Large Language Models Excel at Zero-Shot Conversation Disentanglement

Anonymous ACL submission

Abstract

Disentangling overlapping conversations in multi-party communication is a foundational challenge in natural language processing. Existing state-of-the-art approaches leverage encoder-based language models, often requiring extensive training data and complex feature engineering. In this work, we explore the capabilities of large language models (LLMs) in conversation disentanglement using zero-shot prompting. We propose two simple, principled prompting schemes for conversation disentanglement, along with a self-critic technique for refining results. Testing on the Ubuntu IRC and Movie Dialogue datasets, our methods surpass previous state-of-the-art performance without requiring model fine-tuning. Comparative analysis with human annotators suggests that LLMs perform comparably to humans, but further work is required to uniformly outperform the median annotator on all metrics.

1 Introduction

Complex multi-party and multi-conversation exchanges, such as in online chat rooms or overlapped audio conversations, present a challenging domain for knowledge extraction and modeling tasks that are designed for single dialogues. Conversation disentanglement aims to separate these overlapping conversations into individual conversation threads so that downstream tasks such as information extraction or summarization can then be carried out more effectively. Previous model-based approaches to this task have focused on autoencoding pre-trained language models such as BERT along with hand-crafted features based on discourse structure and pragmatic theory (see Gu et al. (2022) for a detailed survey of research in multi-party conversations). The current work represents, to our knowledge, the first application of enterprise-level autoregressive language models such as GPT-4o to the task of conversation disentanglement. We demonstrate that this family of models is able to out-perform the previous state-of-the art with simple zero-shot prompting strategies, removing the need for fine-tuning on costly annotated data.

2 Related Work

Conversation Disentanglement. Recent model-based approaches to the conversation disentanglement task can largely be organized into two high-level camps: two-step methods that aim to predict utterance reply-to relationships and then use this information to predict shared conversation membership (Elsner and Charniak, 2008; Kummerfeld et al., 2019) and end-to-end methods that aim to predict conversation membership directly in a single inference pass (e.g., (Liu et al., 2020)). Specific architectural choices vary widely, including pre-trained encoder-only language models such as BERT (Zhu et al., 2021; Chang et al., 2023), bi-directional LSTMs to model dependency relations between utterances (Yu and Joty, 2020; Li et al., 2022; Huang et al., 2022), as well as the incorporation of hand-crafted heuristics grounded in pragmatic theory and discourse structure (Kummerfeld et al., 2019; Ma et al., 2022).

Given the limited amount of training data available and the costly nature of training high-quality annotators on this task (Gouravajhala et al., 2023), recent approaches aim to develop unsupervised or self-supervised methods. Liu et al. (2021) exploit the hierarchical nature of the data and task, developing an unsupervised co-training method in which a message-pair classifier and session-level embedding model are jointly trained, leading to increased performance on the disentanglement task. Huang et al. (2022) take a similar hierarchical approach to the task, developing an unsupervised training approach built on bi-contrastive learning to jointly optimize across utterance-level and session-level representations.

Large Language Models. Advancements in large language models (LLMs) have revolutionized the field of natural language processing (NLP), with impressive performance on core NLP tasks such as text summarization, machine translation, sentiment analysis, and named-entity recognition (Yang et al., 2024; Zhao et al., 2024). Recent frontier models are typically decoder-only transformer architectures trained for next-token text prediction on vast and diverse datasets. This training on the structural properties of language leads to impressive emergent properties that enable remarkable performance on a wide variety of tasks and domains.

To our knowledge, this family of LLMs has not yet been applied to the conversation disentanglement task. However, we have strong reason to believe that these models will perform well on this task without significant fine-tuning. Specifically, the conversation disentanglement task relies on knowledge of discourse structure, topic identification and tracking, as well as understanding the pragmatic principles of text-based interactions. Because these higher-order structural characteristics are critical to the task, we believe conversation disentanglement is a productive domain to benchmark the ability of LLMs against previous generations of pre-trained language models and a valuable addition to the suite of more commonly evaluated NLP tasks. The current work addresses this research gap and directly explores the potential of LLMs for disentanglement task and provides a new benchmark for performance on this key NLP task.

3 Methods

3.1 Prompting Schemes for Disentanglement

To evaluate the baseline performance of large language models (LLMs) on the disentanglement task, we explore two zero-shot prompting-based approaches to disentangle overlapping conversations within chat logs. Both methods adopt a turn-based, iterative processing strategy, handling one utterance at a time. We deliberately use simple, principled prompting schemes to highlight the general ability of LLMs on this task rather than tailoring a specific method for maximum performance. Below, we provide comprehensive descriptions and corresponding high-level pseudocode for each method. Prompt text is provided in the Appendix.

