Zero-Shot Prompting for Implicit Intent Prediction and Recommendation with Commonsense Reasoning

Hui-Chi Kuo **Yun-Nung Chen**
National Taiwan University, Taipei, Taiwan
r09922a21@csie.ntu.edu.tw y.v.chen@ieee.org

Abstract

The current generation of intelligent assistants require explicit user requests to perform tasks or services, often leading to lengthy and complex conversations. In contrast, human assistants can infer multiple implicit intents from utterances via their commonsense knowledge, thereby simplifying interactions. To bridge this gap, this paper proposes a framework for multi-domain dialogue systems. This framework automatically infers implicit intents from user utterances, and prompts a large pre-trained language model to suggest suitable task-oriented bots. By leveraging commonsense knowledge, our framework recommends associated bots in a zero-shot manner, enhancing interaction efficiency and effectiveness. This approach substantially reduces interaction complexity, seamlessly integrates various domains and tasks, and represents a significant step towards creating more human-like intelligent assistants that can reason about implicit intents, offering a superior user experience.[1]

1 Introduction

Intelligent assistants have become increasingly popular in recent years, but they require users to *explicitly* describe their tasks within a *single* domain. Yet, the exploration of gradually guiding users through individual task-oriented dialogues has been relatively limited (Chiu et al., 2022). This limitation is amplified when tasks extend across multiple domains, compelling users to interact with numerous bots to accomplish their goals (Sun et al., 2016). For instance, planning a trip might involve interacting with one bot for flight booking and another for hotel reservation, each requiring distinct, task-specific intentions like "*Book a flight ticket*" to activate the corresponding bot, such as an airline bot. In contrast, human assistants can manage high-level intentions spanning *multiple* domains, utilizing commonsense knowledge. This approach renders conversations more pragmatic and efficient, reducing the user's need to deliberate over each task separately. To overcome this limitation of current intelligent assistants, we present a flexible framework capable of recommending task-oriented bots within a multi-domain dialogue system, leveraging commonsense-inferred *implicit* intents as depicted in Figure 1.

Multi-Domain Realization Sun et al. (2016) pinpointed the challenges associated with a multi-domain dialogue system, such as 1) comprehending single-app and multi-app language descriptions, and 2) conveying task-level functionality to users. They also gathered multi-app data to encourage research in these directions. The HELPR framework (Sun et al., 2017) was the pioneering attempt to grasp users' multi-app intentions and consequently suggest appropriate individual apps. Nevertheless, previous work focused on understanding individual apps based on high-level descriptions exclusively through user behaviors, necessitating a massive accumulation of personalized data. Due to the lack of paired data for training, our work leverages external commonsense knowledge to bridge the gap between high-level utterances and their task-specific bots. This approach enables us to consider a broad range of intents for improved generalizability and scalability.

Commonsense Reasoning Commonsense reasoning involves making assumptions about the nature and essence of typical situations humans encounter daily. These assumptions encompass judgments about the attributes of physical objects, taxonomic properties, and individuals' intentions. Existing commonsense knowledge graphs such as ConceptNet (Bosselut et al., 2019), ATOMIC (Sap et al., 2019), and TransOMCS (Zhang et al., 2021) facilitate models to reason over human-annotated commonsense knowledge. This paper utilizes a

[1] Code: http://github.com/MiuLab/ImplicitBot.

