Optimizing Cost-Efficiency with LLM-Generated Training Data for Conversational Semantic Frame Analysis

Shiho Matta[†], **Yin Jou Huang**[†], **Fei Cheng**[†], **Hirokazu Kiyomaru**[*], **Yugo Murawaki**[†]
[†]Kyoto University
[*]NII LLMC
{matta, huang, feicheng, murawaki}@nlp.ist.i.kyoto-u.ac.jp,
kiyomaru@nii.ac.jp

Abstract

Recent studies have shown that few-shot learning enables large language models (LLMs) to generate training data for supervised models at a low cost. However, for complex tasks, the quality of LLM-generated data often falls short compared to human-labeled data. This presents a critical challenge: how should one balance the trade-off between the higher quality but more expensive human-annotated data and the lower quality yet significantly cheaper LLM-generated data? In this paper, we tackle this question for a demanding task: conversational semantic frame analysis (SFA). To address this, we propose a novel method for synthesizing training data tailored to this complex task. Through experiments conducted across a wide range of budget levels, we find that smaller budgets favor a higher reliance on LLM-generated data to achieve optimal cost-efficiency.

1 Introduction

It is costly to construct training data with human annotation for supervised learning models (SLMs). In recent years, large language models (LLMs) like GPT-4 have demonstrated remarkable abilities in generating coherent text, understanding context, and following complex specifications to accomplish tasks (Brown et al., 2020; OpenAI, 2024). Therefore, there have been many attempts to leverage existing LLMs as data synthesizers to generate training data for SLMs, aiming to reduce data costs. Studies have indicated that using LLM-generated data can cut costs significantly while achieving a comparable performance against human-annotated data for certain tasks (Wang et al., 2021; Ding et al., 2023).

In this paper, we explore the feasibility of synthesizing training data for conversational semantic frame analysis (SFA). SFA captures knowledge exchanged between speakers by extracting semantic frames, which consist of a **trigger** (the main action) and its **arguments** (details of the event). For example, in Figure 1, the triggers "line up" (*PLACE*) and "fry" (*BAKE_FRY*) are annotated, with corresponding arguments like *Object*, *Time*, and *Temperature* linked to them. An important characteristic of these dialogues is the frequent repetition and confirmation of technical details. For example, in Figure 1, the interviewer's question introduces a new argument to an existing frame. Refer to Figure 12 in Appendix for a longer and more complex annotation example.

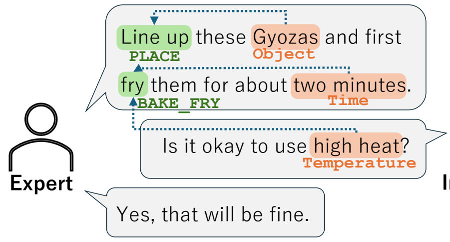

Figure 1: A dialogue piece with semantic frame annotation. Green indicates a trigger, and orange indicates an argument. The argument-trigger relation is illustrated with arrows. This is a simplified demonstration translated from Japanese.

We expect LLM-generated data for SFA to be of lower quality than human-annotated data, as SFA is significantly more complex than the tasks typically addressed in previous LLM-based data synthesis studies (Wang et al., 2021; Ding et al., 2023; He et al., 2024; Josifoski et al., 2023). These studies have primarily focused on simpler tasks such as sentence-level labeling, extracting relation triplets, or tasks with fewer recurring entities and relations. Furthermore, Ma et al. (2023) demonstrated that few-shot LLMs generally underperform in many information extraction tasks, such as named entity recognition, compared to supervised baselines. Given these findings, it is reasonable to expect that LLM-generated data for SFA will also be of lower quality than human-annotated data.

Given that LLM-generated data for SFA may

Proceedings of LaTeCH-CLfL 2025, pages 238–251
May 4, 2025 ©2025 Association for Computational Linguistics

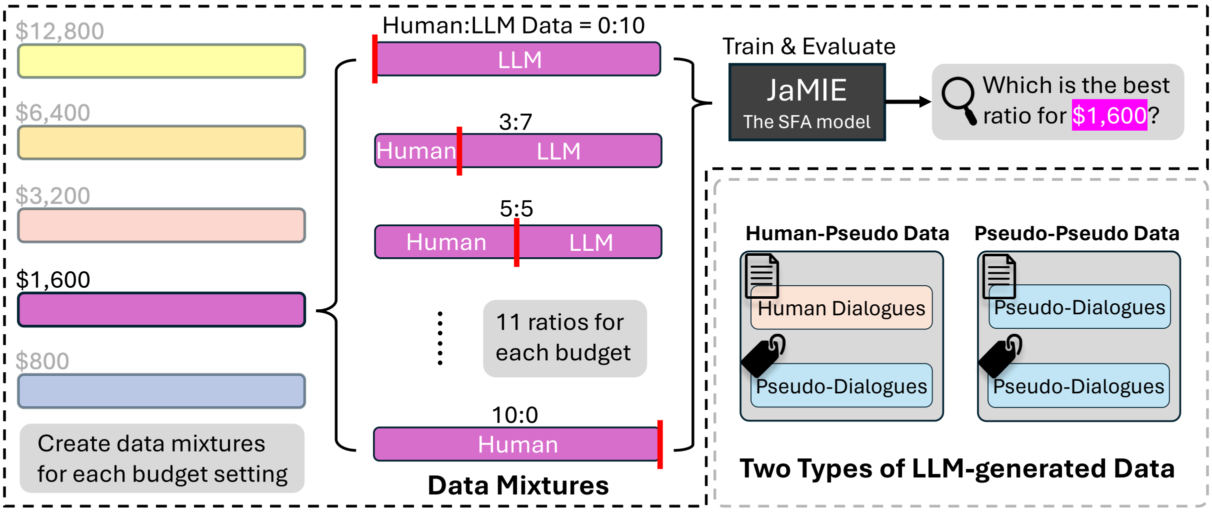

Figure 2: The overview of the cost-efficiency analysis. We mixed human data and LLM-generated data to create data mixtures up to a specific budget. The ratio of human data to LLM-generated data was adjusted in increments of 0.1. These data mixtures were then used to train our SFA model to identify the ratio that achieves optimal cost-efficiency.

be of lower quality compared to human data, it is not feasible to simply replace all human data with LLM-generated data, despite the latter being significantly cheaper. Instead, it becomes essential to consider the **trade-off** between the higher quality of human data and the lower cost of LLM-generated data. This trade-off is particularly relevant in scenarios where the budget is limited. This raises the research question: How to adjust the ratio of human to LLM-generated data within a fixed budget for optimal performance?

We address this question by synthesizing LLM-generated training data and combining it with human-annotated data to train the SLM, evaluating whether this combination achieves optimal performance within the budget (Figure 2). This process is repeated across a wide range of budget settings, from as low as $200 to as high as $12,800. For each budget level, we experiment with different ratios of human and LLM-generated data to identify the combination that maximizes cost-efficiency.

We propose a novel method for synthesizing training data using an LLM for the challenging task of SFA, generating two types of data: Human-Pseudo (HP) and Pseudo-Pseudo (PP). PP data comprises pseudo-dialogues and pseudo-labels that are both synthesized by an LLM, whereas HP data combines human dialogues sampled from a human-annotated dataset with pseudo-labels generated by the LLM. By comparing the performance of models trained on HP and PP data, we aim to determine whether the text component (dialogues) or the label component plays a more critical role in improving SFA performance in this situation.

