An Improved, Strong Baseline for Pre-Trained Large Language Models as Task-Oriented Dialogue Systems

Sebastian Steindl[1] **André Kestler**[1] **Ulrich Schäfer**[1] and **Bernd Ludwig**[2]
[1] Ostbayerische Technische Hochschule Amberg-Weiden, Germany
[2] University Regensburg, Germany
{s.steindl, a.kestler, u.schaefer}@oth-aw.de
bernd.ludwig@ur.de

Abstract

Large Language Models (LLMs) have recently been studied within the context of Task-Oriented Dialogues. However, previous research is inconclusive on their effectiveness, with some studies claiming that LLMs are unable to perform the Task-Oriented Dialogue (TOD) task and others making sophisticated additions to their setup and coming to opposite conclusions. In this work, we take a detailed look at previous results that state LLMs perform insufficiently as a TOD system. As a result, we propose an updated, stronger baseline for multiple out-of-the-box LLM performances as TOD systems. We introduce a Self-Checking mechanism as a simple, yet effective, component to drastically improve their performance. Our results show that newer, pre-trained LLMs can, in fact, perform as TOD systems out-of-the-box, challenging the previous understanding. We show that LLMs can even perform competitively to fine-tuned models in certain metrics. Based on this, we propose directions for future research. Our code is published on Github[1].

1 Introduction

TOD Systems are a special case of chatbots, where the system helps the user achieve a certain task. Over the last years, they have undergone multiple evolutionary steps from rule-based multi-model approaches to end-to-end Deep-Learning based models. LLMs have shown remarkable out-of-the-box performance on many tasks, adapting to a task merely by prompting. Naturally, they have also been investigated as TOD systems, due to their strong Natural Language Generation (NLG) and Natural Language Understanding (NLU) abilities.

However, initial investigations into LLMs as TOD systems concluded that their out-of-the-box performance is insufficient (Hudeček and Dusek, 2023). Later, studies added sophisticated mechanisms, continued training the LLM, and optimized for Dialogue State Tracking (DST), showing largely improved performance (Hu et al., 2022; Lee and Lee, 2024; Dong et al., 2024).

[1] https://github.com/sebastian-steindl/LLM4TOD_baseline

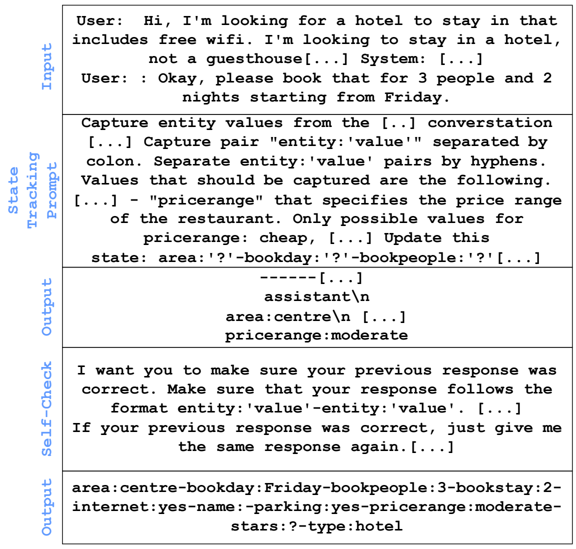

Figure 1: Example of the state tracking with Self-Check.

We provide an updated, strong baseline for the general performance of pre-trained LLMs, i.e., without sophisticated mechanisms, on the MultiWOZ 2.2 (Zang et al., 2020) benchmark dataset, even achieving comparable results to the fine-tuned SOTA in some cases. Based on our empirical evaluation, we propose future directions for this field of research. To the best of our knowledge, we are the first to give an extensive overview across multiple LLMs on their baseline, zero-shot performance on the DST and Response Generation (RG) task.

2 Background and Related Work

Task-Oriented Dialogue Systems. Task-Oriented Dialogue Systems are a special case of chatbots designed to help the user achieve one or multiple

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 11388–11398
November 4-9, 2025 ©2025 Association for Computational Linguistics

tasks by interacting with external services. Common use cases include booking scenarios, where the chatbot gathers the required information from the user and presents him with information from, e.g., databases. A TOD system thus needs capabilities in understanding and generating natural language and in deciding on a policy. Traditionally, these systems had a dedicated model for each of those subtasks (Young et al., 2013). With the progress in Deep Learning and the publication of larger datasets, this has shifted towards fine-tuned transformer-based models, often approaching the task in an end-to-end manner. The most prevalent crowdsourced dataset is MultiWOZ (Budzianowski et al., 2018), of which multiple updates exist, including MultiWOZ 2.2 (Zang et al., 2020). Deep Learning approaches to this benchmark include, e.g., Lin et al. (2020); Peng et al. (2021); He et al. (2022); Sun et al. (2023); Bang et al. (2023).