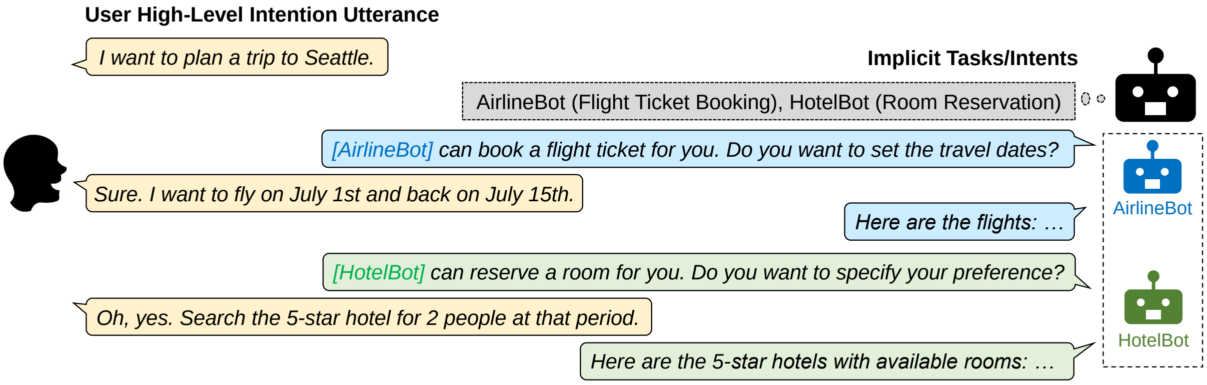

User High-Level Intention Utterance
I want to plan a trip to Seattle.
Implicit Tasks/Intents
AirlineBot (Flight Ticket Booking), HotelBot (Room Reservation)
[AirlineBot] can book a flight ticket for you. Do you want to set the travel dates?
Sure. I want to fly on July 1st and back on July 15th.
Here are the flights: …
AirlineBot
[HotelBot] can reserve a room for you. Do you want to specify your preference?
Oh, yes. Search the 5-star hotel for 2 people at that period.
Here are the 5-star hotels with available rooms: …
HotelBot

Figure 1: Illustration of a multi-task dialogue example.

generative model trained on ATOMIC$^{20}_{20}$ (Hwang et al., 2021) to predict potential intents linking given user high-level utterances with corresponding task-oriented bots. The inferred intents can activate the relevant task-oriented bots and also serve as justification for recommendations, thereby enhancing explainability. This work is the first attempt to integrate external commonsense relations with task-oriented dialogue systems.

Zero-Shot Prompting Recent research has revealed that large language models (Radford et al., 2019; Brown et al., 2020) have acquired an astounding ability to perform few-shot tasks by using a natural-language prompt and a handful of task demonstrations as input context (Brown et al., 2020). Guiding the model with interventions via an input can render many downstream tasks remarkably easier if those tasks can be naturally framed as a cloze test problem through language models. As a result, the technique of prompting, which transposes tasks into a language model format, is increasingly being adopted for different tasks (Zhao et al., 2021; Schick and Schütze, 2021). Without available data for prompt engineering (Shin et al., 2020), we exploit the potential of prompting for bot recommendation in a zero-shot manner. This strategy further extends the applicability of our proposed framework and enables it to accommodate a wider variety of user intents and tasks, thus contributing to a more versatile and efficient multi-domain dialogue system.

2 Framework

Figure 2 illustrates our proposed two-stage framework, which consists of: 1) a commonsense-inferred intent generator, and 2) a zero-shot bot recommender. Given a user's high-level intention utterance, the first component focuses on generating implicit task-oriented intents. The second component then utilizes these task-specific intents to recommend appropriate task-oriented bots, considering the bots' functionality through a large pre-trained language model.

2.1 Commonsense-Inferred Intent Generation

The commonsense-inferred implicit intents function not only as prompts for bot recommendation but also as rationales for the suggested bots, thereby establishing a solid connection between the high-level intention and task-oriented bots throughout the conversation. For instance, the multi-domain system shown in Figure 1 recommends not only the *AirlineBot* but also describes its functionality—"*can book a flight ticket*"—to better convince the user about the recommendation.

2.1.1 Relation Trigger Selection

ATOMIC$^{20}_{20}$ is a commonsense knowledge graph featuring commonsense relations across three categories: social-interaction, event-centered, and physical-entity relations, all of which concern situations surrounding a specified event of interest. Following Hwang et al. (2021), we employ a BART model (Lewis et al., 2020) pre-trained on ATOMIC$^{20}_{20}$ to generate related entities and events based on the input sentence. However, despite having a total of 23 commonsense relations, not all are suitable for inferring implicit intents in assistant scenarios. We utilize AppDialogue data (Sun et al., 2016) to determine which commonsense relations can better trigger the task-specific intents. Given a high-level intention description u_i and its task-specific sentences s_{ij}, we calculate the trigger score of each relation r as an indicator of its