Our empirical results reveal a clear trend across various budget levels: as the budget decreases, the optimal ratio shifts toward relying more on LLM-generated data. Conversely, when the budget is sufficiently large, incorporating LLM-generated data can actually harm performance. Another key contribution of our work is the direct comparison between HP and PP data. Our findings demonstrate that PP data is highly competitive with HP data, indicating that, in this context, replacing human-generated text with LLM-generated text is a viable and cost-effective option. We believe our findings can be applied to SFA in other technical domains or similar tasks (e.g., frame semantic parsing).

2 Related Work

Semantic Frame Analysis (SFA) / Frame Semantics in Dialogues Semantic frame analysis is a task inspired by frame-semantic parsing (FSP) and semantic role labeling (SRL). Unlike the FrameNet project used in FSP (Baker et al., 1998) or PropBank used in SRL (Kingsbury and Palmer, 2002), the frame design in semantic frame analysis differs in two ways: (1) the trigger types are domain-specific and predicate-centered, and (2) the argument types are frame-agnostic and domain-agnostic, meaning that a fixed set of argument types is used across various technical domains. Here, we refer to the process of identifying the span and type of triggers and arguments as **Trigger Detection** and **Argument Detection**.

Frame semantics can be used to capture critical information in dialogue situations. Skachkova and

Kruijff-Korbayova (2021) proposed using frame semantics in the domain of disaster response. The extracted information is used to capture and interpret verbal team communication for mission process assistance. Ebner et al. (2020) tackled argument detection in a multi-sentence setting to better capture events that span across sentences, which is similar to our setting that is done on the dialogue level. In this study, we focus on conversational SFA in Japanese interview dialogues, using the cooking section of the EIDC dataset (Okahisa et al., 2022; Chika et al., 2024) for the experiments and analyses.

Supervised Learning Models (SLMs) for SFA Previous studies have employed probabilistic model (Das et al., 2010) and RNN-based model (Swayamdipta et al., 2017) as SLMs for FSP. Kalyanpur et al. (2020) introduced Transformer-based models (Vaswani et al., 2017) to FSP, utilizing a sequence-to-sequence Transformer model and framing FSP as a text generation task by tagging entities with index numbers for tokens. In Matta et al. (2023), an encoder transformer model was used to address SFA in a cascaded manner: first, a trigger detection model identifies triggers within the context, and then a separate argument detection model determines the arguments for each trigger. However, we are concerned that this cascaded approach might introduce error propagation. Therefore, in this paper, we adopt JaMIE (Cheng et al., 2022), an encoder-centric model that simultaneously detects entities and their relations, offering an end-to-end solution for SFA.

LLMs for SFA-like tasks While no existing work directly targets SFA using LLMs, recent studies have explored related tasks, such as named entity recognition (NER) and relation extraction (RE). Wang et al. (2023a) reformulated NER as a text-generation task by wrapping entities in tag pairs, allowing LLMs to process them efficiently. Zhang et al. (2023) and Wan et al. (2023) enhanced LLM performance on RE tasks by improving prompt design. Sun et al. (2023) tackled various NLP tasks, including NER and RE, by utilizing improved prompting and few-shot retrieval methods, similar to the approaches in Wang et al. (2023a) and Wan et al. (2023). These studies, along with the method proposed by Kalyanpur et al. (2020), have inspired our prompt design for SFA using an LLM (Figure 4).

LLMs as Data Synthesizers There have been numerous efforts to utilize LLMs for generating synthetic data to train SLMs. Wang et al. (2021) utilized few-shot GPT-3 to generate labels for natural language understanding and generation tasks, achieving performance comparable to human labeling while significantly reducing costs. Ding et al. (2023) explored various methodologies for generating labeled data using GPT-3 and demonstrated results on par with human-labeled data in tasks such as sentiment triplet extraction. He et al. (2024) employed GPT-3.5 with chain-of-thought reasoning (Wei et al., 2023) as an alternative to crowdsourced annotators, demonstrating performance that was either superior to or on par with human annotators. However, these studies focus on tasks that are less complex than SFA. They either involve a single label per sequence, extract fewer entities, or do not include relations. Additionally, they do not provide an analysis of the trade-off between human and LLM-generated data.

3 Preliminaries

We define Semantic Frame Analysis (SFA) and introduce the EIDC dataset (Okahisa et al., 2022; Chika et al., 2024), which contains SFA annotations and is used in this study.

3.1 Semantic Frame Analysis (SFA)

Semantic frame analysis aims to extract semantic frames, which represent events, in a given context. The core of a semantic frame is a **trigger**, which is a predicate and the main action of the event. Since each frame has only one trigger, we refer to the frame type by the trigger type from now on without further notice. The event can also include associated details, such as the object, instrument, or temperature, referred to as frame **arguments**, linked to the event-evoking trigger. Note that different from frame designs such as the FrameNet project (Baker et al., 1998), argument types in the EIDC dataset are designed to be both frame-agnostic and domain-agnostic, meaning all frames can accept arguments such as *Object*, *Time*, *Manner*, etc.

SFA consists of two parts: **Trigger Detection** and **Argument Detection**. In trigger detection, the task is to identify the spans of triggers and classify their types, which functions similarly to a named entity recognition task. In argument detection, the goal is not only to identify the spans and types of arguments but also to determine their associated triggers. During evaluation, an argument prediction is considered incorrect if its association with a trigger is wrong, even if the span and type are correctly identified. Additionally, a single trigger can have multiple associated arguments. Our proposed data synthesis method (Section 4.2) can generate data for SFA while adhering to these conditions.

3.2 Technical Interview Dialogue Dataset with SFA Annotation

In this paper, we utilize the *cooking* section of the EIDC dataset (Okahisa et al., 2022; Chika et al., 2024). Note that when referring to the EIDC dataset, we specifically mean the cooking section unless stated otherwise. Examples of dialogues and SFA annotations in this domain are presented in Figure 1 and 12.

Technical Interview Dialogues The EIDC dataset contains interview dialogues where an expert discusses cooking processes with an interviewer. The expert introduces and explains a recipe spontaneously or in response to the interviewer's questions. The interviewer is asked to actively elicit knowledge about the cooking process through interactions, such as asking questions.

Annotation for Semantic Frame Analysis Each dialogue in the EIDC dataset comes with manual annotations of SFA. Human annotators manually assign labels to the dialogues with reference to the annotation guideline, which defines how to label entities and relations in the context and provides demonstrations. We also extracted these information from the annotation guideline and used them in the system prompt for the LLM. The trigger types represents cooking actions such as bake frying and cutting because the semantic frames are designed to capture cooking-related events. A complete list of entity types can be found in Appendix A.6. The original paper by Chika et al. (2024) presents inter-annotator agreement scores, including Cohen's kappa, to demonstrate annotation quality.

4 Data Synthesis With an LLM

This section presents our methodology for constructing training data for conversational semantic frame analysis using an LLM.