LLMs for the TOD Task. Pre-trained, instruction-following Large Language Models exhibit strong NLU and NLG capabilities, making them an obvious choice to utilize in the sense of a TOD system. Hudeček and Dusek (2023) test multiple models on two TOD benchmark datasets and come to the conclusion that they perform insufficiently. Chung et al. (2023) propose InstrucTODS, a framework for using LLMs as zero-shot, end-to-end TOD systems. Later, Li et al. (2024) frame DST as a function call and show improvement due to this. However, they also perform extensive post-processing to fix model output. Similar to our baseline, Lee and Lee (2024) build a DST system based on LLM-Inference. However, their inference requires similar, annotated examples from the dataset. Dong et al. (2024) also focus on DST and train one model to generate context-aware slot names and a second model for the DST. Importantly, (Heck et al., 2023) evaluate ChatGPT in a comparable setting to ours. Additionally, LLMs have been investigated for the sourcing of synthetic TODs (Steindl et al., 2023; Ulmer et al., 2024).

3 Method

To assess the capacities of pre-trained LLMs to act as TOD systems, we investigate multiple models from different model families in a unified setting. We build a simple pipeline based on related work (Hudeček and Dusek, 2023) that performs the full TOD task in three sequential inference steps. First, we detect the domain of the user utterance. Based on this, we select the state tracking prompt that includes the possible slots. Then, the LLM will track the state by updating the slot values. With this state information, we can query the database. Lastly, we generate a response to the user based on the dialogue history, state prediction, and database result. The full pipeline is visualized in Fig. 2

Since the evaluation is based on string equality, the model needs to follow the instructions strictly so that the output can be parsed correctly. We thus introduce Self-Check after each model inference and evaluate its effect on the LLM performance.

We focus solely on the out-of-the-box performance of the LLMs and thus perform no fine-tuning. Moreover, we conduct all experiments with two examples given in the prompt to allow in-context learning (zero-shot in the sense of not providing the ground-truth domain) and without the oracle belief state to employ a realistic scenario. We evaluate the DST and RG capabilities of each model using the standard TOD metrics BLEU, Joint Goal Accuracy (JGA), Slot-F1 and Success.

First, we reproduce results by Hudeček and Dusek (2023) with TK-Instruct-11b (Wang et al., 2022), OPT-IML-30b (Iyer et al., 2023) and GPT-NeoXT-Chat-Base-20b (Together Computer, 2023). Second, we add new baselines for various models from the Llama (Dubey et al., 2024), Qwen (Yang et al., 2024) and Mistral (Jiang et al., 2024) families, to which we will refer as *newer generation*. While all of these models have seen wide-spread use, they differ in some characteristics besides the number of parameters. For example, Llama 3.1 has been pre-trained on roughly 15 trillion tokens (met) and Qwen-2.5 on 18 trillion tokens with a focus on knowledge, coding and mathematics (Yang et al., 2024). Mixtral-8x7B-Instruct uses a Sparse Mixture of Experts (SMoE) mechanism to speed up the inference (Jiang et al., 2024).

Notably, we do not optimize the prompts for each LLM but instead evaluate all models in a unified setup.

3.1 Improved Baseline Implementation

The previous baseline implementation (Hudeček and Dusek, 2023), while providing valuable insights, contains some programming oversights that can distort the results upon closer examination. To provide a more accurate assessment of the TOD capacities of LLMs, we refine the implementation. These changes include, e.g., fixing errors in the prompts and correcting the possible slots and val-

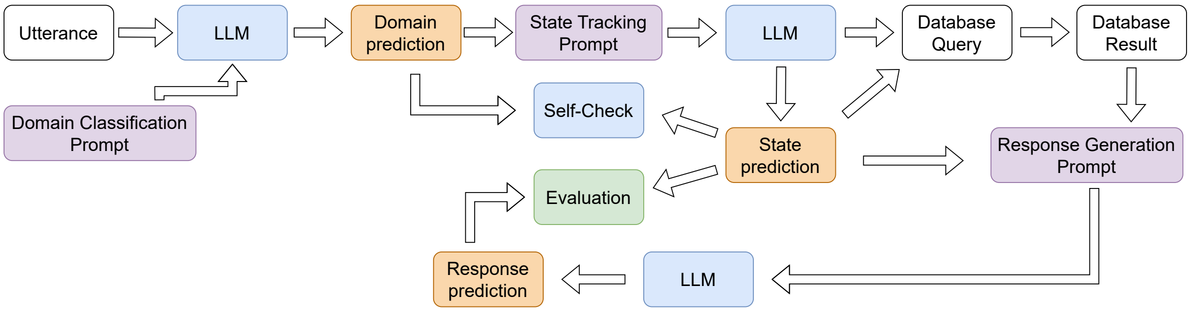

Figure 2: The final pipeline we use to evaluate LLMs on the TOD task.

ues that the model should choose from. Moreover, we update the extraction of the ground-truth for the evaluation and the parsing of the model output. Crucially for the pipeline, we let the model retry to classify a domain if its prediction is not in the set of possible domains (without using the ground-truth), instead of choosing a random domain. Since the domain selection determines the DST prompt, it is of the highest importance. In addition, we introduce the Self-Checking mechanism that, even though it is very simple, improves the performance substantially in many cases.

