Exploring LLM-based Data Synthesis Strategies for Conversational Semantic Frame Analysis

Shiho Matta[†], Yin Jou Huang[†], Fei Cheng[†], Hirokazu Kiyomaru[*], Yugo Murawaki[†]
[†]Kyoto University
[*]Research and Development Center for Large Language Models, National Institute of Informatics
{matta, huang, feicheng, murawaki}@nlp.ist.i.kyoto-u.ac.jp,
kiyomaru@nii.ac.jp

Abstract

Creating training data for supervised learning models has traditionally been time-consuming and costly. However, recent advancements in large language models (LLMs) have enabled many studies to leverage these models for synthesizing training data. In this paper, we explore data synthesis strategies for conversational semantic frame analysis, a complex task involving the extraction of entities and relations from dialogue contexts. We propose two novel methods tailored for this purpose: Forward Synthesis and Reverse Synthesis. Our results demonstrate that Forward Synthesis can achieve performance levels comparable to its creator LLM. Additionally, we provide an in-depth analysis of Reverse Synthesis, highlighting the challenges in this approach.

1 Introduction

Collecting training data for supervised learning models (SLMs) can be costly. As a result, many studies have proposed leveraging large language models (LLMs) to synthesize training data to address this issue. Recent studies have explored generating training data for various SLMs and tasks using techniques such as few-shot learning [1] and self-instruct [2], aiming for high-quality and diverse synthetic data. For tasks such as text classification and question answering, studies have demonstrated that synthetic training data performs comparable to human-annotated data with significantly reduced costs [3, 4].

In this paper, we explore data synthesis strategies for the task of conversational semantic frame analysis (SFA). This task aims to capture knowledge transfer between two speakers in a dialogue by extracting semantic frames that represent events. Each semantic frame consists of a **trigger**, which denotes the core action of the event, and **arguments**, which provide details about the event and are always linked to the event-evoking trigger. An example of dialogue and SFA annotation is presented in Figure 1. Compared to the target tasks in previous data synthesis efforts, SFA is significantly more labeling-intensive and requires the analysis of complex relational structures among entities within the dialogue. As a result, few prior works address tasks as complex as SFA, necessitating independent exploration and the development of novel approaches in this study.

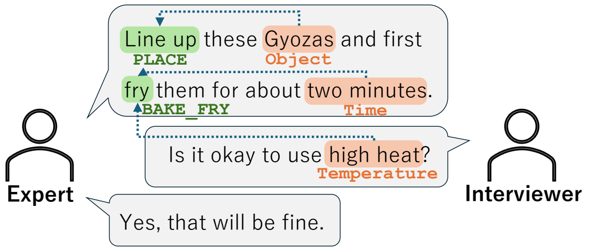

Line up these Gyozas and first
PLACE
Object
fry them for about two minutes.
BAKE_FRY
Time
Is it okay to use high heat?
Temperature
Expert
Interviewer
Yes, that will be fine.

Figure 1: An example dialogue with SFA annotation, translated from Japanese. Triggers are marked in green, and arguments in orange. Relations are illustrated with arrows.

We explored two data synthesis strategies for SFA: **Forward Synthesis** and **Reverse Synthesis**. In Forward Synthesis, we first synthesize pseudo-dialogues and then apply pseudo-labels to them. In Reverse Synthesis, we reverse the process: we first synthesize pseudo-labels, and then generate pseudo-dialogues that contain those labels. The latter approach is inspired by Josifoski et al. [5], who showed that LLMs are more effective at generating context when provided with the label first. This is particularly relevant for information extraction tasks like relation extraction, which share similarities with SFA.