4.1 Pseudo-dialogue Generation

To generate pseudo-dialogues, the LLM is prompted with few-shot dialogues and asked to generate new ones that are close to the few-shots in format but contain different contents (Figure 3). For the few-shot examples, we not only sample from a preserved pool of human dialogues but also adopt the self-instruct strategy (Wang et al., 2023b) to sample from the previously generated pseudo-dialogues to increase diversity. The pre-filtering and post-filtering methods, along with the detailed settings for the self-instruction of pseudo-dialogues, are explained in Section 5.1.

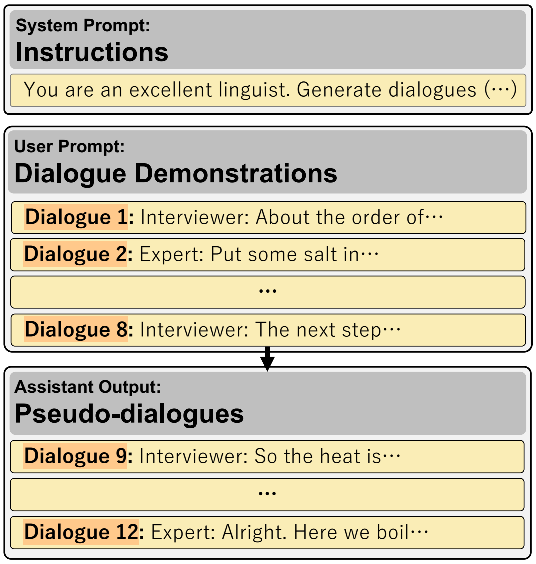

Figure 3: The overview of the prompt design for pseudo-dialogue generation. Refer to the actual prompt design in Appendix A.3.

4.2 Pseudo-labels by LLM

We apply pseudo-labels to the dialogues via a novel three-step tagging and labeling prompting scheme that converts SFA into a text generation task. An example of this pseudo-labeling process is illustrated in Figure 4. The steps are as follows, given an input context:

1. **Entity Tagging**: Insert entity tags (<En> and </En>, $n \in \mathbb{N}$) to mark the start and end of entities.

2. **Trigger Detection**: Identify the triggers among the entities tagged in Step 1.

3. **Relation Detection**: Determine argument relations among the entities tagged in Step 1.

This output format captures multiple entities and relations simultaneously and can be easily converted into the data format required by the SLM. We provide type definitions as outlined in the annotation guidelines within the system prompt and

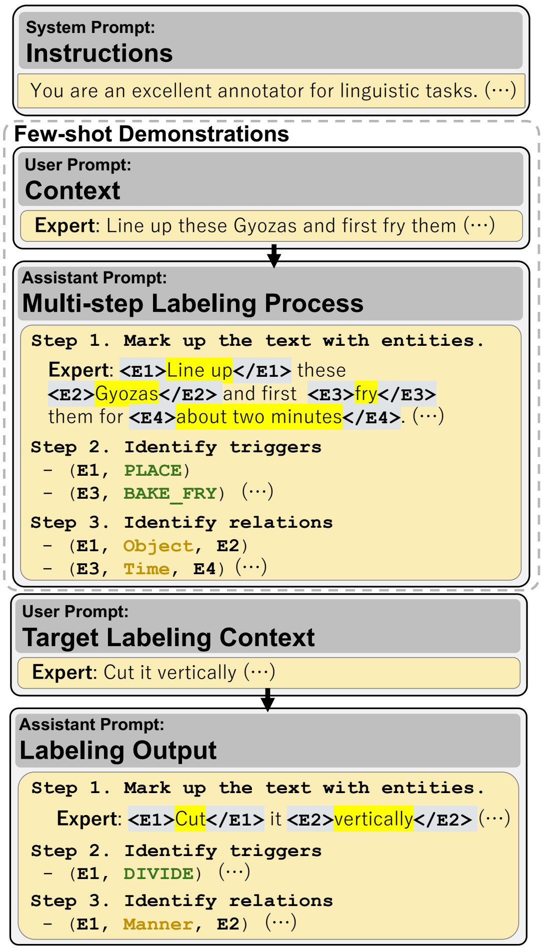

Figure 4: We designed a novel multi-step labeling scheme for LLMs to handle SFA in text generation. Refer to the full prompt design in Appendix A.4.

demonstrate the tagging process using a few-shot approach.

4.3 Data Variants

We construct three data variants used in this study: Human-Human (HH), Human-Pseudo (HP), and Pseudo-Pseudo (PP). In this context, "Human" refers to data collected from humans, while "Pseudo" denotes data generated by an LLM. We did not consider a Pseudo-Human variant because human annotation is too precious to be assigned to lower-quality LLM-generated dialogues.

Human-Human (HH) We sampled human dialogues and labels directly from the EIDC dataset and formed HH data. The Human-Human data is the most expensive and is also expected to have the highest label accuracy, closely aligning with the desired standards defined in the annotation guidelines.

Human-Pseudo (HP) In HP data, SFA labels are assigned by an LLM to human dialogues sampled from the EIDC dataset using the pseudo-labeling method from Section 4.2.

Pseudo-Pseudo (PP) PP data is fully synthesized, with LLM-generated dialogues and labels.

5 Experimental Settings

To study how to achieve optimal cost-efficiency by collecting both human and LLM-generated data with a fixed budget, we conducted the following steps for the experiments.

1. **Collecting data**: Sample/synthesize Human-Human (HH), Human-Pseudo (HP), and Pseudo-Pseudo (PP) data.
2. **Defining budget settings**: Define a range of budgets to simulate the fixed budget scenario.
3. **Creating HH+HP and HH+PP mixtures**: For each budget setting, construct human and LLM-generated data mixtures to simulate budget allocations.
4. **Training and evaluating the SFA model**: Train the SFA model using the data mixtures and evaluate its performance to identify the optimal data ratio.

In the following sections, we provide detailed descriptions of these steps. An overview of the cost-efficiency analysis is demonstrated in Figure 2.

5.1 Details of Data Synthesis Procedures

We provide details on the data synthesis procedures. We reserved 3 dialogues[1] from the EIDC training data as few-shot examples for both the pseudo-dialogue generation and pseudo-labeling process.

Pseudo Dialogue Generator As introduced in Section 4.1, we adopted the self-instruct strategy (Wang et al., 2023b) to bootstrap pseudo-dialogue generation. Following the settings in their work, we provide the model with 6 human dialogues and 2 pseudo-dialogues as few-shots. We synthesized the first 100 pseudo-dialogues with only human dialogues as few-shots. Afterward, we moved on to mixing few-shot examples. Before adding pseudo-dialogues back into the dialogue pool, we filtered them by ROUGE-L score (<0.7) against

[1]To fit within the context length limits of both the LLM and the SLM, we divide dialogues into smaller **sessions** using a heuristic method. Hereafter, a 'dialogue' will refer to a 'dialogue session' unless otherwise specified. Each session consists of up to 10 utterances.

Data Type	Data Size (Sessions)	Cost		
		Text ($)	Label ($)	Total ($)
Human-Human	1,472	6.4k	6.4k	12.8k
Human-Pseudo	2,858	12.4k	0.37k	12.8k
Pseudo-Pseudo	4,293	0.28k	0.56k	0.84k

Table 1: The size and cost statistics of the three data variants.

existing dialogues to ensure that the newly generated ones were not extremely similar to the existing ones. None of the pseudo-dialogues exceeded this limit. We then filtered the most similar ones using ROUGE-L to reduce them to the desired size shown in Table 1, which ended with a max ROUGE-L score of 0.52. We used GPT-4-0613 (accessed 01/2024) and set the generation temperature to 0.7, the presence penalty to 2.