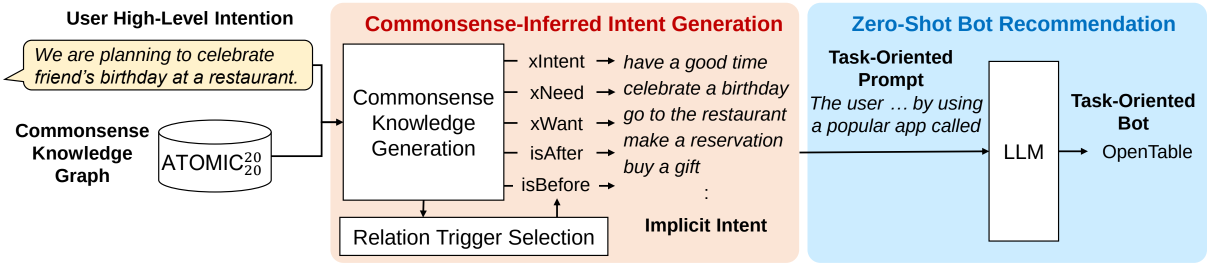

User High-Level Intention
We are planning to celebrate friend's birthday at a restaurant.
Commonsense Knowledge Graph
ATOMIC$^{20}_{20}$
Commonsense-Inferred Intent Generation
Commonsense Knowledge Generation
xIntent → have a good time
xNeed → celebrate a birthday
xWant → go to the restaurant
isAfter → make a reservation
isBefore → buy a gift
:
Relation Trigger Selection
Implicit Intent
Zero-Shot Bot Recommendation
Task-Oriented Prompt
The user … by using a popular app called
LLM
Task-Oriented Bot
OpenTable

Figure 2: Our zero-shot framework for triggering task-oriented bots via the commonsense-inferred prompts.

	Relation	Definition
Social	**xIntent**	the likely intent or desire of an agent (X) behind the execution of an event "X gives Y gifts" → X wanted "to be thoughtful"
	xNeed	a precondition for X achieving the event "X gives Y gifts" → X must first "buy the presents"
	xWant	post-condition desires on the part of X "X gives Y gifts" → X may also desire "to hug [Y]"
Event	**isAfter**	events that can precede an event "X is in a hurry to get to work" → "X wakes up late"
	isBefore	events that can follow an event "X is in a hurry to get to work" → "X drives too fast"

Table 1: Selected relations from ATOMIC$^{20}_{20}$.

suitability as a trigger relation.

$$T(r) = \sum_{i}\sum_{j} P_{BART}([u_i, r, s_{ij}]), \qquad (1)$$

where $P_{BART}([u_i, r, s_{ij}])$ represents the probability of the sentence beginning with the high-level user description u_i, followed by a relation trigger r, and the corresponding task-specific sentences s_{ij}. By summing up multiple task-specific sentences over j and all samples over i, a higher $T(r)$ implies that the relation r can better trigger implicit task-oriented intents in assistant scenarios.

We identify a total of five relations with the highest $T(r)$ and present their definitions (Sap et al., 2019) in Table 1. These relations are also reasonable from a human perspective to trigger implicit user intents.

2.1.2 Commonsense Knowledge Generation

Given the selected relations $R = \{r_1, r_2, ..., r_5\}$, where r_i represents the i-th relation from **{xIntent, xNeed, xWant, isAfter, isBefore}**, we concatenate each relation with a user utterance u to serve as the context input for our pre-trained BART model:

$$\texttt{<s>} \;\; u \;\; r_i \;\; \texttt{[GEN]} \;\; \texttt{</s>},$$

where `<s>` and `</s>` are special tokens in BART, and `[GEN]` is a unique token employed during the pre-training of BART to initiate the commonsense-related events. BART accepts this input and decodes the commonsense events into implicit task-oriented intents $Y = y^1_{1:k}, y^2_{1:k}, ..., y^5_{1:k}$, where y^i_k denotes the k-th generated commonsense event of the relation r_i.

2.2 Zero-Shot Bot Recommendation

With the inferred intents, the second component aims to recommend appropriate bots capable of executing the anticipated tasks. To pinpoint the task-specific bots based on the required functionality, we leverage the remarkable capacity of a large pre-trained language model, assuming that app descriptions form a part of the pre-trained data.