Our experimental results show that Forward Synthesis generates data that achieves performance comparable to

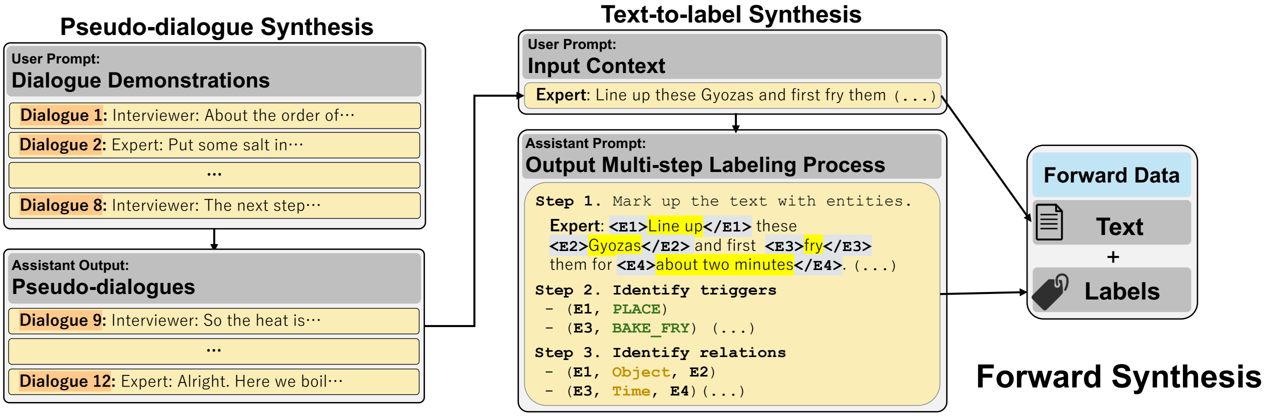

Pseudo-dialogue Synthesis
User Prompt:
Dialogue Demonstrations
Dialogue 1: Interviewer: About the order of···
Dialogue 2: Expert: Put some salt in···
···
Dialogue 8: Interviewer: The next step···
Assistant Output:
Pseudo-dialogues
Dialogue 9: Interviewer: So the heat is···
···
Dialogue 12: Expert: Alright. Here we boil···
Text-to-label Synthesis
User Prompt:
Input Context
Expert: Line up these Gyozas and first fry them (...)
Assistant Prompt:
Output Multi-step Labeling Process
Step 1. Mark up the text with entities.
Expert: <E1>Line up</E1> these <E2>Gyozas</E2> and first <E3>fry</E3> them for <E4>about two minutes</E4>. (...)
Step 2. Identify triggers
– (E1, PLACE)
– (E3, BAKE_FRY) (...)
Step 3. Identify relations
– (E1, Object, E2)
– (E3, Time, E4)(...)
Forward Data
Text
+
Labels
Forward Synthesis

Figure 2: The overview of Forward Synthesis.

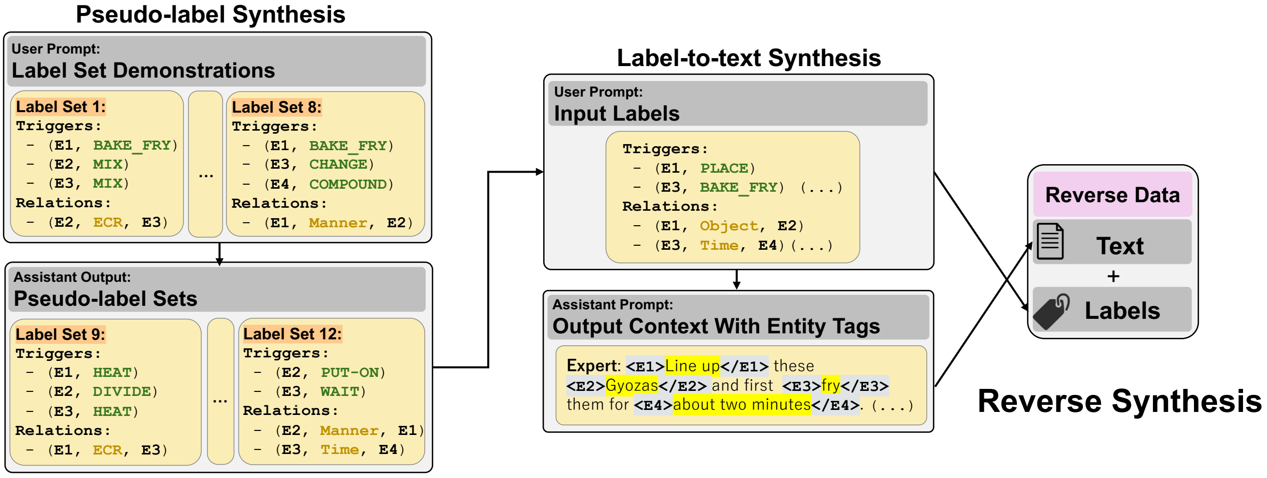


Figure 3: The overview of Reverse Synthesis.

its creator, GPT-4. In contrast, Reverse Synthesis faces a diversity issue in our setting, which limits its effectiveness. We investigate the root cause of this limitation in this paper.