Pseudo SFA Labeler We adopted GPT-4-0613 (accessed 01/2024) to generate pseudo-labels for SFA. For few-shots, we sampled 3 complete human dialogues, then filtered them to remove sessions with too few entities, resulting in 37 dialogue sessions. For each labeling target, we used 3 few-shots: the top 2 most similar dialogue sessions, determined by the ROUGE-L score to ensure similarity to the target, and 1 specially preserved dialogue session containing as many as 30 entities. This special few-shot was included in all cases because we empirically observed that GPT-4 tends to overlook entities if the few-shots lack sufficient entities.We conducted an ablation study to determine this prompt design, which we report in Appendix A.1. We further provide a case analysis of LLM-generated labels in Appendix A.2.

5.2 Data and Budget Settings

We provide details on the data statistics, data mixtures, and budget settings.

Data Statistics As shown in Table 1, we collected up to $12,800 for both HH and HP data, which roughly aligns with the three-year total of scholarship funds for a PhD student at a Japanese university.[2] For HH data, we sampled $12,800 worth of human dialogue and label pairs from the EIDC dataset, out of a maximum of 4,600 instances and a total cost of $40,000 of the original dataset. For HP data, we repeatedly sampled human dialogues in the EIDC dataset and then applied pseudo-labels to them until the cost reached $12,800, which was calculated based on the cumulative costs of the human dialogues and OpenAI API usage. For PP data, due to the low cost of both pseudo-dialogue and pseudo-labels, we collected 1.5x times the data size compared to HP data while only costing $840. The costs for pseudo-dialogues and pseudo-labels were also calculated from the token usage of the OpenAI API service. We ceased further collection of PP data upon discovering that performance had reached saturation and would not improve with additional data.

We conducted a quantitative analysis comparing human dialogues and pseudo-dialogues. We found that the average length of pseudo-dialogues generated by GPT-4 was similar to that of human dialogues (127 tokens vs. 136 tokens) and exhibited fewer extreme outliers in terms of length. By comparing the label density of HP and PP data, we observed that pseudo-dialogues tended to contain more entities than human dialogues, leading to a higher count for certain label types. For more details on the length and label distributions of pseudo-dialogues, refer to Appendix A.5 and Appendix A.6.

Data Mixtures We create two types of data mixtures: **HH+HP**[3] and **HH+PP mixtures**, to simulate the situation where one collects human data and LLM-generated data at the same time. Refer to Appendix A.8 for a demonstration of the budget allocation between the two types of data.

Budget Settings We set different budget ranges for the HH+HP mixture and the HH+PP mixture, with the budget range for the latter being lower due to the significantly lower cost of PP data. For each budget, we adjust the proportion of HH data within the budget from 0 to 1 with an interval of 0.1, creating 11 ratio variants for each budget level.

- For HH+HP mixture ($):
  800, 1,200, 1,600, 3,200, 6,400, 12,800

[2]We excluded the collection cost of few-shot examples sampled from the training split of the EIDC dataset, as well as the instructions derived from the annotation guidelines.

[3]When creating HH+HP mixtures, we avoided choosing data with the same human dialogues to avoid confusion to the SFA model.

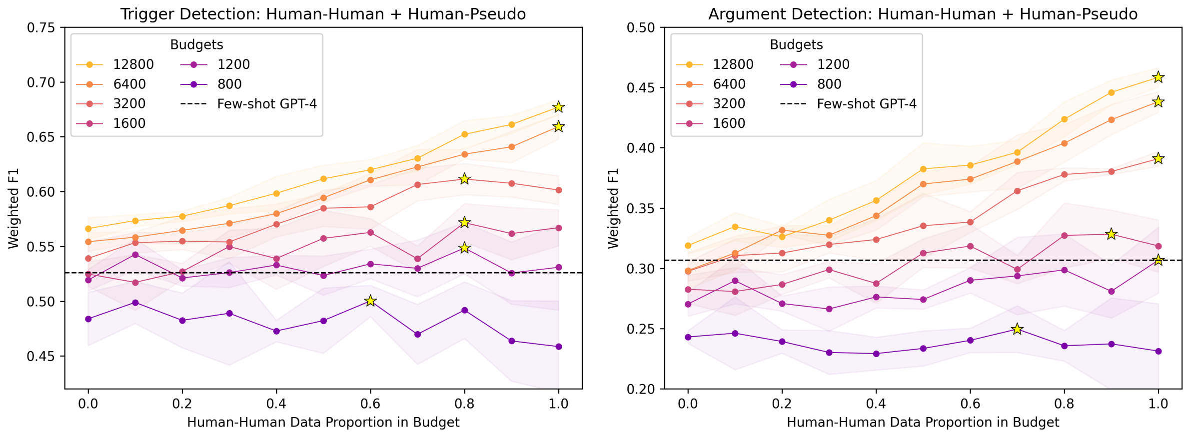

Figure 5: The cost-efficiency plot for HH+HP mixture. The black dotted line represents the performance of few-shot GPT-4. Each budget curve features a star marking its optimal point. The shaded region around each curve indicates the standard deviation across five different seeds.

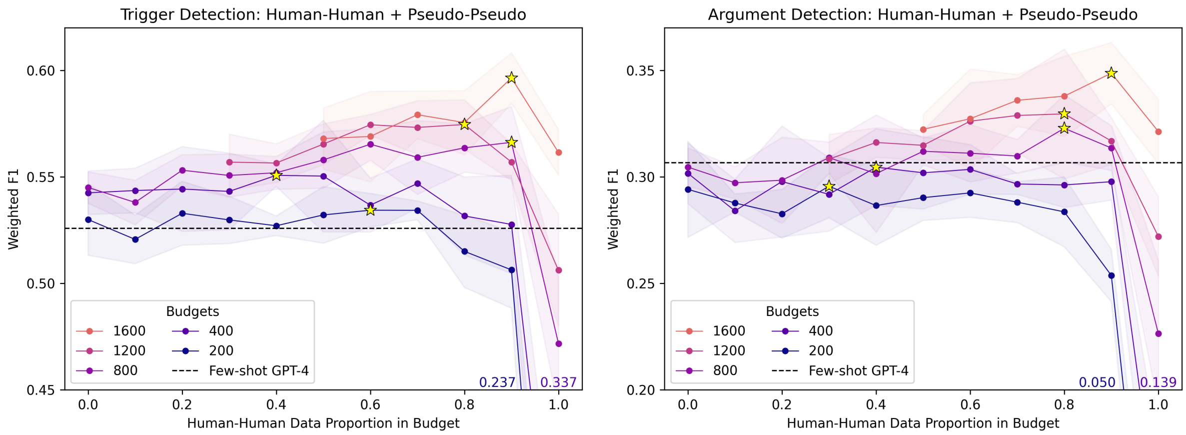

Figure 6: The cost-efficiency plot for HH+PP mixture. Due to the collection limit of $840 worth of PP data, the plot only shows the right portion of the curve for budgets $1,200 and $1,600, where the data is combined with HH data. The values of some outlier points are displayed on the plot with colors corresponding to the budget curve.