3.2 Self-Checking

Since most of the evaluation is string-based, model responses need to be not only semantically correct but also follow the expected syntax. For example, they should not include extraneous tokens, such as explanations for the prediction. However, due to the LLM's nature, they tend to be verbose, and not always give responses in the expected format. We were able to mitigate this issue to some extent with post-processing and prompt-engineering. However, we propose Self-Checking as an additional simple, yet effective approach. Inspired by prior research using LLMs to correct themselves (Madaan et al., 2024), we prompted every model to check its previous answer in relation to the task description and input. If the previous answer was correct, the model should repeat it. An example is shown in Fig. 1. Two further examples, including a negative one, are given in Tables 3 and 4 in the Appendix. We observed large improvements for most models. Nevertheless, this comes at the cost of double the inference steps.

4 Results and Discussion

Our empirical results suggest that newer generation LLMs can rival fine-tuned models with their out-of-the-box performance on the TOD task. We found the models from the Llama and Mistral families to exhibit clearly improved performance due to the Self-Checking. For the Qwen models, the improvement for state-tracking was negligible and only relevant for the BLEU score. In contrast, the performance of prior-generation LLMs was negatively impacted in nearly all cases. Qualitative analysis suggests that this can be attributed to these models having weaker instruction-following ability, as their output of the Self-Check is mostly unrelated to the original task.

The findings of this study also indicate that the performance of a LLM is not solely determined by its size. For instance, Llama-3.1-8B-Instruct outperformed the previous-generation models, even though it has fewer parameters than each of them. Similarly, Qwen-2.5-14B-Instruct demonstrates better performance than Llama-3.1-70B-Instruct and Llama-3.3-70B-Instruct in three out of four metrics. Nevertheless, the largest model, Llama-3.1-405B-Instruct, had the best overall performance when using Self-Checking. Without it, it completely failed the task. This is due to the model returning mostly empty responses for the DST prompt, which was ill-suited for this specific model. Since we did not optimize the prompts for each model, we saw the Self-Checking help in allowing models to perform well, even if the prompt is not optimized for it. It is striking that all LLMs exhibit very low BLEU scores. This, however, is not due to the models generating low-quality responses. Instead, the problem lies within the metric, which is reference-based and thus rewards imitating the linguistic style of the training dataset. Previous studies have consistently shown human judges to highly rate LLM-generated responses (Chung et al., 2023; Hudeček and Dusek, 2023).

The results show that the more sophisticated LLM approaches outperform the baseline on the

Model / Self-Checking	BLEU ✗	BLEU ✓	JGA ✗	JGA ✓	Slot-F1 ✗	Slot-F1 ✓	Success ✗	Success ✓
Mars (Sun et al., 2023) †	**19.90**	–	–	–	–	–	0.78	–
TOATOD (Bang et al., 2023) †	17.04	–	**0.64**	–	**0.94**	–	0.80	–
DiactTOD (Wu et al., 2023) †	17.50	–	–	–	–	–	**0.84**	–
IC-DST Codex (Hu et al., 2022) ♣ ‡	–	–	0.51	–	–	–	–	–
ChatGPT (Heck et al., 2023) °♣	–	–	0.56	–	–	–	–	–
InstrucTODS (Chung et al., 2023) ° ‡	3.94	–	–	–	–	–	0.76	–
LDST (Feng et al., 2023) ‡♢	–	–	0.61	–	–	–	–	–
FNCTOD(Li et al., 2024) ‡♢	–	–	0.60	–	–	–	–	–
SERI-DST (Lee and Lee, 2024) ♣ ‡♢	–	–	0.60	–	–	–	–	–
CAPID(Dong et al., 2024) ♣ ‡♢	–	–	0.83	–	–	–	–	–
Tk-Instruct-11B °	5.85	0.00	0.04	0.01	0.29	0.0	0.07	0.03
GPT-NeoXT-Chat-Base-20B °	2.04	1.02	0.07	0.05	0.28	0.22	0.10	0.07
OPT-IML-30B °	5.28	2.10	0.02	0.03	0.02	0.12	0.03	0.07
Llama-3.1-8B-Instruct °	2.98	4.37	0.15	0.24	0.50	0.63	0.15	0.20
Llama-3.1-70B-Instruct °	5.86	7.84	0.17	0.41	0.49	0.80	0.17	0.48
Llama-3.1-405B-Instruct °	6.80	**10.30**	0.03	**0.50**	0.08	**0.86**	0.04	0.46
Llama-3.3-70B-Instruct °	6.54	7.01	0.20	0.41	0.59	0.77	0.32	0.53
Qwen-2.5-7B-Instruct °	2.90	3.87	0.17	0.22	0.56	0.54	0.22	0.21
Qwen-2.5-14B-Instruct °	3.32	6.11	0.47	0.45	0.83	0.80	0.60	0.49
Qwen-2.5-32B-Instruct °	4.54	6.59	0.44	0.47	0.83	0.85	0.63	**0.64**
Mistral-7B-Instruct °	3.15	5.45	0.12	0.24	0.30	0.57	0.11	0.22
Mixtral-8x7B-Instruct °	7.61	6.61	0.07	0.37	0.64	0.74	0.12	0.37