2.2.1 Pre-trained Language Model

The language model used in this study is GPT-J 6B[2], an GPT-3-like causal language model trained on the Pile dataset[3] (Radford et al., 2019), a diverse, open-source language modeling dataset that comprises 22 smaller, high-quality datasets combined together. Making the assumption that app descriptions in mobile app stores are incorporated in the pre-training data, we exploit the learned language capability to suggest task-oriented bots based on the given intents.

2.2.2 Prompting for Bot Recommendation

To leverage the pre-trained language capability of GPT-J, we manually design prompts for each relation type. For social-interaction relations, the prompt is formulated as "*The user r_i $y^i_{1:k}$ by using a popular app called*". For instance, Figure 2 generates a prompt "*The user needs to go to the restaurant and make the reservation by using a popular app called*". For event-centered relations, we

[2] https://huggingface.co/EleutherAI/gpt-j-6B
[3] https://pile.eleuther.ai/

Method	Precision	Recall	F1	Human Score (Mean±STD)
1-Stage Prompting Baseline	30.3	20.6	23.7	1.73±1.03
2-Stage Prompting (GPT-3)	28.6	**41.7**	31.8	2.11±0.46
Proposed 2-Stage (COMeT)	**36.0**	35.7	**32.9**	**2.18±0.34**
Proposed 2-Stage (COMeT) w/o Reasons	-	-	-	2.15±0.35
Gold				*2.44±0.27*

Table 2: Evaluation scores (%).

User Input	*We are planning to celebrate friend's birthday at a restaurant in [City].*	**Score**
User-labeled	Line (Communication), Google Maps (Maps & Navigation), Calendar (Productivity)	2.25
1-Stage Prompting	Tinder (Lifestyle), Grindr (Lifestyle)	1.83
2-Stage Prompting	Zomato can help to book the restaurant in advance. WhatsApp can find out about their contact information.	2.00
Proposed 2-Stage	WhatsApp can help have a good time and to celebrate a friend's birthday OpenTable can help book a table at the restaurant and go to the restaurant.	**2.67**
w/o Reasons	WhatsApp (Communication), OpenTable (Food & Drink)	2.17

Table 3: Generated results for given user high-level descriptions.

simply concatenate the generated events and app-prompt to trigger the recommended task-oriented apps/bots.

3 Experiments

To evaluate the zero-shot performance of our proposed framework, we collected a test set specific to our multi-domain scenarios. We recruited six volunteers who were knowledgeable about the target scenarios to gather their high-level intention utterances along with the associated task-oriented bots. Upon filtering out inadequate data, our test set incorporated a total of 220 task-oriented bots and 92 high-level utterances, each linked with an average of 2.4 bots. The number of bot candidates considered in our experiments is 6,264, highlighting the higher complexity of our tasks.

Our primary aim is to connect a high-level intention with its corresponding task-oriented bot recommendation by leveraging external commonsense knowledge. Therefore, we assess the effectiveness of the proposed methodology and compare it with a 1-stage prompting baseline using GPT-J to maintain fairness in comparison. For this baseline, we perform simple prompting on the user's high-level utterance concatenating with a uniform app-based prompt: "*so I can use some popular apps called.*" In response to these context prompts, GPT-J generates the associated (multiple) app names, serving as our baseline results.

To further investigate whether our proposed commonsense-inferred implicit intent generator is suitable for our recommendation scenarios, we introduce another 2-stage prompting baseline for comparison. Taking into account that contemporary large language models exhibit astonishing proficiency in commonsense reasoning, we substitute our first component with the state-of-the-art GPT-3 (Brown et al., 2020) to infer implicit intents, serving as another comparative baseline.

3.1 Automatic Evaluation Results

Considering that multiple bots can fulfill the same task (functionality), we represent each app by its category as defined on Google Play, then compute precision, recall, and F1 score at the *category* level. This evaluation better aligns with our task objective; for instance, both "*WhatsApp*" and "*Line*" belong to the same category—"communication" as demonstrated in Table 3.