- For HH+PP mixture ($):
  200, 400, 800, 1,200, 1,600

5.3 Supervised Learning Model and Evaluation Metrics for SFA

We adopt JaMIE (Cheng et al., 2022) as our supervised learning model (SLM) for SFA. JaMIE is an architecture featuring one transformer encoder and multiple decoding heads for sequence labeling and can handle relation extraction by design. We employ the Japanese DeBERTa-V2-base as the pre-trained encoder for JaMIE and train the decoding heads from scratch.[4] Refer to the training hyperparameters in Appendix A.7.

We evaluated the performance of Trigger Detection and Argument Detection using a classification metric that accounts for both the type and span accuracy of entities.[5] Correct predictions require both the entity's type and span to be accurate. We award partial scores if the predicted entity's type is correct but the span only overlaps with the true answer. Argument predictions are marked false if their associated trigger is incorrect.[6] The overall performance is measured using a weighted F1 score, aggregated from the F1 scores of each class.

6 Results and Analyses

The objectives of the cost-efficiency analysis are as follows:

1. **Optimal Data Ratio**: What is the optimal ratio for combining human data and LLM-generated data within a limited budget? Is the ratio budget-dependent?

2. **HP vs. PP**: Should one pay more to collect human-dialogues instead of pseudo-dialogues

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

[5] We modified the evaluation code from seqeval (https://github.com/chakki-works/seqeval).

[6] In addition to semantic frames, the data also included Event Coreference Relations (ECR). We did not evaluate ECR directly, however, we evaluated argument detection by allowing the target trigger to be any of the events on the same ECR event sequence in the true labels.

for a potential performance increase?

We analyze the experimental results to answer these objectives in the following sections.

6.1 Cost-efficiency Analysis

In this section, we address the first objective: optimal data ratio for HH+HP and HH+PP mixtures, and if it is budget-dependent.

HH+HP Mixture In Figure 5, we observe that when the budget is lower than $6,400 for trigger detection and $3,200 for argument detection, optimal cost-efficiency is achieved by combining HH and HP data. The lower the budget is, the more HP data should be included for best performance. In this case, the trade-off between human data and LLM-generated data has a positive impact on the performance.

On the other hand, we see that when the budget is higher than above, the optimal cost-efficiency is brought by using 100% HH data. This shows that LLM-generated data cannot be used in all situations because it may harm the performance.

HH+PP Mixture In Figure 6, we see that for all the budgets we set, the optimal performance was achieved by combining HH and PP data. We specifically observed that since PP data is so much cheaper, allocating 10% of the budget to PP data in budget $1,600 brought a significant performance boost for both trigger and argument detection. Although we did not further raise the budget for PP data, we can estimate that the optimal will be achieved by using 100% HH data if we raise the budget to $6,400 and above. Therefore, we conclude that when the budget is not high enough to reach saturation (optimal performance by using 100% HH data), one should combine human and LLM-generated data and adjust the ratio to using more LLM-generated data as the budget declines.

6.2 Human-Pseudo vs. Pseudo-Pseudo

We further investigated the second objective: is HP data better than PP data for having human dialogues instead of pseudo-dialogues?

We observed no significant disadvantage caused by replacing human dialogues with pseudo-dialogues for LLM-generated data. In fact, with the same budget of $1,600, one could achieve a slightly higher performance in trigger detection using PP data compared to HP data (0.596 in Figure 6 vs. 0.571 in Figure 5). Therefore, from a cost-sensitive perspective, PP data is a superior option.

6.3 Data Augmentation for Low-resource Setting

We review the effectiveness of LLM-generated data from a data augmentation perspective (Figure 7). In this setting, we trained the SLM first using all LLM-generated data, i.e., either all HP or PP data, then continued training it on different costs of HH data, ranging from $800 to $12,800. The result shows that when the amount of HH data is limited (lower than $3,200), both HP and PP data help boost performance. The effectiveness of LLM-generated data is more significant when the budget for HH data is low. Notably, while the cost of PP data is significantly cheaper than HP data in this setting ($840 vs. $12,800), the former is arguably competitive against the latter as the max performance gap (green line vs. red line) is less than 0.02 F1 score.

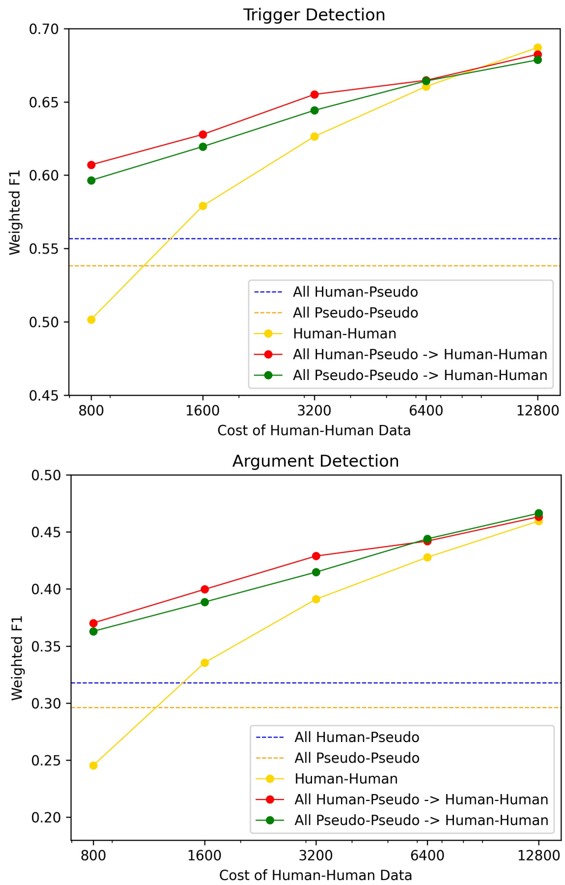

Figure 7: The effectiveness of LLM-generated data from a data augmentation perspective. We trained the SLM on all HP or PP data (blue and orange dotted lines), then continued training on different sizes of HH data (red and green lines).

7 Conclusion

In this paper, we conducted a comprehensive analysis to evaluate the cost-efficiency to combine

LLM-generated data with human-annotated data for Japanese conversational semantic frame analysis under various budget constraints. We proposed a novel method to synthesize two types of training data: Human-Pseudo (HP) data and Pseudo-Pseudo (PP) data, for the experiments and analyses. Our findings indicate that the ideal ratio to combine human and LLM-generated data is budget-dependent, with a tendency to favor a higher proportion of LLM-generated data as the budget decreases. Furthermore, our results suggest that fully synthesized data (PP data) is a viable option, as it is significantly cheaper while maintaining comparable performance levels to the half-synthesized counterpart (HP data). In future work, we aim to extend our analysis to other domains and tasks to validate the generalizability of our findings.

Limitations

While we believe our conclusions are comprehensive within our experimental settings, our work has several limitations. Firstly, determining the **exact** ratio of human to LLM-generated data remains challenging, as it depends on factors such as the specific task, dataset characteristics, and budget constraints. Secondly, we only focused on the task of SFA in the cooking domain in this work. We hope that future work could extend the findings of our work to other domains and related tasks.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

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.

