in the update turn and a mock update in the bot utterance. (2) GPT-3.5 exceeds the performance of GPT-4 in both implicit and explicit update experiments (Exp. 1 and Exp. 2), possibly due to GPT-4 trained to mitigate malicious attacks. (3) In Gemini, the multi-turn input format significantly underperforms in both dynamic alignment experiments compared to the single-turn format, which performs slightly better than the multi-turn format of GPT-3.5 in both experiments. (4) Vicuna surpasses Llama-2 in this task, with its 33B model being on par with GPT-4 in Exp. 2. (5) In Exp. 2, Vicuna and Llama-2 unanimously perform best in the most natural conversational setting, while GPT-3.5 is better when provided with the entire user utterance and options. The results shed light on this novel and essential contextual understanding task from different perspectives. Our future work will be to apply these insights to DPO to build a better, well-aligned chatbot.

Ethical Statement

Any LLM should not be used for fact-checking, even though we analyze these LLMs' outputs as definite answers. When approaching this task, researchers should be aware of the difference between mechanical parroting and genuine understanding. Since our DynDST dataset is built upon the MultiWOZ 2.2, we do not foresee any ethical issues in it.

Acknowledgements

This work was supported by National Science and Technology Council, Taiwan, under the grant 112-2221-E-001-016-MY3 and by Academia Sinica, under the grant 236d-1120205.

References

Bae, S.; Kwak, D.; Kang, S.; Lee, M. Y.; Kim, S.; Jeong, Y.; Kim, H.; Lee, S.-W.; Park, W.; and Sung, N. 2022. Keep Me Updated! Memory Management in Long-term Conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 3769–3787. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Budzianowski, P.; Wen, T.-H.; Tseng, B.-H.; Casanueva, I.; Ultes, S.; Ramadan, O.; and Gašić, M. 2018. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 5016–5026. Brussels, Belgium: Association for Computational Linguistics.

Clark, K.; Luong, M.-T.; Le, Q. V.; and Manning, C. D. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *International Conference on Learning Representations*.

De Cao, N.; Aziz, W.; and Titov, I. 2021. Editing Factual Knowledge in Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 6491–6506. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Elazar, Y.; Kassner, N.; Ravfogel, S.; Ravichander, A.; Hovy, E.; Schütze, H.; and Goldberg, Y. 2021. Measuring and Improving Consistency in Pretrained Language Models. *Transactions of the Association for Computational Linguistics*, 9: 1012–1031.

Eric, M.; Goel, R.; Paul, S.; Sethi, A.; Agarwal, S.; Gao, S.; Kumar, A.; Goyal, A.; Ku, P.; and Hakkani-Tur, D. 2020. MultiWOZ 2.1: A Consolidated Multi-Domain Dialogue Dataset with State Corrections and State Tracking Baselines. In Calzolari, N.; Béchet, F.; Blache, P.; Choukri, K.; Cieri, C.; Declerck, T.; Goggi, S.; Isahara, H.; Maegaard, B.; Mariani, J.; Mazo, H.; Moreno, A.; Odijk, J.; and Piperidis, S., eds., *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 422–428. Marseille, France: European Language Resources Association. ISBN 979-10-95546-34-4.

Gupta, P.; Wu, C.-S.; Liu, W.; and Xiong, C. 2022. DialFact: A Benchmark for Fact-Checking in Dialogue. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3785–3801. Dublin, Ireland: Association for Computational Linguistics.

Han, T.; Liu, X.; Takanobu, R.; Lian, Y.; Huang, C.; Wan, D.; Peng, W.; and Huang, M. 2021. MultiWOZ 2.3: A multi-domain task-oriented dialogue dataset enhanced with annotation corrections and co-reference annotation. arXiv:2010.05594.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large Language Models are Zero-Shot Reasoners. In *Advances in Neural Information Processing Systems*, volume 35, 22199–22213.

Levy, O.; Seo, M.; Choi, E.; and Zettlemoyer, L. 2017. Zero-Shot Relation Extraction via Reading Comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, 333–342. Vancouver, Canada: Association for Computational Linguistics.

Lin, S.; Hilton, J.; and Evans, O. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3214–3252. Dublin, Ireland: Association for Computational Linguistics.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Meng, K.; Bau, D.; Andonian, A.; and Belinkov, Y. 2022. Locating and Editing Factual Associations in GPT. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 17359–17372. Curran Associates, Inc.

Meng, K.; Sharma, A. S.; Andonian, A. J.; Belinkov, Y.; and Bau, D. 2023. Mass-Editing Memory in a Transformer. In *The Eleventh International Conference on Learning Representations*.

Mitchell, E.; Lin, C.; Bosselut, A.; Finn, C.; and Manning, C. D. 2022. Fast Model Editing at Scale. In *International Conference on Learning Representations*.

Nie, Y.; Williamson, M.; Bansal, M.; Kiela, D.; and Weston, J. 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)*, 1699–1713. Online: Association for Computational Linguistics.

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray,

A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. arXiv:2203.02155.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Team, G.; Anil, R.; Borgeaud, S.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A. M.; Hauth, A.; Millican, K.; et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; and Mittal, A. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 809–819. New Orleans, Louisiana: Association for Computational Linguistics.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D.; Blecher, L.; Ferrer, C. C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A.; Hosseini, S.; Hou, R.; Inan, H.; Kardas, M.; Kerkez, V.; Khabsa, M.; Kloumann, I.; Korenev, A.; Koura, P. S.; Lachaux, M.-A.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet, X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X. E.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.; Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M.; Narang, S.; Rodriguez, A.; Stojnic, R.; Edunov, S.; and Scialom, T. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q. V.; Chi, E. H.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*.

Welleck, S.; Weston, J.; Szlam, A.; and Cho, K. 2019. Dialogue Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3731–3741. Florence, Italy: Association for Computational Linguistics.

Ye, F.; Manotumruksa, J.; and Yilmaz, E. 2022. MultiWOZ 2.4: A Multi-Domain Task-Oriented Dialogue Dataset with Essential Annotation Corrections to Improve State Tracking Evaluation. In Lemon, O.; Hakkani-Tur, D.; Li, J. J.; Ashrafzadeh, A.; Garcia, D. H.; Alikhani, M.; Vandyke, D.; and Dušek, O., eds., *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 351–360. Edinburgh, UK: Association for Computational Linguistics.

Zang, X.; Rastogi, A.; Sunkara, S.; Gupta, R.; Zhang, J.; and Chen, J. 2020. MultiWOZ 2.2 : A Dialogue Dataset with Additional Annotation Corrections and State Tracking Baselines. In Wen, T.-H.; Celikyilmaz, A.; Yu, Z.; Papangelis, A.; Eric, M.; Kumar, A.; Casanueva, I.; and Shah, R., eds., *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, 109–117. Online: Association for Computational Linguistics.

Zheng, C.; Zhou, J.; Zheng, Y.; Peng, L.; Guo, Z.; Wu, W.; Niu, Z.-Y.; Wu, H.; and Huang, M. 2022. CDConv: A Benchmark for Contradiction Detection in Chinese Conversations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 18–29. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; Zhang, H.; Gonzalez, J. E.; and Stoica, I. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.