2 SFA Data Synthesis Using LLMs

SFA is a complex task that requires extracting entities and relations from a given context. Previous attempts to create training data for downstream tasks only focused on sentence-level labels or were limited to identifying a single class of entity at a time. [4, 6].

To create training data for SFA, we design two data synthesis methods: **Forward Synthesis** and **Reverse Synthesis** that enable an LLM to handle this task in a text generation manner, efficiently capturing all the entities, spans, and relations within the context in a single run. To be noted, it is essential to consider entity spans within the context to capture multiple and recurring entities across utterances due to the colloquial nature of the dialogue.

2.1 Forward Synthesis

In Forward Synthesis (Figure 2), we first generate pseudo-dialogues, and then apply pseudo-labels to them.

2.1.1 Pseudo-dialogue Synthesis

The first step of Forward Synthesis is to generate the dialogues, which are the text part of the data. Adopting the self-instruct method [2], we utilize an LLM to bootstrap generating pseudo-dialogues based on a few reserved seed dialogues from human-generated data. We do this by randomly sampling human dialogues and previously generated pseudo-dialogues as few-shot examples for the LLM. Pseudo-dialogues are included in the few-shot examples to further encourage diversity. The model is instructed to mimic the style of the dialogue examples while generating new and diverse topics.

2.1.2 Text-to-label Synthesis

After generating the pseudo-dialogues, we apply pseudo-labels to them via a novel three-step tagging and labeling prompting scheme that converts SFA into a text generation task, which we refer to as **the multi-step labeling process**. The steps are as follows, given an input context:

1. **Entity Tagging:** Insert entity tags in numerical order,

such as <En> and </En> (n is an integer starting from 1), to mark the start and end of entities within the context. The LLM should copy the context perfectly while adding the entity tags where appropriate.

2. **Trigger Detection:** Identify the triggers among the entities tagged in Step 1.
3. **Relation Detection:** Determine the relationships between the entities tagged in Step 1.

Specifically, Step 1 is inspired by Wang et al. [7], who use tag pairs to indicate the span of an entity.

We provide few-shot labeling demonstrations from reserved human-annotated data to the LLM for each pseudo-dialogue created. Definitions and common examples for each label type are provided in the model's instructions.

2.2 Reverse Synthesis

In Reverse Synthesis (Figure 3), we adopt a label-first, text-next strategy.

2.2.1 Pseudo-label Synthesis

We first generate pseudo-label sets, also adopting the self-instruct method. Each label set contains only trigger and relation labels, corresponding to the style of steps two and three of the multi-step labeling process outlined in Forward Synthesis (Section 2.1.2). It is important to note that the pseudo-labels here include only the label type, such as *BAKE_FRY*, and do not specify the entities, such as "炒める (to fry)" corresponding to *BAKE_FRY*. In addition to the task descriptions for this step, we provide the LLM with a complete list of available entity types in the instructions.

2.2.2 Label-to-text Synthesis

To generate dialogue contexts containing these labels, we prepare few-shot demonstrations for each pseudo-label set. The input in the user prompt is a pseudo-label set, while the output in the assistant prompt is structured in the style of the first step in the multi-step labeling process in Section 2.1.2. The LLM is expected to generate the dialogue context while inserting entity tag pairs to denote the entities, ensuring alignment with the labels provided in the input. We provided the LLM with definitions and common examples for each label type.

3 Experimental Settings

To outline our experiments, we first synthesized **Forward** and **Reverse Data** and then trained the supervised learning model (SLM) for SFA using these data. Then, we evaluated the performance using a classification metric, where a higher F1 score indicates better data quality.