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

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

Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A. Smith. 2010. Probabilistic frame-semantic parsing. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 948–956, Los Angeles, California. Association for Computational Linguistics.

Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. 2023. Is GPT-3 a good data annotator? In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 11173–11195, Toronto, Canada. Association for Computational Linguistics.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8057–8077, Online. Association for Computational Linguistics.

Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2024. AnnoLLM: Making large language models to be better crowdsourced annotators. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track), pages 165–190, Mexico City, Mexico. Association for Computational Linguistics.

Martin Josifoski, Marija Sakota, Maxime Peyrard, and Robert West. 2023. Exploiting asymmetry for synthetic training data generation: SynthIE and the case of information extraction. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 1555–1574, Singapore. Association for Computational Linguistics.

Aditya Kalyanpur, Or Biran, Tom Breloff, Jennifer Chu-Carroll, Ariel Diertani, Owen Rambow, and Mark Sammons. 2020. Open-domain frame semantic parsing using transformers. Preprint, arXiv:2010.10998.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02), Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 10572–10601, Singapore. Association for Computational Linguistics.

Shiho Matta, Yin Jou Huang, Hirokazu Kiyomaru, and Sadao Kurohashi. 2023. Utilizing pseudo dialogue in conversational semantic frame analysis. ANLP2023.

Taro Okahisa, Ribeka Tanaka, Takashi Kodama, Yin Jou Huang, and Sadao Kurohashi. 2022. Constructing a culinary interview dialogue corpus with video conferencing tool. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pages 3131–3139, Marseille, France. European Language Resources Association.

OpenAI. 2024. Gpt-4 technical report. Preprint, arXiv:2303.08774.

Natalia Skachkova and Ivana Kruijff-Korbayova. 2021. Automatic assignment of semantic frames in disaster response team communication dialogues. In Proceedings of the 14th International Conference on Computational Semantics (IWCS), pages 93–109, Groningen, The Netherlands (online). Association for Computational Linguistics.

Xiaofei Sun, Linfeng Dong, Xiaoya Li, Zhen Wan, Shuhe Wang, Tianwei Zhang, Jiwei Li, Fei Cheng, Lingjuan Lyu, Fei Wu, and Guoyin Wang. 2023. Pushing the limits of chatgpt on nlp tasks. Preprint, arXiv:2306.09719.

Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A. Smith. 2017. Frame-semantic parsing with softmax-margin segmental rnns and a syntactic scaffold. Preprint, arXiv:1706.09528.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. Preprint, arXiv:1706.03762.

Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. Gpt-re: In-context learning for relation extraction using large language models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 3534–3547.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023a. Gpt-ner: Named entity recognition via large language models. arXiv preprint arXiv:2304.10428.

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Want to reduce labeling cost? GPT-3 can help. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4195–4205, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-instruct: Aligning language models with self-generated instructions. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. Preprint, arXiv:2201.11903.

Kai Zhang, Bernal Jiménez Gutiérrez, and Yu Su. 2023. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. arXiv preprint arXiv:2305.11159.

A Appendix

A.1 Ablation for Prompt Design for Pseudo-labeling

We conducted an ablation study to determine the impact of different prompt design choices on pseudo-labeling performance and adopted one of the top-performing prompt designs. We evaluate the effects of varying instruction styles and few-shot selection strategies by measuring the performance of few-shot LLMs on the validation dataset, as detailed below:

- **Instruction Style**
  - w/ entity demo.: The instruction includes entity demonstrations.
  - wo/ entity demo.: The instruction only has a description for each entity type, but no examples are provided (Figure 13).
- **Few-shot Selection Methods**
  - By ROUGE-L: Examples are selected based on the highest ROUGE-L similarity score to the input.
  - Mandatory: A single hand-picked example that is entity-rich, containing up to 30 entities, is always included.
  - Random: Examples are randomly selected from the few-shot pool.

We observed that enabling ROUGE-L-based few-shot retrieval, incorporating the mandatory few-shot example, and providing entity demonstrations in the instruction generally improved performance. Additionally, not all LLMs performed well on SFA. For instance, GPT-4-1106-preview occasionally failed to recognize entities in the context, even when they were present. GPT-3.5-turbo-0125 exhibited similar errors but also struggled with output formatting, sometimes producing invalid outputs that had to be evaluated as empty predictions. Moreover, it suffered from hallucinations, generating non-existent entity types. Based on these observations, we conclude that SFA requires LLMs at least at the GPT-4 level to achieve reliable performance.

A.2 Case Analysis on LLM-generated labels

We conduct an error case analysis on two common types of mistakes made by the LLM during the

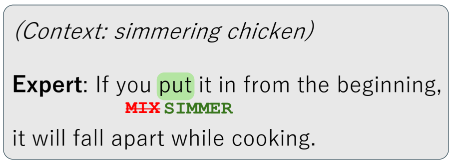

Figure 8: It is difficult for LLM to label correctly when it is necessary to infer the entity type from the context.

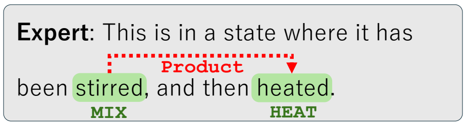

Figure 9: It is difficult for LLM to handle complex relations, such as *Product*.

pseudo-labeling process. These cases were identified by comparing HH and HP data, both of which contain the same human dialogues.

In the first case (Figure 8), the expert and interviewer discuss a simmering process in the preceding context. In this context, the action of *put* refers to placing something into boiling water and should therefore be labeled as *SIMMER*. However, the LLM tends to interpret the word literally, labeling it as *MIX* instead. It is challenging to instruct the model to account for this type of inference accurately.

Another common challenge for the LLM is handling complex argument relations, such as *Product*. *Product* is a unique type of argument that requires the argument itself to be an existing trigger. In Figure 9, the predicate *stirred* functions both as a *MIX* type trigger and as a *Product* argument for the trigger *heated*. However, the LLM failed to recognize the *Product* argument relation.

A.3 Prompt For Pseudo-dialogue Generation By LLM

An example of the prompt for pseudo-dialogue generation is shown in Figure 14.

A.4 Prompt For LLM SFA Labeling

The adopted prompt design for SFA labeling is shown in Figures 15, 16, and 17.

A.5 Length Distribution of Pseudo-dialogues

We present the length distributions of human dialogues and pseudo-dialogues in Figure 11. We observed that GPT-4 generally followed the length specification in the instruction, resulting in an average length of 127 tokens (token count by Japanese DeBERTa-V2 tokenizer) compared to an average of 136 tokens in human dialogue sessions. Moreover, pseudo-dialogues have a more short-tailed distribution, which means there are fewer extremely short or long outliers.

A.6 Label Distribution in Pseudo-dialogues

We present the label distributions across three data types: Human-Human, Human-Pseudo, and Pseudo-Pseudo in Figure 10. When comparing Human-Human to Human-Pseudo, we observe that replacing human labelers with GPT-4 leads to fluctuations in certain label types. Specifically, there is a decrease in types such as "BAKE_FRY" and "SIMMER" in triggers and "Manner" in arguments, and an increase in types like "PLACE" in triggers and "Instrument" in arguments. While we believe that these fluctuations will not be a significant issue, it is important to point out that in addition to the fluctuations, the labels generated by GPT-4 may not be accurate either.

