



# Factors affecting the in-context learning abilities of LLMs for dialogue state tracking

Pradyoth Hegde<sup>1,2</sup>, Santosh Kesiraju<sup>1</sup>, Ján Švec<sup>1</sup>, Šimon Sedláček<sup>1</sup>, Bolaji Yusuf<sup>1</sup>, Oldřich Plchot<sup>1</sup>, Deepak K T<sup>2</sup>, Jan Černocký<sup>1</sup>

<sup>1</sup>Speech@FIT, Brno University of Technology, Czechia

<sup>2</sup>Indian Institute of Information Technology Dharwad, India

pradyothhegde@gmail.com, {kesiraju, isvecjan, isedlacek, iyusuf, iplchot, cernocky}@fit.vut.cz, deepak@iiitdwd.ac.in

## Abstract

This study explores the application of in-context learning (ICL) to the dialogue state tracking (DST) problem and investigates the factors that influence its effectiveness. We use a sentence embedding based k-nearest neighbour method to retrieve the suitable demonstrations for ICL. The selected demonstrations, along with the test samples, are structured within a template as input to the LLM. We then conduct a systematic study to analyse the impact of factors related to demonstration selection and prompt context on DST performance. This work is conducted using the MultiWoZ2.4 dataset and focuses primarily on the OLMo-7B-instruct, Mistral-7B-Instruct-v0.3, and Llama3.2-3B-Instruct models. Our findings provide several useful insights on in-context learning abilities of LLMs for dialogue state tracking.

**Index Terms:** in-context learning, dialog state tracking

## 1. Introduction

Instruction-tuned large language models (LLMs) have demonstrated enhanced capabilities compared to traditional language models, enabling them to perform a broader range of tasks based on instructions [1, 2, 3]. A key capability of LLMs is in-context learning (ICL), which allows them to generalise to new tasks without requiring explicit fine-tuning, thus reducing the need for extensive task-specific training data and computational resources [4, 5, 6]. This adaptability makes them particularly attractive for complex applications, such as task-oriented dialogue systems, where the ability to quickly adapt to new domains and user intents is highly desirable [7].

Task-oriented dialogue (TOD) systems are conversational systems designed to achieve specific, predefined goals, such as booking a flight, ordering food, or scheduling an appointment. To achieve these goals, the dialogue manager within the system maintains a representation of the current dialogue state. The dialogue state comprises the domain of the conversation (e.g. restaurant, taxi) and the relevant information or values associated with that domain, referred to as slots (e.g. restaurant-name, taxi-destination). In multi-turn dialogues, accurately tracking the domain and the corresponding slot values is known as dialogue state tracking (DST). This is crucial for the overall performance of TOD systems, as it directly impacts the system's ability to understand user requests, provide relevant information, and ultimately achieve the desired task completion [8].

The effectiveness of ICL is heavily influenced by the selection of appropriate demonstrations [9, 10, 11, 12, 13]. Several recent works have focused on improving demonstration selection strategies for LLMs [14, 15, 16]. Rubin et al. [17] explored a retrieval-based approach, training a model to score demonstrations based on LLM performance. Venkateswaran et al. [18]

fine-tuned a language model using ICL and observed that while fine-tuning can improve performance, the quality and diversity of the demonstrations remain critical factors, noting that low diversity and semantic similarity in the selected demonstrations can negatively impact performance. Interestingly, Min et al. [19] found that the ground truth slot values in the demonstrations may not be as important as previously thought, as replacing them with random words did not significantly degrade performance, suggesting that the structure and context provided by the demonstrations may be more influential than the specific slot values themselves.