In the experiments, we sampled few-shot examples and used the test data from the EIDC dataset [8, 9], which includes transcriptions of Japanese interview dialogues paired with their corresponding semantic frame annotations.[1] The semantic frames in the cooking domain are designed to capture cooking-related events. A complete list of entity types for this domain is shown in Appendix A.2.

We created 4,300 instances each for the Forward and Reverse Data. Detailed data statistics are available in Appendix A.2. Hyperparameters such as the number of few-shots and temperature in each process are listed in Table 3 in the Appendix.

3.1 Settings for Forward Synthesis

For pseudo-dialogue synthesis, we used GPT-4-0613. The seed human dialogue examples were sampled from a reserved pool of 51.[2] We initially sampled 8 human dialogues for few-shot learning when generating the first 100 pseudo-dialogues. Then, we adjusted the sampling strategy to include 6 human dialogues and 2 pseudo-dialogues.

In text-to-label synthesis, we utilized GPT-4-0613. Few-shot examples were retrieved by calculating the ROUGE-L similarity between the context of the labeling target and the candidate examples, selecting the highest-scoring ones.

3.2 Settings for Reverse Synthesis

To synthesize pseudo-label sets, we used GPT-4o-2024-11-20[3] in a manner that is similar to the pseudo-dialogue synthesis process (Table 3).

We utilized GPT-4-0613 in label-to-text synthesis. The few-shot examples were selected based on their similarity

[1] In the following experiments, we used a heuristic method to segment a dialogue into smaller **sessions**, each consisting of up to 10 turns of utterances. All data in this paper were created in this manner.

[2] A fixed set of 51 training data samples from the EIDC dataset is designated as the exclusive pool of few-shot candidates for all LLM-related data synthesis processes discussed in this paper.

[3] We empirically observed that the pseudo-label sets generated by GPT-4-0613 lacked diversity in various classes. Switching to GPT-4o significantly improved this issue.

Table 1: Trigger and argument detection performance in weighted-F1 score.

Training data	Trigger F1	Argument F1
3-shot GPT-4-0613	0.526	0.307
51 Few-shot Data	0.398	0.177
Forward Data	**0.538**	**0.296**
Reverse Data	0.389	0.186

in label occurrences to the target pseudo-label set. Refer to Appendix A.1 for a detailed demonstration.

3.3 SFA Model and Evaluation Metric

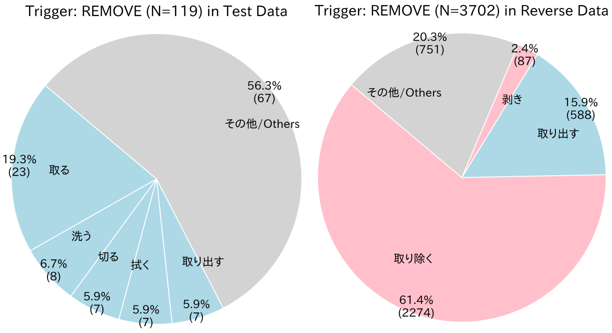
We adopted JaMIE [10] as the SLM for SFA. Its architecture consists of a transformer encoder and multiple decoding heads, allowing it to perform sequence labeling and relation extraction tasks. We employed the Japanese DeBERTa-V2-base[4] as the pre-trained encoder for JaMIE and trained the relation decoding heads from scratch.

We evaluated the performance of **Trigger Detection** and **Argument Detection** using a classification metric that accounts for both type and span accuracy of entities. Correct predictions require both the entity's type and span to be accurate. Argument predictions are marked false if their associated trigger is incorrect. The overall performance is measured using a weighted F1 score, aggregated from the F1 scores of each class.

4 Results and Analysis

The performance of JaMIE trained on Forward and Reverse Data compared to their creator: GPT-4, is presented in Table 1. Forward Data achieved performance comparable to few-shot GPT-4, whereas Reverse Data significantly underperforms, reaching comparable levels only when trained on the 51 few-shot examples.

To understand why Reverse Data performed worse than Forward Data, we conducted a case study on the *REMOVE* trigger. In this case, Forward Data achieved an F1 score of 0.681, while Reverse Data only reached 0.403 (-0.278). Analyzing **mentions** (words/tokens) for *REMOVE* triggers in the test and Reverse Data (Figure 4), we observed a dominant mention, “取り除く (to remove),” in Reverse Data (>50%, Figure 4b), which appears in <5% of the test data. Additionally, top mentions except for “取り出す (to take out)” in the test data (Figure 4a) were underrepresented in Reverse Data. Forward Data, on the other hand, has a better mention diversity (Figure 5 in Appendix), explaining the performance gap.