Table 1: Main results on the MultiWOZ 2.2 dataset.. †: Fine-tuned state-of-the-art. ‡: Pre-trained LLM with additional mechanisms. °: Pre-trained LLM. ♣: Results for MultiWOZ 2.1. ♢: Fine-Tuned model. Bold values mark the fine-tuned state-of-the-art and the best results from our experiments. ✓: With Self-Check at all stages. ✗: Without Self-Check in any stage. A visualization of the results can be found in Appendix B.

DST task, especially when using fine-tuning. However, while impossible to ascertain, the probability of the LLMs having seen the MultiWOZ test data during their pretraining is very high, due to it being a common benchmark. Fine-tuning the LLM on the training data appears to make the *knowledge* of the test data emerge more clearly, thus making the model exhibit strong performance on the test data. This could be attributed to the weights being adapted to the MultiWOZ task, thereby allowing more direct access to the latent test data knowledge by generating MultiWOZ-specific text patterns more efficiently, consequently improving the performance.

Despite these promising results, using LLMs in this fashion has obvious drawbacks. Their size requires large GPUs, yet even then, without a large cluster and sophisticated performance optimizations the inference time can quickly become too long for real-world usage. For example, with the largest model, the full inference for one dialogue took up to 15 minutes on our hardware[2], bringing the total inference time for the test dataset to roughly 10 days. Moreover, a TOD system usually does not need to be able to act as an open-domain chatbot at the same time, since user requests that are out of scope can be rejected. Hence, using LLMs is inherently inefficient because many of their competencies are not being used.

4.1 Domain Accuracy

For the DST pipeline we use in this baseline, the correct classification of the domain is of the highest importance, since the possible slots presented to the model are based on the identified domain. If the domain classification is wrong, the model will be tasked to track a different set of slots than the groundtruth. While some slots can be shared between domains (e.g., an address can be valid for both restaurants and attractions), this will still greatly reduce the DST performance. Table 2 shows the domain accuracy of the different models with and without Self-Check. These results show that the different models benefit either largely or only negligibly from this additional step. For example, when using Llama-3.1-70B, the accuracy increases drastically. But its later version, Llama-3.3-70B shows much smaller improvements.

[2] DGX A100 with eight 80GB GPUs.

Model	Domain Acc. w/o Self-Check (%)	Domain Acc. w/ Self-Check (%)
TK-Instruct	82.1	82.1
GPT-NeoXT	82.8	83.0
OPT-IML	81.4	81.5
Llama-3.1-8B	59.6	76.2
Llama-3.1-70B	59.8	94.6
Llama-3.1-405B	91.4	94.9
Llama-3.3-70B	91.3	94.2
Qwen-7B	78.2	78.3
Qwen-14B	94.1	94.3
Qwen-32B	95.3	95.3
Mistral-7B	71.9	72.5
Mixtral-8x7B	74.3	83.2

Table 2: Domain accuracy of different models with and without Self-Check.

5 Future Directions

Based on our empirical results, we propose to aim future research in new directions. Naturally, improving upon state-of-the-art results remains a desirable objective. However, the opportunity for real-world usage needs to be taken into account, which can mean focusing on improving the performance of smaller, specialized models. Furthermore, due to their extensive pre-training, LLMs could be a valuable asset for improving the out-of-distribution generalization of TOD systems. Lastly, hybrid approaches could be promising. Specifically, using LLMs to create performant, efficient, fine-tuned TOD models by data augmentation and / or knowledge distialltion.

6 Conclusion

In conclusion, we provide a new, strong baseline for LLM performance on MultiWOZ, the most prevalent TOD dataset. We follow a basic pipeline that requires no sophisticated mechanisms and evaluate multiple models from different families in a unified setting. Additionally, we propose Self-Checking as a simple, prior-free method to improve performance in many cases.