In this work, we study a nearest-neighbour-based retrieval to select relevant demonstrations (dialogue turns) for ICL-based DST. These demonstrations are then formatted using a carefully designed prompt template to effectively guide the LLM towards accurate state tracking. Specifically, we propose a modular prompt structure for DST, consisting of distinct blocks for the conversation history, the domain(s), and the corresponding slot values. This modularity allows for easy adaptation and experimentation with different prompt configurations. This study focuses exclusively on in-domain examples. Due to the poor performance of zero-shot prompting in our format, we limit our investigation to demonstration-based DST. We conduct a systematic investigation and identify the factors influencing both the demonstration retriever and the prompt structure in the final DST performance. Our investigation reveals that

- Using embeddings of only the user's utterances within the retriever for demonstration selection yields better performance than including both user and agent utterances. This suggests that user input is a more reliable indicator of the underlying dialogue state.
- Embedding models trained on multilingual data (LaBSE) or specifically for dialogue tasks (D2F) achieve comparable results in our setup, indicating that domain-specific fine-tuning of the embedding model may not be necessary for achieving good performance in our ICL-based DST system.
- We also find that tags indicating speaker roles (user, agent), the number of demonstrations, and dialogue history in the demonstrations play an important role in the final DST performance.

The remainder of this paper is organised as follows. Section 2 details our methodology and describes the key components involved. Section 3 describes our experiments and presents the results, showing the performance of our system under different configurations. Section 4 provides an in-depth analysis of our results and highlights the main findings of our investigation. Finally, Section 5 concludes by summarising our contributions and outlining potential avenues for future work in the field of ICL-based DST.

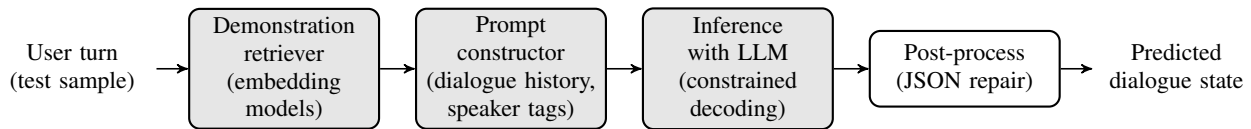


Figure 1: Block diagram of the ICL scheme used in our experiments. Factors influencing the shaded parts are studied in this paper.

## 2. Methodology

This section provides a detailed step-by-step description of our proposed system. Figure 1 provides a high-level representation of the data flow and the interactions between different modules of the system.

### 2.1. Demonstration retriever

The selection of appropriate demonstrations is crucial for effective in-context learning, as the quality and relevance of the demonstrations directly influence the LLM’s ability to generalise to new examples. While various strategies exist for selecting demonstrations, such as ensuring coverage across all domains or including multi-domain examples to enhance robustness, we observed that retrieving demonstrations from semantically similar dialogues yield better performance. Semantically similar dialogues are likely to contain relevant contextual information (domains) and slot keys that will be helpful for accurately tracking the dialogue state.

Our system employs a retriever that extracts embeddings of each dialogue turn of the user along with its history, using a pre-trained sentence embedding. In this work, we compare two embedding models. The first is LaBSE that was trained on multilingual parallel data, which efficiently captures cross-lingual features that are language agnostic [20]. The second is Dialog2Flow (D2F), which is trained with a contrastive objective to produce semantically similar embeddings for turns with similar dialogue acts [21]. To improve efficiency and reduce computational overhead, embeddings of the training set are pre-computed. During inference time, the retriever selects the  $K$  closest training samples to the test sample based on *cosine similarity*.

### 2.2. Prompt construction

The prompt serves as the primary interface of the LLM and guides its output. Our prompt template consists of three modular blocks: the conversation history, the domain(s) identified in the conversation, and the corresponding slot key-value pairs for each domain. This modular design provides flexibility, allowing the template to accommodate long conversations, multiple domains, and respective slots while maintaining a structure that is easy to parse and interpret by the LLM. The demonstrations are ordered so that the most relevant example is closest to the test sample [9]. A simple instruction is given to the model at the beginning of the prompt. Fig. 2 shows an example input prompt. The slots are represented using a JSON dictionary schema, as this allows for easy post-processing.

### 2.3. Inference

The prompt with the demonstration examples are passed as input the LLM, and we perform constrained decoding, i.e., decoding each slot value given the domain and slot key. Note that for a given domain (e.g. taxi), the possible slot keys (e.g. leaveBy, arriveAt, destination, departure) are predefined in the schema.