(a) *REMOVE* in test data. (b) *REMOVE* in Reverse Data.

Figure 4: Mentions of trigger *REMOVE* in test and Reverse data. The pink color in (b) means the same mention is less than 5% in the test data.

The issue of limited and biased mention diversity in Reverse Data stemmed from the design of Reverse Synthesis. During label-to-text synthesis, the LLM is expected to generate the context by referencing both the instruction and the few-shot examples. However, the LLM focused excessively on a single mention of *REMOVE* in the instruction:

- REMOVE: 何かから何かを取り除く。（例：洗う、剥く、**取り除く**、剥ぐ、取る）

Although other mentions for *REMOVE* (e.g., “取る”, “洗う”, and “流す”) were included in the few-shot examples, the LLM showed little inclination to generate these alternatives. In contrast, during Forward Synthesis, the LLM generated the context in the pseudo-dialogue synthesis process without being constrained by entity mention demonstrations in the instruction, as none were provided.

5 Conclusion

In this paper, we explored the **Forward** and **Reverse** data synthesis strategies for semantic frame analysis (SFA). Experimental results demonstrate that Forward Synthesis can generate training data that achieves performance on par with its creator, GPT-4. In contrast, Reverse Synthesis in our setting suffers from a label diversity issue, which limits its effectiveness. We hope our in-depth analysis will contribute to advancing data synthesis methods, enabling the creation of high-quality and diverse LLM-generated data for tasks such as SFA.

[4] https://huggingface.co/ku-nlp/deberta-v2-base-japanese

Acknowledgement

This paper is based on results obtained from a project, JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[2] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 13484–13508, Toronto, Canada, July 2023. Association for Computational Linguistics.

[3] Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. AnnoLLM: Making large language models to be better crowdsourced annotators. In Yi Yang, Aida Davani, Avi Sil, and Anoop Kumar, editors, **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)**, pp. 165–190, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[4] Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. Want to reduce labeling cost? GPT-3 can help. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, **Findings of the Association for Computational Linguistics: EMNLP 2021**, pp. 4195–4205, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[5] Martin Josifoski, Marija Sakota, Maxime Peyrard, and Robert West. Exploiting asymmetry for synthetic training data generation: SynthIE and the case of information extraction. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 1555–1574, Singapore, December 2023. Association for Computational Linguistics.

[6] Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. Is GPT-3 a good data annotator? In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 11173–11195, Toronto, Canada, July 2023. Association for Computational Linguistics.

[7] Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. GPT-NER: Named entity recognition via large language models. **arXiv preprint arXiv:2304.10428**, 2023.

[8] Taro Okahisa, Ribeka Tanaka, Takashi Kodama, Yin Jou Huang, and Sadao Kurohashi. Constructing a culinary interview dialogue corpus with video conferencing tool. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis, editors, **Proceedings of the Thirteenth Language Resources and Evaluation Conference**, pp. 3131–3139, Marseille, France, June 2022. European Language Resources Association.

[9] Taishi Chika, Taro Okahisa, Takashi Kodama, Yin Jou Huang, Yugo Murawaki, and Sadao Kurohashi. Domain transferable semantic frames for expert interview dialogues, 05 2024.

[10] Fei Cheng, Shuntaro Yada, Ribeka Tanaka, Eiji Aramaki, and Sadao Kurohashi. JaMIE: A pipeline Japanese medical information extraction system with novel relation annotation. In **Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC 2022)**, 2022.

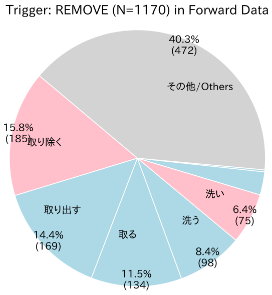


Figure 5: Mentions of trigger *REMOVE* in Forward Data.

Table 2: Statistics of the data.