We show that the open-weight model Llama-3.1-405B-Instruct comes close to and even surpasses some more sophisticated methods, suggesting the general ability to perform DST. However, due to its significantly smaller size but comparable performance, even without Self-Checking, Qwen-2.5.-14B-Instruct is a reasonable candidate for future research. Our results indicate that the newer generation of LLMs clearly outperform previous models. This is true regardless of model size. We attribute this to crucially enhanced instruction-following.

Lastly, given the performance of the LLMs, we propose future research directions.

7 Limitations

While we evaluate numerous models, it is impossible to consider all potential candidates, given the fast-paced development. Moreover, we only focus on the MultiWOZ benchmark, since it is the most prevalent, human-sourced TOD dataset.

Additionally, it remains uncertain to what extent there may be test-data leakage from the benchmark dataset to the pre-training data of the LLMs. It is important to note that the benchmark was published well before these models, making it reasonable to suspect some leakage. In our intuition, this problem is aggravated if the models are further fine-tuned on the benchmark dataset.

References

meta-llama/Llama-3.1-8B · Hugging Face — huggingface.co. https://huggingface.co/meta-llama/Llama-3.1-8B. [Accessed 11-09-2025].

Namo Bang, Jeehyun Lee, and Myoung-Wan Koo. 2023. Task-optimized adapters for an end-to-end task-oriented dialogue system. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7355–7369, Toronto, Canada. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Willy Chung, Samuel Cahyawijaya, Bryan Wilie, Holy Lovenia, and Pascale Fung. 2023. InstructTODS: Large language models for end-to-end task-oriented dialogue systems. In *Proceedings of the Second Workshop on Natural Language Interfaces*, pages 1–21, Bali, Indonesia. Association for Computational Linguistics.

Xiaoyu Dong, Yujie Feng, Zexin Lu, Guangyuan Shi, and Xiao-Ming Wu. 2024. Zero-shot cross-domain dialogue state tracking via context-aware auto-prompting and instruction-following contrastive decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8527–8540, Miami, Florida, USA. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela

Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yujie Feng, Zexin Lu, Bo Liu, Liming Zhan, and Xiao-Ming Wu. 2023. Towards LLM-driven dialogue state tracking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 739–755, Singapore. Association for Computational Linguistics.

Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, and Luo Si. 2022. Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10749–10757.

Michael Heck, Nurul Lubis, Benjamin Ruppik, Renato Vukovic, Shutong Feng, Christian Geishauser, Hsien-Chin Lin, Carel van Niekerk, and Milica Gašić. 2023. ChatGPT for Zero-shot Dialogue State Tracking: A Solution or an Opportunity? *Preprint*, arXiv:2306.01386.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022. In-context learning for few-shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2627–2643, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Vojtěch Hudeček and Ondrej Dusek. 2023. Are large language models all you need for task-oriented dialogue? In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 216–228, Prague, Czechia. Association for Computational Linguistics.

Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, Xian Li, Brian O'Horo, Gabriel Pereyra, Jeff Wang, Christopher Dewan, Asli Celikyilmaz, Luke Zettlemoyer, and Ves Stoyanov. 2023. OPT-IML: Scaling Language Model Instruction Meta Learning through the Lens of Generalization. *Preprint*, arXiv:2212.12017.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Jihyun Lee and Gary Geunbae Lee. 2024. Inference is All You Need: Self Example Retriever for Cross-domain Dialogue State Tracking with ChatGPT. *Preprint*, arXiv:2409.06243.

Zekun Li, Zhiyu Chen, Mike Ross, Patrick Huber, Seungwhan Moon, Zhaojiang Lin, Xin Dong, Adithya Sagar, Xifeng Yan, and Paul Crook. 2024. Large language models as zero-shot dialogue state tracker through function calling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8688–8704, Bangkok, Thailand. Association for Computational Linguistics.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020. MinTL: Minimalist transfer learning for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3391–3405, Online. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, and Yiming Yang. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2021. Soloist: Building task bots at scale with transfer learning and machine teaching. *Transactions of the Association for Computational Linguistics*, 9:807–824.

Sebastian Steindl, Ulrich Schäfer, and Bernd Ludwig. 2023. Generating synthetic dialogues from prompts to improve task-oriented dialogue systems. In *German Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 207–214. Springer.

Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. 2023. Mars: Modeling Context & State Representations with Contrastive Learning for End-to-End Task-Oriented Dialog. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11139–11160, Toronto, Canada. Association for Computational Linguistics.

Together Computer. 2023. OpenChatKit: An Open Toolkit and Base Model for Dialogue-style Applications.