**Instruction:** Identify the slot value.

**User:** Can you help me get a taxi to Pizza Hut Fen Ditton? **Agent:** Sure. Where do you want to depart from? **User:** I want to depart from Sidney, Sussex College, also I need a reservation there. **Domain:** ["taxi", "restaurant"] **Slots:** {"taxi": {"arriveBy": "not mentioned", "departure": "sidney sussex college", "destination": "pizza hut fenditton", "leaveAt": "not mentioned"}, "restaurant": {"area": "centre", "day": "not mentioned", "food": "not mentioned", "name": "not mentioned", "people": "not mentioned", "pricerange": "expensive", "time": "not mentioned"}}

×  $K$

**User:** I would like a taxi from Saint John’s College to Pizza Hut Fen Ditton. **Domain:** ["taxi"] **Slots:** {"arriveBy":

Figure 2: Illustration of the prompt format used for DST with simple a instruction. Demonstrations consist of a User-Agent turns (olive), domain(s) (teal), and corresponding slot values (brown). After  $K$  such demonstrations, the test sample is presented in the same format (orange).

### 2.4. Evaluation metric

We evaluate the performance of our system using precision and recall of the predicted slot values. Precision measures the proportion of correctly predicted slots out of all predicted slots, while recall measures the proportion of correctly predicted slots out of all ground-truth slots.

## 3. Experiments and results

In this section, we present the experimental setup and results obtained from our evaluation of the proposed system. We investigated the impact of various factors related to both the demonstration retriever and the prompt structure on overall DST performance. The primary goal of these experiments is to identify the factors that affect the tracking of *the dialogue state* using in-context learning in LLM. All experiments consider the ground-truth domain when evaluating slot prediction performance.

### 3.1. Dataset and pre-processing

The MultiWOZ 2.4 dataset [22, 23] is a multi-domain, multi-turn, task-oriented dialogue dataset comprising a total of 8438 dialogues, with 6438 in training and 1000 each in development and test sets. The dialogue conversations are pre-processed to make them suitable for the task of dialogue state tracking. Specifically, each turn is constructed by accumulating the current user turn with all preceding (user + agent) turns from the same dialogue, followed by the accumulated domain and slots. This yields 56,778 training samples (turns) and 7372 test samples (turns). The training set serves as the source for demon-

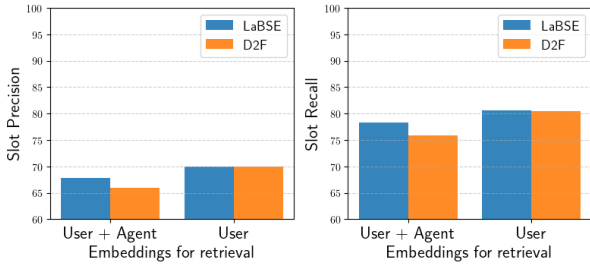


Figure 3: Slot precision and recall for LaBSE and Dialog2Flow models when User-Agent and User only sentences are considered for embeddings for retrieval.

Table 1: Precision and recall for DST system relying on retriever based on LaBSE. Embeddings were extracted using either only User or User-Agent turns; while user-agent dialogue history was used in the context.

Max. demos	Precision		Recall	
	U	UA	U	UA
10	69.9	67.8	80.6	78.3
3	67.0	67.7	81.2	79.0
1	66.0	64.4	81.3	79.1

stration retrieval in our in-context learning framework.

### 3.2. Large Language Models (LLMs)

In this work, we focus mainly on the fully open-source OLMo-7B-Instruct model [24]. Its transparency in terms of both model release and training dataset is a key consideration, aligning with principles of reproducible research. For comparison, we also include the open-weight models Mistral-7B-Instruct-v0.3 [25] and Llama3.2-3B-Instruct [26]. Unless explicitly stated, all the experiments are with OLMo-7B-Instruct.

### 3.3. Factors related to the retrieved demonstrations

#### 3.3.1. Choice of embedding extractor

Figure 3 compares the DST performance of OLMo-7B-Instruct when using demonstration retrievers based on LaBSE and D2F. Here, embeddings are extracted for User-Agent turns and User-only turns. When User-Agent turns are used, LaBSE outperforms D2F. However, the performance of both models is comparable when only user turns are used for embeddings. This suggests that focussing on user utterances provides good semantic information for effective demonstration retrieval.