When comparing Human-Pseudo to Pseudo-Pseudo, we observe that replacing human dialogues with pseudo-dialogues leads to a higher frequency of certain types than in human dialogues. For example, types like "MIX" and "BAKE_FRY" in triggers and all argument types appear more frequently. This increase occurs because GPT-4 tends to fit a whole story into a pseudo-dialogue, resulting in a higher overall entity count. In contrast, human dialogues are heuristically cut into smaller sessions, which can lead to fewer entities per session. Also, the increase in trigger types "MIX" and "BAKE_FRY" indicates that GPT-4 tends to mention these specific events, creating a bias toward specific topics.

A.7 Training Hyperparameters for the SLM

We adopted JaMIE (Cheng et al., 2022) as our SLM for SFA. For the encoder, we used a pretrained Japanese DeBERTa-V2-base model with an encoder learning rate of 2e-5 and a relation decoder learning rate of 1e-2, without a learning rate schedule.[7] The model was trained for up to 30 epochs, and the best checkpoint was selected based on the highest validation weighted F1 score. The validation and test sets are defined in the EIDC dataset with sizes of 269 and 379 dialogue sessions, respectively.

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

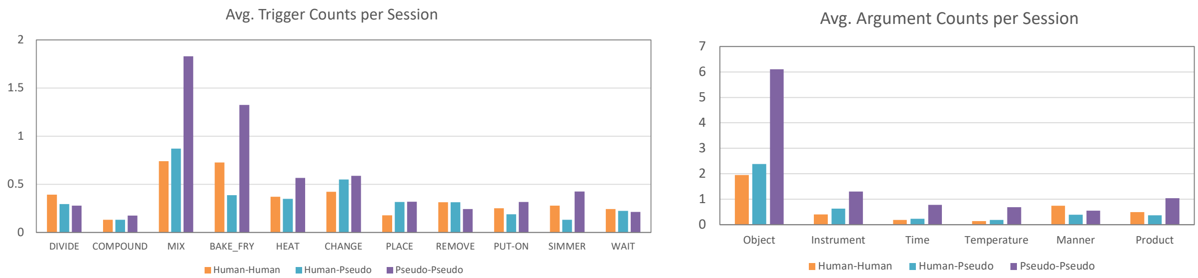

Figure 10: Trigger and argument label distribution.

LLM	Instruction	Few-shot Selection	T. F1	Arg. F1
GPT-3.5-turbo-0125	w/ entity demo.	2 by ROUGE-L + 1 mandatory	0.434	0.170
GPT-4-1106-preview	w/ entity demo.	2 by ROUGE-L + 1 mandatory	0.484	0.256
GPT-4-0613	w/ entity demo.	3 random	0.484	0.269
	w/ entity demo.	3 by ROUGE-L	0.519	0.277
	w/ entity demo.	2 random + 1 mandatory	0.513	0.293
	w/ entity demo.	1 by ROUGE-L + 1 mandatory	**0.519**	0.303
	w/o entity demo.	2 by ROUGE-L + 1 mandatory	0.460	0.245
GPT-4-0613†	w/ entity demo.	2 by ROUGE-L + 1 mandatory	0.514	**0.314**

Table 2: Ablation study on prompt design for pseudo-labeling. T. F1 and Arg. F1 denote the weighted-F1 scores for trigger and argument detection, respectively. † indicates the final prompt design chosen for pseudo-labeling. Performance is measured on the validation set.

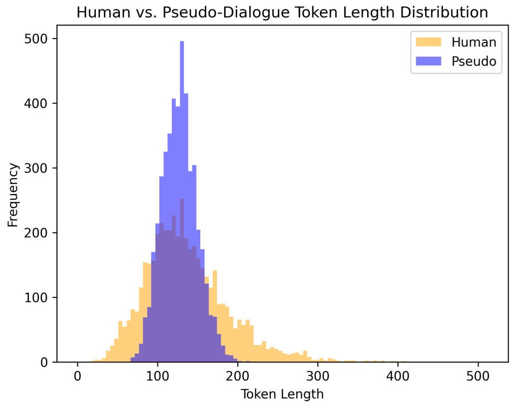

Figure 11: The length distributions of human and pseudo-dialogues.

A.8 Demonstration for Budget Allocation

For example, when one has $1,600 of budget and wants to allocate 30% ($480) of that to HH data and 70% ($1,120) to HP data, the final mixture will contain 55 instances of HH data and 250 instances of HP data.

- $1,600 (30% HH, 70% HP) =
  55 ($480) HH + 250 ($1,120) HP

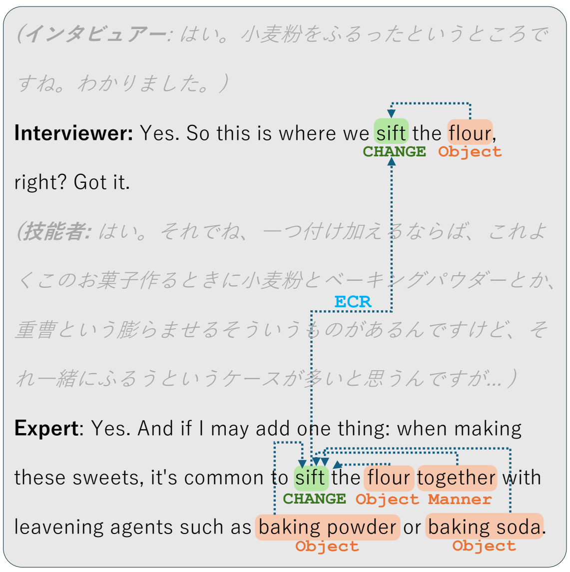

Figure 12: This human-annotated data example demonstrates that (1) the same event is mentioned across multiple utterances, (2) a single trigger can have multiple same type arguments, and (3) an ECR relation is present, although it is not directly evaluated in this paper. The example is translated from Japanese, and the original text is provided in gray italic font.