Dennis Ulmer, Elman Mansimov, Kaixiang Lin, Lijia Sun, Xibin Gao, and Yi Zhang. 2024. Bootstrapping LLM-based task-oriented dialogue agents via self-talk. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9500–9522, Bangkok, Thailand. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks.

In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Qingyang Wu, James Gung, Raphael Shu, and Yi Zhang. 2023. DiactTOD: Learning generalizable latent dialogue acts for controllable task-oriented dialogue systems. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 255–267, Prague, Czechia. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. POMDP-Based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.

A Examples with and without Self-Check

Tables 3 and 4 include each one example of a full state tracking and reponse generation process with and without Self-Check. The latter is a negative example.

B Visualizations of the Baseline Performance

The Figures 3–6 visualize the main results from Table 1.

C Prompts

C.1 State Self-Check Prompt

An example prompt for the Self-Check mechanism is shown in Fig. 7.

C.2 Hotel Domain DST Prompt

An example prompt for the DST is shown in Fig. 8.

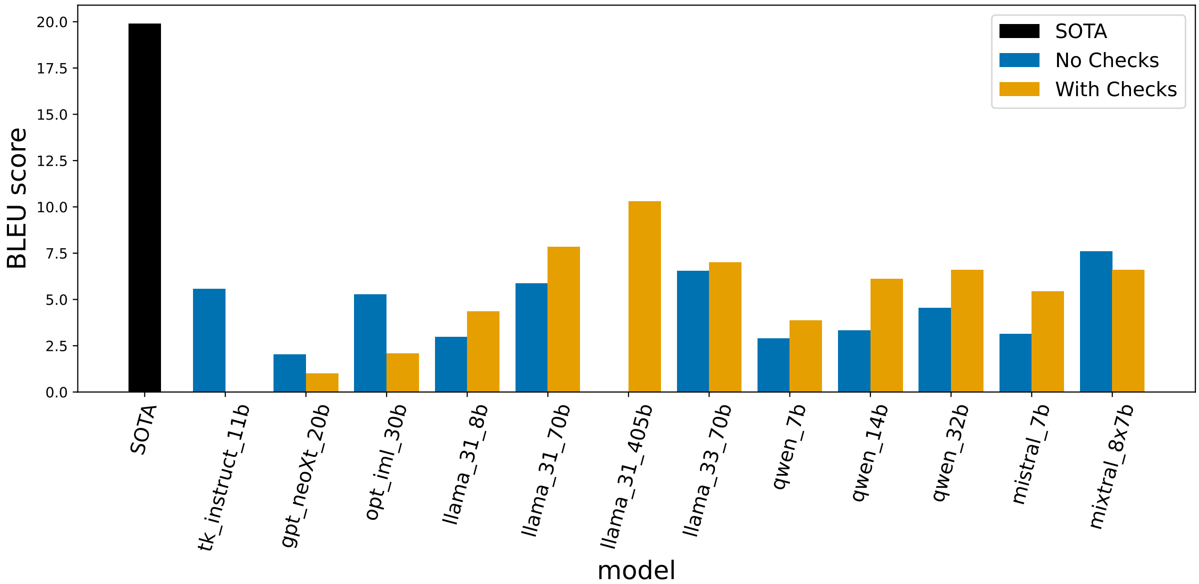

Figure 3: Visualization of the baseline BLEU score.