#### 3.3.2. Choice of dialogue turn for embeddings

Here, we compare the performance of computing embedding using the user versus the user-agent turns. In both cases, the user-agent dialogue history is used as a context within the prompt. Table 1 presents the precision and recall scores for our system using a LaBSE based retriever. Computing embeddings using only the user’s utterances consistently outperforms using the full user-agent dialogue turns, especially when using a larger number of demonstrations. This suggests that the user’s utterances are more informative for capturing the relevant semantic information needed for effective demonstration retrieval.

Table 2: Impact of speaker tags on DST performance of OLMo-7B-Instruct, Mistral-7B-Instructv0.3 and Llama3.2-3B-Instruct models.

Model	Speaker tag	Precision	Recall
OLMo-7BI	Yes	67.8	<b>78.3</b>
	No	<b>68.1</b>	76.1
Mistral-7BI	Yes	69.5	<b>88.2</b>
	No	<b>70.2</b>	87.4
Llama3.2-3BI	Yes	67.5	<b>87.5</b>
	No	67.5	86.5

Table 3: Performance comparison of DST systems employing different numbers of demonstrations across different LLMs. (LaBSE based retriever and User-only dialogue history).

Max. demos	LLM	Precision	Recall
1	OLMo-7BI	66.5	82.3
	Mistral-7BI	<b>74.0</b>	83.1
	Llama3.2-3BI	66.9	<b>84.0</b>
3	OLMo-7BI	70.4	81.6
	Mistral-7BI	<b>73.9</b>	<b>83.1</b>
	Llama3.2-3BI	70.3	82.8
5	OLMo-7BI	71.6	80.0
	Mistral-7BI	<b>74.3</b>	<b>83.0</b>
	Llama3.2-3BI	71.5	82.6
10	OLMo-7BI	73.2	79.3
	Mistral-7BI	<b>74.8</b>	<b>83.1</b>
	Llama3.2-3BI	72.2	82.3

### 3.4. Factors related to the prompt structure

#### 3.4.1. Speaker tags

Speaker tags (“User:” and “Agent:”) are commonly used in dialogue systems to distinguish between user utterances and agent responses. This provides additional information to LLMs to classify domains and extract relevant slots. As we see in Table 2, not having a speaker tag marginally improves precision but negatively affects recall.

#### 3.4.2. Number of demonstrations

The number of demonstrations included in the context can significantly impact the performance of ICL. Since the OLMo series models can accommodate a maximum of 2048 tokens, we make sure to select only the most relevant ones well within the token limit. Table 3 shows the performance comparison of DST systems employing different numbers of demonstrations, focussing on the User dialogue history configuration. We observe that performance generally improves with more demonstrations, up to a point. However, the optimal number of demonstrations may vary depending on the specific model and the complexity of the dialogue.

#### 3.4.3. Dialogue history in the prompt

Here, we compare the impact of using user-agent turns or user-only utterances as the dialogue history within the prompt template. In both cases, the demonstrations are retrieved based on

Table 4: Impact of considering user (U) vs. user-agent turns as dialogue history in the prompt template.

Max. demos.	Precision		Recall	
	U	UA	U	UA
10	73.2	67.7	79.3	78.2
3	70.4	67.0	81.6	79.0
1	66.5	64.4	83.2	79.1

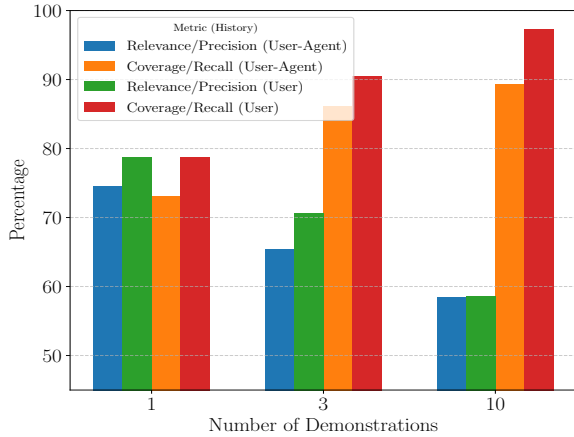


Figure 4: Demonstration relevance and coverage for User-Agent and User only dialogue history in 1, 3, 10 number of demonstrations from LaBSE based retriever.

the respective type of sentences (user-agent or user-only). Table 4 presents the results of this comparison. It can be observed that, even with a small number of demonstrations (1 or 3), using User-only utterances in the dialogue history consistently yields better precision and recall than using User-Agent turns.