	Few-shots	Test Data	Forward	Reverse
Data Size	51	379	4300	4300
Length	108 ± 41	103 ± 33	107 ± 19	103 ± 29
Turns	5.61 ± 2.04	5.87 ± 1.90	4.66 ± 1.13	6.57 ± 2.03
Triggers	4.22 ± 2.78	4.08 ± 2.51	6.24 ± 2.39	5.52 ± 0.71
Relations	6.51 ± 5.44	5.51 ± 4.09	12.9 ± 4.80	7.92 ± 1.11

Table 3: Hyperparameters in each process.

Process	Model	Temperature	Presence Penalty	#Few-shot (H: Human, P: Pseudo)
Pseudo-dialogues	GPT-4-0613	0.7	2	8 H (for first 100) → 6 H + 2 P
Text-to-label	GPT-4-0613	0	0	3H
Pseudo-labels	GPT-4o-2024-11-20	0.7	0	8 H (for first 100) → 6 H + 2 P
Label-to-text	GPT-4-0613	0	0	4H

A Experimental Settings: Details

A.1 Few-shot retrieval in Reverse Synthesis: label-to-text.

To measure similarity, we count the occurrences of each label type and calculate the cosine similarity. For instance, if the target pseudo-label set is represented as (3,0,2), corresponding to 3 *BAKE_FRY*, 0 *Object*, and 2 *Instrument*, then the most similar few-shot example would be one with a vector like (2,0,3) — that is, 2 *BAKE_FRY*, 0 *Object*, and 3 *Instrument*. This is because it has a higher cosine similarity to the target vector compared to an example like (0,1,0).[5)]

A.2 Data Statistics

We provide data statistics for the data used or synthesized in this paper (Table 2). The length, number of turns, trigger counts, and relation counts are averaged across sessions, with the standard deviation indicated by ±. The frequencies for each label per dialogue session are presented in Figure 6.

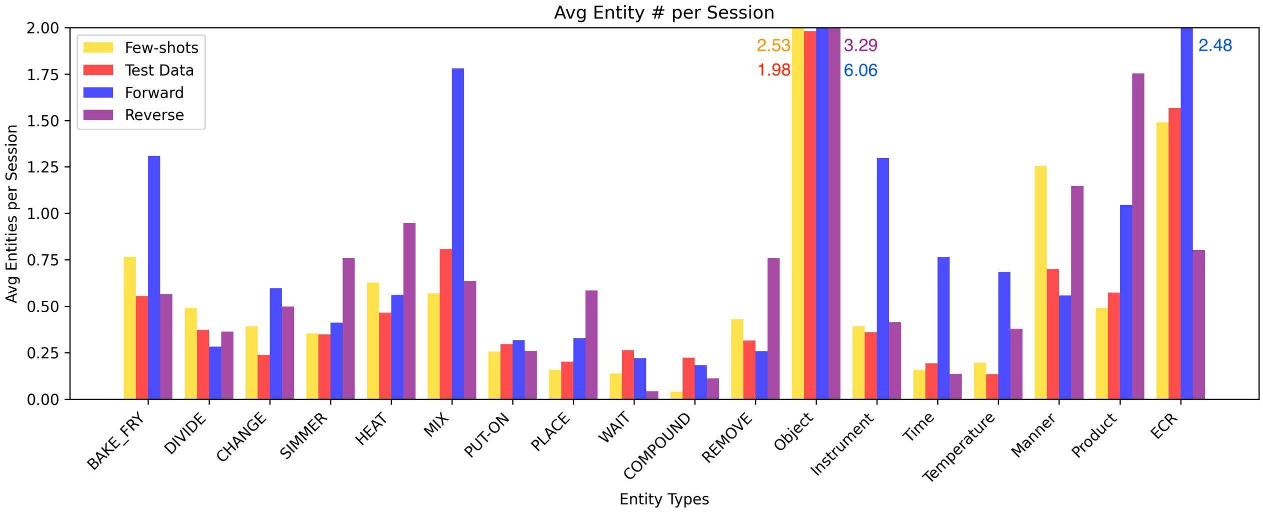


Figure 6: The distributions of label frequencies.

5) This is a simplified demonstration. In practice, there are 18 types of triggers and relations, meaning the vectors have a length of 18.