Table 2 presents that the 2-stage methods significantly outperform the 1-stage baseline, suggesting that commonsense knowledge is useful to bridge high-level user utterances with task-oriented bots. Further, our proposed approach, which leverages external commonsense knowledge, achieves superior precision over GPT-3, a quality that is more important in recommendation scenarios. The reason is that GPT-3 may generate hallucinations for inferring more diverse but may not suitable intents.

3.2 Human Evaluation Results

Given that our goal can be interpreted as a recommendation task, the suggested bots different from user labels can be still reasonable and useful to users. Therefore, we recruited crowd workers from

Method	Win	Lose	Tie
Ours vs. 2-Stage Prompt (GPT-3)	57.6	40.2	2.2
Ours vs. Ours w/o Reasons	55.1	38.8	6.1

Table 4: Pair-wise human preference results (%).

Amazon Mechanical Turk (AMT) to evaluate the relevance of each recommended result given its high-level user utterance. Each predicted bot or app is assessed by three workers on a three-point scale: **irrelevant** (1), **acceptable** (2), and **useful** (3). The human-judged scores are reported in the right part of Table 2, and our proposed framework achieves the average score of 2.18, implying that most recommended tasks are above acceptable. Compared with the 1-stage baseline with a score below 2, it demonstrates that commonsense inferred implicit intents can more effectively connect the reasonable task-oriented bots. Considering that the score of 2-stage prompting is also good, we report the pairwise comparison in Table 4, where we can see that humans prefer ours to 2-stage prompting baseline for 57% of the data.

In additon to simply suggesting task-oriented bots, providing the rationale behind their recommendation could help users better judge their utility. Within our proposed framework, the commonsense-inferred implicit intents, which are automatically generated by the first component, can act as the explanations for the recommended task-oriented bots, as illustrated in Table 3. Consequently, we provide these rationales alongside the recommended results using the predicted intents and undergo the same human evaluation process. Table 4 validates that providing these justifications results in improved performance from a human perspective, further suggesting that commonsense-inferred intents are useful not only for prompting task-oriented bots but also for generating human-interpretable recommendation.

4 Discussion

Table 5 showcases the implicit intents generated by our proposed COMeT generator and GPT-3. It is noteworthy that GPT-3 occasionally produces hallucinations, which can render the recommended bots unsuitable. For instance, given the text prompt "*My best friend likes pop music.*", GPT-3 infers an intent to "*buy a ticket to see Justin Bieber*", which may not align accurately with the user's need.

Generated Intent Example	
Input	*My best friend likes pop music.*
COMet	Want → to listen to music Intent → to be entertained Need → to listen to music
GPT-3	Want → to get her tickets to see Justin Bieber for her birthday Intent → to buy her a CD by Taylor Swift for her birthday Need → to find songs that are pop and appropriate for her
Input	*I am looking for a job.*
COMet	Want → to apply for a job Intent → to make money Need → to apply for a job
GPT-3	Want → to learn more Intent → to apply for a job Need → to update my resume

Table 5: Generated commonsense-inferred intents.

Hence, our experiments reveal that while the 2-stage prompting achieves higher recall, its precision is lower. As our objective is to recommend reasonable task-specific bots, a higher precision is more advantageous in our scenarios.

5 Conclusion

This paper introduces a pioneering task centered around recommending task-oriented dialogue systems solely based on high-level user intention utterances. The proposed framework leverages the power of commonsense knowledge to facilitate zero-shot bot recommendation. Experimental results corroborate the reasonability of the recommended bots through both automatic and human evaluations. Experiments show that the recommended bots are reasonable for both automatic and human evaluation, and the inferred intents can provide informative and interpretable rationales to better convince users of the recommendation for practical usage. This innovative approach bridges the gap between user high-level intention and actionable bot recommendations, paving the way for a more intuitive and user-centric conversational AI landscape.

Limitations

This paper acknowledges three main limitations: 1) the constraints of a zero-shot setting, 2) an uncertain generalization capacity due to limited data in the target task, and 3) the longer inference time required by a large language model.

Given the absence of data for our task and the complexity of the target scenarios, collecting a large dataset for supervised or semi-supervised learning presents a significant challenge. As the first approach tackling this task, our framework performs the task in a zero-shot manner, but is applicable to fine-tuning if a substantial dataset becomes available. Consequently, we expect that future research could further train the proposed framework using supervised learning or fine-tuning, thereby enhancing the alignment of inferred implicit intents and recommended bots with training data. This would expand our method to various learning settings and validate its generalization capacity.