## 4. Analysis

### 4.1. Relevance and coverage of demonstrations

Here, we analyse the relevance and coverage of the selected demonstrations with respect to the slots that need to be predicted in the target instance. To quantify the relevance and coverage of demonstrations, we compare the slot-keys (e.g. restaurant-name, restaurant-people, taxi-departure) present in the demonstrations to the ground truth slot-keys of the corresponding test sample. Specifically, we compute “slot-key” precision and “slot-key” recall, which can be interpreted as relevance and coverage, respectively.

Figures 4 and 5 illustrate the slot relevance and coverage for demonstrations retrieved using LaBSE and D2F embeddings, respectively. From these figures, we observe that slot relevance tends to decrease as slot coverage increases. A notable observation is that demonstrations retrieved using LaBSE exhibit better slot relevance and coverage compared to those retrieved using D2F.

The effects of the relevance and coverage of the demonstration can be observed in Table 5, which presents the precision and recall predicted for different configurations.

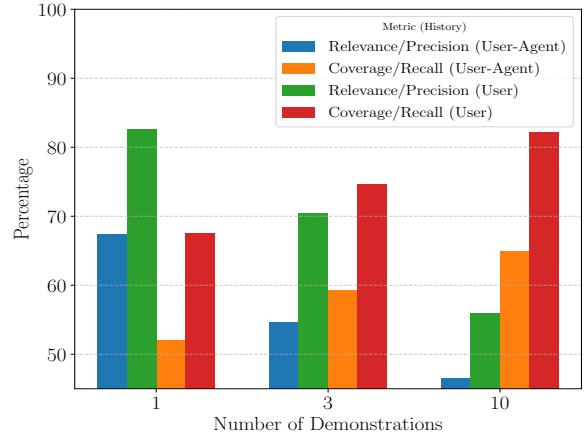


Figure 5: Demonstration relevance and coverage for User-Agent and User only dialogue history in 1, 3, 10 number of demonstrations from D2F based retriever.

Table 5: Performance comparison of max. number of demonstrations with LaBSE and D2F based retrievers.

Max. number of demonstrations	Precision		Recall	
	LaBSE	D2F	LaBSE	D2F
10	73.2	73.2	<b>79.3</b>	78.9
3	<b>70.3</b>	68.4	81.5	<b>81.66</b>
1	<b>66.5</b>	66.1	83.2	83.2

Table 6: Performance Comparison of Decoding Strategies for Dialogue State Tracking using OLMo-7BI model.

Decoding strategy	Precision	Recall
Slot value prediction (given key)	67.8	<b>78.3</b>
Slot key-value generation	<b>69.2</b>	53.3

### 4.2. Decoding strategy

We evaluate the effectiveness of the model using two distinct approaches: (1) prediction of the slot value when the slot key is provided as input and (2) generation of the entire pair of slot key value. Although slot key-value generation achieves higher precision, it exhibits a significantly lower recall due to a higher error rate with respect to the ground truth.

## 5. Conclusion

In this paper, we systematically studied various factors that influence the *in-context learning* abilities of LLMs for dialogue state tracking. We found that (a) demonstration retrievers based on a general-purpose embedding model such as LaBSE perform as good as dialogue-specific modes like D2F when using a maximum of 10 demonstrations. (b) User turns are better candidates for retrieval than user-agent turns. (c) Speaker tags have a minor but significant effect on the precision and recall across the three LLMs studied. (d) Three or more demonstrations do not yield significantly better results. (e) LaBSE and D2F based retrievers yield distinctive examples, where LaBSE has more slot relevance and coverage and hence results in better DST performance when having fewer demonstrations. We believe that our study helps us to understand the ICL abilities of LLMs for DST.

## 6. Acknowledgements

The work was supported by European Union’s Horizon Europe project No. SEP-210943216 “ELOQUENCE”, Czech Ministry of Interior project No. VK01020132 “112”, European Defence Fund project ARCHER, and by Czech Ministry of Education, Youth and Sports (MoE) through the OP JAK project “Linguistics, Artificial Intelligence and Language and Speech Technologies: from Research to Applications” (ID:CZ.02.01.01/00/23\_020/0008518). Computing on IT4I supercomputer was supported by MoE through the e-INFRA CZ (ID:90254).