BAKE_FRY: 油を用いて火や熱源で調理する。~~（例：焼く、揚げる、炒める、焦がす、ローストする）~~

Figure 13: Without entity demonstration means to remove ~~the examples~~, only keeping the description.

```
System Prompt:

You are an excellent linguist.
Generate dialogues that are similar to the given dialogue
context's length, but have different content.
The dialogue should include an interviewer (インタビュアー)
and an expert (技能者), who will talk about the cooking
process of a dish.
The expert will try to explain the details of the cooking
process, and the interviewer will ask questions to clarify
the details.
Try to add details to the dialogue, such as the tools,
techniques, time, temperature, etc.
Dialogues can start or end halfway through the cooking
process.
Make sure each dialogue contains more than 3 turns.
Do NOT copy from existing dialogue context, make sure each
dialogue is diverse and creative.
Each dialogue must be in a single line, like the
demonstration from the user. Only start new lines when you
want to start a new dialogue piece.
The user will provide you with 8 example dialogue pieces.
You must continue generating from Dialogue 9 to Dialogue 12.

Few-shots:

Dialogue 1: 技能者:これは先ほどのレンコンのみじん切りと、あと長ネギ2
分の1本ぐらいを、これもみじん切りにしておきます。インタビュアー:うん、2
分の1本結構ね量を使われるんですね。技能者:そうですね、はい。インタビュ
アー:ヘルシー。技能者:はい。インタビュアー:もしネギ、青ネギとかでも大丈
夫なんですかね？技能者:もちろん大丈夫です。インタビュアー:大丈夫ですか。
技能者:はい。あと玉ねぎでも大丈夫です。インタビュアー:玉ねぎでも。いろ
いろ使えますね。家にある、あるものをなんか。

Dialogue 2: インタビュアー:このパスタソースの作り方について教えていた
だけますか？技能者:まずオリーブ油とにんにくをフライパンに入れ、香ばしい
匂いが立つまで炒めます。次にトマト缶を加えて5分ほど煮込みます。インタ
ビュアー:その後に塩コショウで調味するんですよね？技能者:はい、そして最
後にベーシックなパスタソースが完成します。あとはお好みでバジルやチーズ
を加えることもできます。インタビュアー:パスタは何分茹でると良いですか？
技能者:袋に記載されている時間通りに茹でると最適な硬さになりますよ。
(…)

Model Output:

Dialogue 9: インタビュアー:この焼き鳥の作り方について教えてもらえます
か？技能者:まず、鶏肉を一口大に切り分け、竹串に刺します。次に塩とコショ
ウで下味をつけ、炭火でじっくりと焼き上げます。インタビュアー:それはどれ
くらいの時間焼くのですか？技能者:各面がキツネ色になるまで約5分間ほど焼
きます。そして最後に特製のたれを塗って仕上げます。インタビュアー:その特
製のたれは何を使っていますか？技能者:主にしょうゆ、みりん、砂糖を基本と
しており、あとは秘密のスパイスが入っていますよ。
(…)
```

Figure 14: The prompt design for pseudo-dialogue generation. In this example, Dialogue 1 is a human dialogue, and Dialogue 2 is a previously generated pseudo-dialogue.

```
System Prompt:
You are an excellent annotator for linguistic tasks. You will be provided
with a piece of Japanese dialogue. Your task is to annotate it with labels
for semantic frame analysis.

Step 1. Mark up the text with entities.
Mark up the text using the following tags:
- <En>: The start of the n-th entity
- </En>: The end of the n-th entity

2. Identify triggers
Identify triggers from the extracted entities. Trigger types include:
- BAKE_FRY: 油を用いて火や熱源で調理する。（例：焼く、揚げる、炒める、焦がす、ローストす
る）
- DIVIDE: 何かの全体ないしは部分を2つ以上に分ける。（例：切る、刻む、割く、ほぐす、割る、
砕く、カットする、スライスする）
- CHANGE: 形状・向き・温度を変化させる。（例：擦る、溶かす、冷ます、ひっくり返す、泡立て
る、潰す、広げる、伸ばす、こねる、発酵させる）
(…)
3. Identify Relations
Identify relations between entities. There are 3 categories of relations:
Argument, ECR and Product. Arguments should be further determined by types:
- Object: 当該の料理イベントの主たる材料を表す要素。（例：豆腐、人参、お肉、じゃがいも、
葉っぱ）
- Instrument: 当該の料理イベントを実行するために用いる道具や調理器具を表す要素。（例：包
丁、フライパン、冷水、泡立て器、鍋、ベーキングシート）「包丁で切る」、「泡立て器でよくかき
混ぜる」、「ひじきを冷水で戻す」などの形で現れる。
(…)
If two event triggers indeed refer to the same event/action, you should
label their relation as ECR. As for Product, if E2 uses the outcome from E1,
then you should label the two (E2, Product, E1).
```

Figure 15: The system prompt for the LLM. Some type definitions are omitted.

```
Few-shots:
user
技能者:これは豚挽肉をボウルに入れたところです。はい。
インタビュアー:お願いします。
技能者:はい。室温にし、室温でよくこねておきます。はい。
インタビュアー:はい。よくこねるというのは、だいたいどの状態が目安になりますか？
技能者:そうですね、ちょっとモチモチってなるぐらいまでこねていただければ。はい、そんなにハン
バーグのときほどこねないんですけれども。はい。
インタビュアー:はい。ハンバーグほどではない、なく、でもしっかりとこねる。
assistant
Step 1. Mark up the text with entities.
技能者:これは<E1>豚挽肉</E1>を<E2>ボウル</E2>に入れたところです。はい。
インタビュアー:お願いします。
技能者:はい。室温にし、<E3>室温</E3>で<E4>よく</E4><E5>こねて</E5>おきます。はい。
インタビュアー:はい。よくこねるというのは、だいたいどの状態が目安になりますか？
技能者:そうですね、ちょっとモチモチってなるぐらいまで<E6>こねて</E6>いただければ。はい、
そんなに<E7>ハンバーグのときほど</E7><E8>こね</E8>ないんですけれども。はい。
インタビュアー:はい。ハンバーグほどではない、なく、でも<E9>しっかりと</E9>こねる。

Step 2. Identify triggers (trigger entity, type).
- (E5, CHANGE)
- (E6, CHANGE)
- (E8, CHANGE)

Step 3. Identify argument and ECR relations (entity1, relation, entity2).
- (E5, Object, E1)
- (E5, Instrument, E2)
- (E5, Temperature, E3)
- (E5, Manner, E4)
- (E6, Manner, E9)
- (E6, ECR, E5)
- (E8, Manner, E7)
- (E8, ECR, E6)
```

Figure 16: The few-shot demonstration for the LLM. We show one example here.

```
Labeling Target:
user
技能者：今日はフライドポテトの作り方についてお話します。
インタビュアー：さいの目に切ったジャガイモを直接揚げていくんですか？
技能者：実はそうではありません。先に70〜80℃でじっくりと揚げて水分を飛ばすのがポイントです。
インタビュアー：その後はどうすればいいんですか？
技能者：一度取り出して油を切ったら、今度は180℃でサクッと揚げます。これにより外側はパリッ
と、中はホクホクのフライドポテトが完成します。
======================
Model output:
assistant
Step 1. Mark up the text with entities.
技能者：今日は<E1>フライドポテト</E1>の作り方についてお話します。
インタビュアー：<E2>さいの目に切ったジャガイモ</E2>を直接<E3>揚げ</E3>ていくんですか？
技能者：実はそうではありません。先に<E4>70～80℃</E4>でじっくりと<E5>揚げ</E5>て水分を
飛ばすのがポイントです。
インタビュアー：その後はどうすればいいんですか？
技能者：<E6>一度</E6><E7>取り出し</E7>て油を切ったら、今度は<E8>180℃</E8>でサクッと
<E9>揚げ</E9>ます。これにより外側はパリッと、中はホクホクのフライドポテトが完成します。

Step 2. Identify triggers (trigger entity, type).
- (E3, BAKE_FRY)
- (E5, BAKE_FRY)
- (E7, REMOVE)
- (E9, BAKE_FRY)

Step 3. Identify argument and ECR relations (entity1, relation, entity2).
- (E3, Object, E2)
- (E5, Temperature, E4)
- (E5, ECR, E3)
- (E7, Time, E6)
- (E9, Temperature, E8)
- (E9, Product, E7)
```

Figure 17: The input labeling target and an actual labeling output from the LLM. This is an example from the Pseudo-Pseudo data.