Conversely, the GPT-J model used for recommending task-oriented bots is considerably large given academic resources, thereby slowing down inference speed. To mitigate this, our future work intends to develop a lightweight student model that accelerates the prompt inference process. Such a smaller language model could not only expedite the inference process to recommend task-oriented bots but also be conveniently fine-tuned using collected data.

Despite these limitations, this work can be considered as the pioneering attempt to leverage commonsense knowledge to link task-oriented intents. The significant potential of this research direction is evidenced within this paper.

Ethics Statement

This work primarily targets the recommendation of task-oriented bots, necessitating a degree of personalization. To enhance recommendation effectiveness, personalized behavior data may be collected for further refinement. Balancing the dynamics between personalized recommendation and privacy is a critical consideration. The data collected may contain subjective annotations, and the present paper does not dive into these issues in depth. Future work should address these ethical considerations, ensuring an balance between personalized recommendations and privacy preservation.

Acknowledgements

We thank the reviewers for their insightful comments. This work was financially supported by the Young Scholar Fellowship Program by the National Science and Technology Council (NSTC) in Taiwan, under Grants 111-2222-E-002-013-MY3 and 111-2628-E-002-016.

References

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Ssu Chiu, Maolin Li, Yen-Ting Lin, and Yun-Nung Chen. 2022. SalesBot: Transitioning from chit-chat to task-oriented dialogues. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6143–6158.

Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. RICO: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 845–854.

Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. (Comet-) Atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6384–6392.

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.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3027–3035.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235.

Ming Sun, Yun-Nung Chen, Zhenhao Hua, Yulian Tamres-Rudnicky, Arnab Dash, and Alexander Rudnicky. 2016. AppDialogue: Multi-app dialogues for intelligent assistants. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3127–3132.

Ming Sun, Yun-Nung Chen, and Alexander I Rudnicky. 2017. HELPR: A framework to break the barrier across domains in spoken dialog systems. In *Dialogues with social robots*, pages 257–269. Springer.

Hongming Zhang, Daniel Khashabi, Yangqiu Song, and Dan Roth. 2021. TransOMCS: from linguistic graphs to commonsense knowledge. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4004–4010.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

A Implementation Details

In our zero-shot bot recommendation experiments, which are evaluated using Android apps based on RICO data (Deka et al., 2017), we append the phrase "*in Android phone*" to all prompts. This helps guide the resulting recommendations. Task-oriented prompts are fed into GPT-J to generate token recommendations for bots/apps, such as "*OpenTable*", an Android app, which aligns better with our evaluation criteria.

In the 2-stage prompting baseline, our prompts for GPT-3, designed to generate commonsense-related intents, are coupled with our selected relations to ensure a fair comparison. These prompts are outlined in Table 6.

B Reproducibility

To enhance reproducibility, we release our data and code. Detailed parameter settings employed in our experiments are as follows.

In commonsense knowledge generation, we apply beam search during generation, setting *beam_size=10*. In prompting for bot recommendation, a sampling strategy is implemented during recommendation generation, with *max_length=50*, *temperature=0.01*, and *top_p=0.9*.

	Relation	**GPT-3 Prompt**
Social	**xIntent**	so I intend
	xNeed	so I need
	xWant	so I want
Event	**isAfter**	Before, the user needs to
	isBefore	After, the user needs to

Table 6: Designed prompts of GPT-3. The prompts are converted from selected relations of ATOMIC$^{20}_{20}$ for a fair comparison.

C Crowdsourcing Interface

Figure 3 and 4 display annotation screenshots for both types of outputs. Workers are presented with a recommendation result from 1) user-labeled ground truth, 2) the baseline, and 3) our proposed method. Note that results accompanied by reasons originate only from our proposed method.