## 7. References

- [1] L. Ouyang *et al.*, “Training language models to follow instructions with human feedback,” 2022, arXiv:2203.02155.
- [2] H. W. Chung *et al.*, “Scaling instruction-finetuned language models,” 2022, arXiv:2210.11416.
- [3] J. Wei *et al.*, “Finetuned Language Models are Zero-Shot Learners,” in *International Conference on Learning Representations*, 2022.
- [4] T. B. Brown *et al.*, “Language models are few-shot learners,” in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS ’20. Red Hook, NY, USA: Curran Associates Inc., 2020.
- [5] Q. Dong *et al.*, “A Survey on In-context Learning,” in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 1107–1128.
- [6] Z. Wu *et al.*, “OpenICL: An Open-Source Framework for In-context Learning,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 489–498.
- [7] Y. Hu *et al.*, “In-Context Learning for Few-Shot Dialogue State Tracking,” in *Findings of the Association for Computational Linguistics: EMNLP 2022*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 2627–2643.
- [8] V. Balaraman *et al.*, “Recent Neural Methods on Dialogue State Tracking for Task-Oriented Dialogue Systems: A Survey,” in *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Singapore and Online: Association for Computational Linguistics, Jul. 2021, pp. 239–251.
- [9] J. Liu *et al.*, “What Makes Good In-Context Examples for GPT-3?” in *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Dublin, Ireland and Online: Association for Computational Linguistics, May 2022, pp. 100–114.
- [10] X. Li and X. Qiu, “Finding Support Examples for In-Context Learning,” in *Findings of the Association for Computational Linguistics: EMNLP 2023*. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 6219–6235.
- [11] Y. Zhang *et al.*, “Active Example Selection for In-Context Learning,” in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 9134–9148.
- [12] T.-Y. Chang and R. Jia, “Data Curation Alone Can Stabilize In-context Learning,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 8123–8144.
- [13] K. Peng *et al.*, “Revisiting demonstration selection strategies in in-context learning,” in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 9090–9101.
- [14] D. Chen *et al.*, “Stabilized In-Context Learning with Pre-trained Language Models for Few Shot Dialogue State Tracking,” in *Findings of the Association for Computational Linguistics: EACL 2023*. Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 1551–1564.
- [15] X. Li *et al.*, “Unified Demonstration Retriever for In-Context Learning,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 4644–4668.
- [16] D. Shu and M. Du, “Comparative Analysis of Demonstration Selection Algorithms for LLM In-Context Learning,” 2024, arXiv:2410.23099.
- [17] O. Rubin *et al.*, “Learning to retrieve prompts for in-context learning,” in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 2655–2671.
- [18] P. Venkateswaran *et al.*, “District: Dialogue state tracking with retriever driven in-context tuning,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 5101–5112.
- [19] S. Min *et al.*, “Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?” in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Abu Dhabi, UAE: Association for Computational Linguistics, December 2022, pp. 11 048–11 064.
- [20] F. Feng *et al.*, “Language-agnostic BERT sentence embedding,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 878–891.
- [21] S. Burdisso *et al.*, “Dialog2Flow: Pre-training soft-contrastive action-driven sentence embeddings for automatic dialog flow extraction,” in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 5421–5440.
- [22] P. Budzianowski *et al.*, “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*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 5016–5026.
- [23] F. Ye *et al.*, “MultiWOZ 2.4: A Multi-Domain Task-Oriented Dialogue Dataset with Essential Annotation Corrections to Improve State Tracking Evaluation,” in *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Edinburgh, UK: Association for Computational Linguistics, Sep. 2022, pp. 351–360.
- [24] D. Groeneveld *et al.*, “OLMo: Accelerating the science of language models,” in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 15 789–15 809.
- [25] A. Q. Jiang *et al.*, “Mistral 7B,” 2023, arXiv:2310.06825.
- [26] A. Grattafiori *et al.*, “The Llama 3 Herd of Models,” 2024, arXiv:2407.21783.