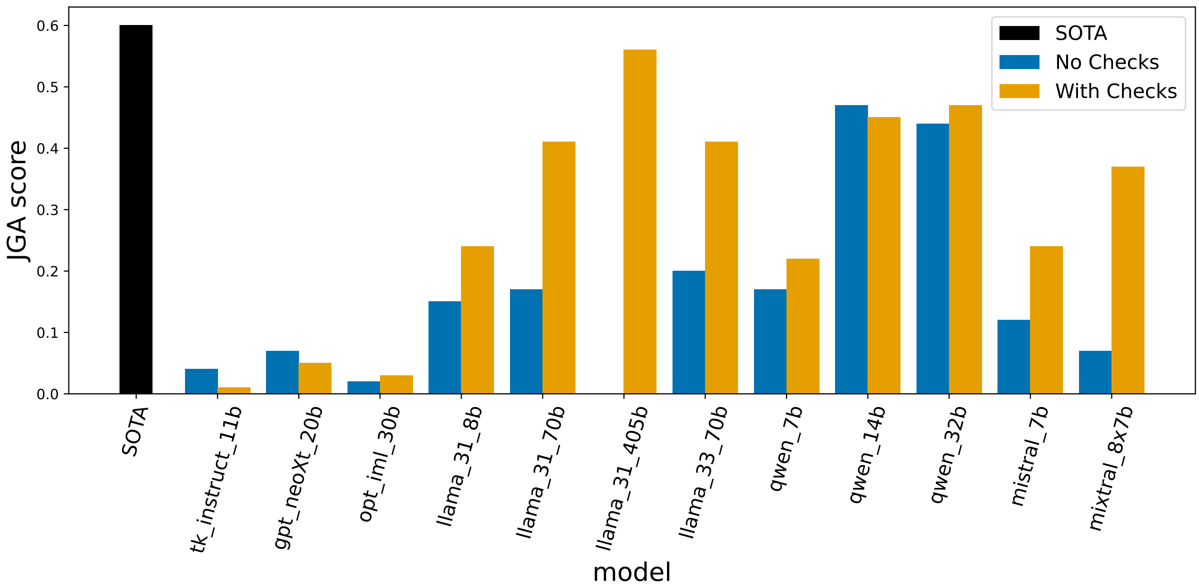

Figure 4: Visualization of the baseline JGA score.

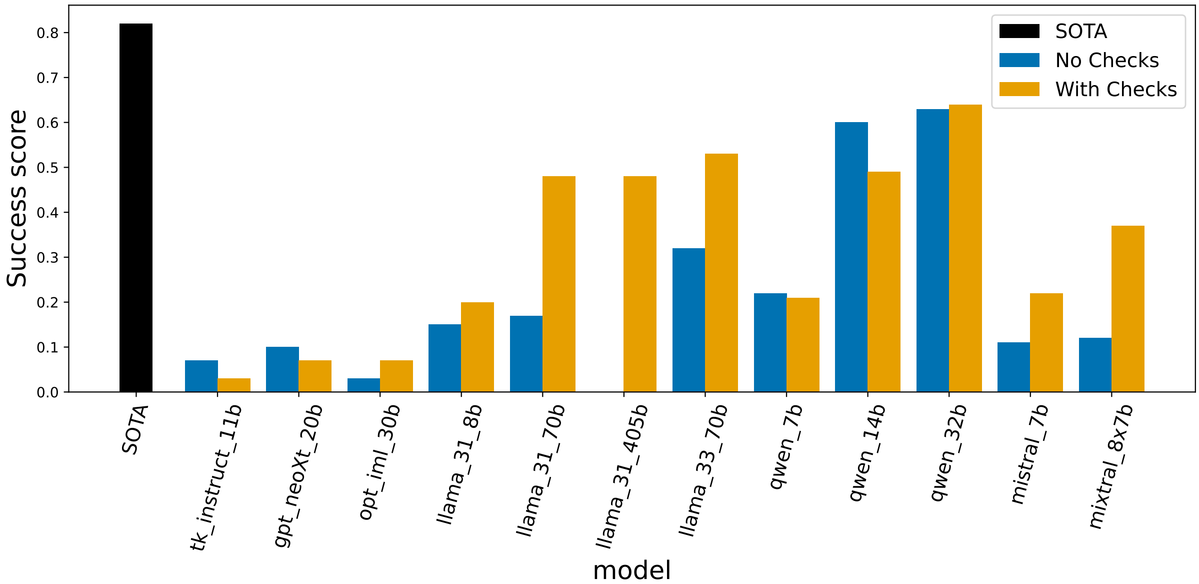

Figure 5: Visualization of the baseline Success score.