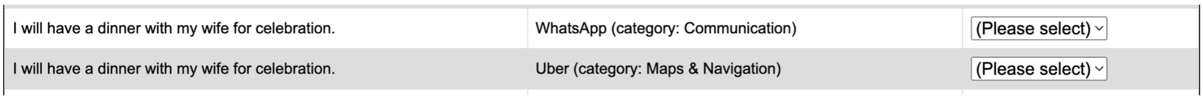
D Qualitative Analysis

Table 7 features additional examples from our test set, highlighting our method's ability to use commonsense knowledge to recommend more appropriate apps than the baseline, and broaden user choices.

In the first example, our method discerns the user's financial needs and suggests relevant financial apps such as *Paypal*. Conversely, the baseline method could only associate the user's needs with communication apps like *WeChat*, possibly influenced by the term *friend* in the high-level description.

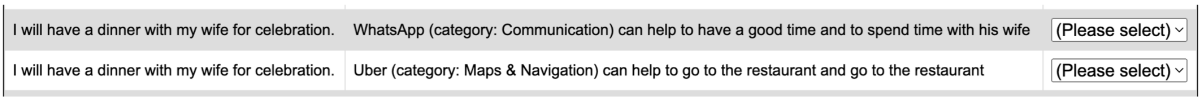
In the second example, our method infers potential user intents about checking their bank account and purchasing a new notebook, thus recommending *Paypal* for bank account management and *Amazon* for shopping.

In the third example, the user mentions having a tight schedule. Hence, our method suggests *Uber* to expedite the user's commute to the movie theater or *Netflix* for instant access to movies.



I will have a dinner with my wife for celebration.	WhatsApp (category: Communication)	(Please select)
I will have a dinner with my wife for celebration.	Uber (category: Maps & Navigation)	(Please select)

Figure 3: An annotation screenshot of annotating the recommended apps/bots on the Amazon Mechanical Turk, where the results may come from the ground truth, the baseline, or the proposed method.



I will have a dinner with my wife for celebration.	WhatsApp (category: Communication) can help to have a good time and to spend time with his wife	(Please select)
I will have a dinner with my wife for celebration.	Uber (category: Maps & Navigation) can help to go to the restaurant and go to the restaurant	(Please select)

Figure 4: An annotation screenshot of annotating the recommended apps/bots together with the predicted intents as reasons on the Amazon Mechanical Turk.

Data Example	
User Input	*Check if my friend sent the money to me.*
User-labeled	Bank (Finance), Messenger (Communication)
Baseline	WhatsApp (Communication), WeChat (Communication)
Proposed	Google Wallet (Finance), WhatsApp (Communication), Paypal (Finance)
Reasons	Google Wallet can help check if the money was sent to the right place and check if the money was sent to the correct place WhatsApp can help find out where the money came from and find out who sent the money Paypal can help to give the money to my friend and to give the money to the person who sent it to me
User Input	*My notebook was broken. I need to get a new one. Check how much money is left in my account.*
User-labeled	Shopee (Shopping)
Baseline	Google Play (Google Play)
Proposed	Google Play (Google Play), Amazon (Shopping), Mint (Tools), Paypal (Finance)
Reasons	Google Play can help to buy a new one and to buy a new notebook. Amazon can help to buy a new one and find out how much money is left. Mint can help to buy a new one and to buy a new notebook. PayPal can help my credit card is maxed out and my credit card is maxed out and I can't afford a new one.
User Input	*I really like watching movie, but my schedule is so tight.*
User-labeled	Calendar (Productivity), Movies (Entertainment)
Baseline	MovieBox (Entertainment)
Proposed	WhatsApp (Communication), Netflix (Entertainment), Youtube (Media), Uber (Maps & Navigation)
Reasons	WhatsApp can help to be entertained and to have fun. Netflix can help find a movie to watch and find a movie to watch. Youtube can help go to the movies and to find a movie to watch. Uber can help when you have a lot of work to do and have to go to work.

Table 7: Generated results for given user high-level descriptions.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
The separate section after the main paper

☑ A2. Did you discuss any potential risks of your work?
The separate section after the main paper

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

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

B ☒ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
Not applicable. Left blank.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

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

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

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

C ☑ Did you run computational experiments?

Section 3

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

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

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix B

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Table 2 and Section 3

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 3

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Section 3

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

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Section 3

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Section 3

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Section 3

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
The annotators are recruited from the platform, and their characteristics cannot be accurately identified.