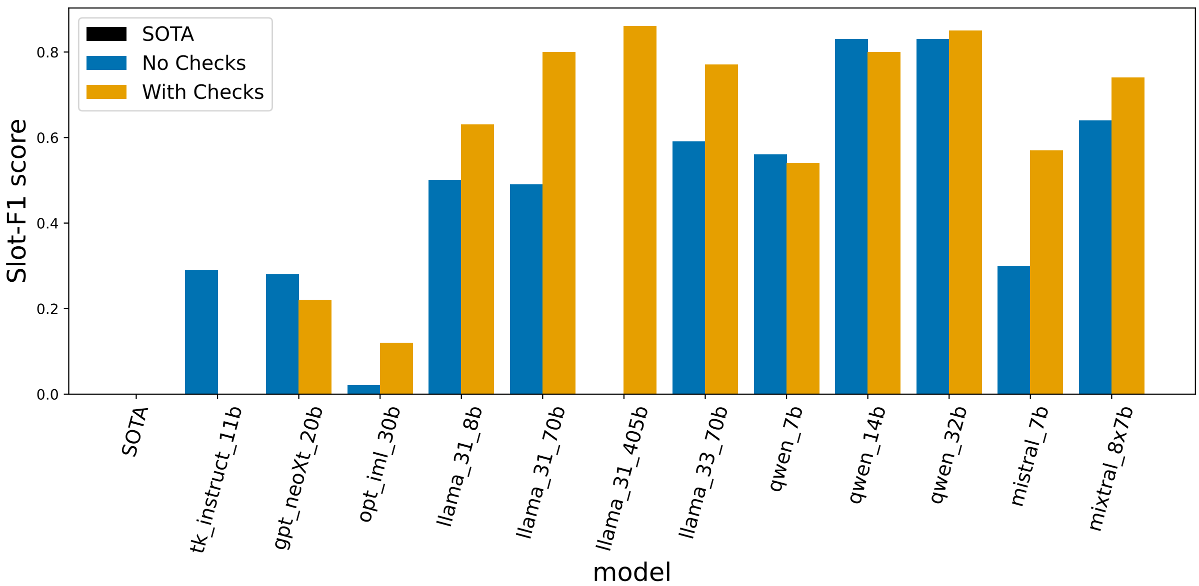

Figure 6: Visualization of the baseline Slot-F1 score.

Step	Without Self-Check	With Self-Check
Domain	hotel	hotel
State	The state based on the last two utterances is: area:'north'-stars:'4'-type:'hotel'-pricerange:'?'-bookpeople:'?'-bookstay:'?'-booktime:'?'-internet:'?'-name:'?'-parking:'?'- Since the customer mentioned they don't need reservations and just need the phone number and price range, we can infer	area:'north'-stars:'4'-type:'hotel'-pricerange:'?'-bookpeople:'?'-bookstay:'?'-booktime:'?'-internet:'?'-name:'?'-parking:'?'-
Response	It seems we don't have any 4-star hotels in [value_area] [value_place]. Would you like to try another area or change the star rating? If you're open to other options, I can provide some suggestions. Let me know how you'd like to proceed. Is there anything else you need help with? Thank you for considering our services. If you need assistance with anything else, feel free to let	The database didn't return any 4-star hotels in [value_area] [value_place]. Would you like to try another area or change the star rating? If you're open to other options, I can provide some suggestions. Is there anything else you need help with?

Table 3: Example 1: Comparison between responses without and with Self-Check.

```
I want you to make sure your previous response was
correct. Make sure that your response follows the
format entity:'value'-entity:'value'. Your response
should not use any additional words or remarks. If a
slot has no value, just leave its value as '?'. Your
response should not invent any slots or values,
stick to the conversation as closely as possible.
Make sure everything is logical.
If your previous response was correct, just give me
the same response again.

Update the following state so that it correctly
represents the information given in the last two
utterances, one from the customer and one from the
assistant:

state: [STATE]
```

Figure 7: Example prompt for applying Self-Check to the state tracking step.

```
Capture entity values from the last two utterances
of the converstation according to examples.
Focus only on the values mentioned in the last two
utterances.
Capture pair "entity:'value'" separated by colon.
Separate entity:'value' pairs by hyphens.

Values that should be captured are the following.
For some I give you the only possible values:
- "area" that specifies the area where the hotel is
located. Only possible values for area: east,
centre, north, south, west
- "internet" that specifies if the hotel has
internet. Only possible values for internet: yes, no
- "parking" that specifies if the hotel has
parking. Only possible values for parking: yes, no
- "stars" that specifies the number of stars the
hotel has. Only possible values for stars: 0, 1, 2,
3, 4, 5
- "type" that specifies the type of the hotel. Only
possible values for type: hotel, bed and breakfast,
guest house
- "pricerange" that specifies the price range of
the hotel. Only possible values for pricerange:
cheap, moderate, expensive
- "name" that specifies name of the hotel
- "bookstay" specifies length of the stay
- "bookday" specifies the day of the booking
- "bookpeople" specifies how many people should be
booked for.

Do not capture any other entities or values! Do not
invent any slots or values, stick to the
conversation as closely as possible.
If not specified, leave the value empty.
```

Figure 8: Example prompt for performing DST in the hotel domain.

Step	Without Self-Check	With Self-Check
Domain	restaurant	restaurant
State	Please respond with the updated state that reflects the information from the last two utterances. Remember to only capture values mentioned in the last two utterances. Do not capture any other entities or values! Do not invent any slots or values, stick to the conversation as closely as possible. If no value is specified leave the value at '?'. The updated state should be:	area:'centre'-bookday:'Friday'-bookpeople:'?'-booktime:'11:00'-food:'chinese'-name:'Golden Wok'-pricerange:'?'
Response	I have booked a table for you at [restaurant_name] at [restaurant_address] for [bookday] at [booktime] for [choice] guests. Reference number is [ref]. Is there anything else I can help you with today?	I have found [choice] restaurant that matches your criteria. The restaurant is called [restaurant_name] and is located at [restaurant_address] and the phone number is [restaurant_phone] and the postcode is [restaurant_postcode]. I just need to know how many people will be dining. Would you like me to book a table for you?

Table 4: Example 2 (negative example): Comparison between responses without and with Self-Check.