3.1.1 Best Response Clustering

Our Best Response approach iterates through the utterances sequentially and uses the LLM to identify the most appropriate prior utterance that a new utterance could respond to. This creates a disconnected directed graph over the utterances. If the LLM determines that the new utterance is not a response to any prior utterance, we treat the new utterance as the start of a new cluster. After constructing the graph, we remove the response context and treat each cluster as a distinct group.

Algorithm 1 Best Response Clustering

```
for each utterance in session do
    Construct prompt with all prior utterances
    Send prompt to LLM and receive response
    Create utterance node in graph
    Create edge from node to parent utterance
end for
Construct clusters from graph
return clusters
```

3.1.2 Direct Assignment Clustering

Our Direct Assignment method provides the LLM with the current state of conversation clusters and tasks it with assigning the newest utterance to the most appropriate existing cluster or creating a new one. This approach differs from Best Response because it directly tasks the LLM with cluster assignment and provides the LLM with the full state of the clusters at each step instead of constructing a graph and determining the clusters after the fact.

Algorithm 2 Direct Assignment Clustering

```
for each utterance in session do
    Construct prompt with current clusters
    Send prompt to LLM and receive response
    Assign utterance to cluster
end for
return clusters
```

3.1.3 Self-Critic with Naive Chain-of-Thought

In addition to our base methods, we experiment with a post-processing self-critic approach that refines an existing set of conversation assignments from a session. We frame this approach as an agent problem where the LLM is deployed within an EVALUATE-ACT loop with three possible actions each turn:

1. **Reassign** an utterance from one conversation to another.
2. **Create** a new conversation and move an utterance to the new conversation.
3. **Finish** the refinement process.

During each turn, the LLM agent is prompted to "think step-by-step" about what should be changed about the current assignment state, based on previous zero-shot chain-of-thought work (Kojima et al., 2023). Then, its output is fed into a Structured Output prompt that returns a function call to one of the three actions. We apply the self-critic process to the conversation assignments of both our Best Response and Direct Assignment methods using the same language model that generated the original results.

3.2 Implementation Details

All of our prompting schemes are implemented in Python using OpenAI's `chat.completions` API (OpenAI, 2024) for interacting with the LLM. For Best Response and Direct Assignment, we use the Structured Outputs feature to ensure the model only returns the output representing either its best response prediction or cluster prediction. Importantly, this means the LLM cannot use any extra tokens to reason about the problem for these methods. For each of our experiments, we test both GPT-4o-2024-08-06 (henceforth "GPT-4o") and GPT-4o-mini (as of November 24, 2024). We set the temperature to zero for replicability.

3.3 Data

We test our methods on the two main datasets commonly used in conversation disentanglement: the **Movie Dialogue** dataset (Liu et al., 2021) and the **Ubuntu IRC** dataset (Kummerfeld et al., 2019).

The Movie Dialogue dataset contains sessions with randomly interleaved scripts from 869 movies. Each session contains dialogue from between 2 and 6 different scripts. Because the Movie Dialogue sessions were synthetically interleaved, they likely do not perfectly align with the structural characteristics of naturally co-occurring conversations. We find that each Movie Dialogue session contains between 10 and 40 lines of dialogue.

The Ubuntu IRC dataset contains sessions sampled from the Ubuntu IRC technical support chatroom, which were then hand-labeled to identify the conversations therein. Each session contains 250 or 500 chat messages, but following prior work, we separate these into chunks of exactly 50 messages. Additionally, some system messages (e.g., a message stating that a user has joined the server) are labeled as each belonging to a unique cluster. Following previous work, we keep these messages in our evaluation data, but our prompts must specify how these messages should be treated.

3.4 Evaluation Metrics

Following (Huang et al., 2022), we evaluate against Normalized Mutual Information (NMI), Adjusted Rand Score (ARI), and Shen-F. NMI and ARI are common clustering metrics that have implementations in the `scikit-learn` library. Shen-F is a variant of F1 score unique to conversation disentanglement and originally proposed by Shen et al. (2006); we implement this metric ourselves.

4 Experiments

4.1 Ubuntu IRC

We present the results for the test partition of the Ubuntu IRC dataset in Table 1. We find that the Direct Assignment method performs competitively against previous state-of-the-art with both GPT-4o and GPT-4o-mini, with GPT-4o exceeding previous state-of-the-art performance on all three metrics.

Ubuntu IRC			
Method	**NMI**	**ARI**	**Shen-F**
TRANSITION-BASED	0.626	0.206	0.497
BI-CL	0.624	0.360	0.707
PTR-NET	-	0.801	-
Best Response (4o)	0.859	0.665	0.814
with Self-Critic	0.872	0.711	0.837
Best Response (4o-mini)	0.368	0.156	0.479
with Self-Critic	0.381	0.163	0.483
Direct Assignment (4o)	**0.912**	**0.823**	**0.912**
with Self-Critic	**0.919**	**0.836**	**0.916**
Direct Assignment (4o-mini)	0.889	0.764	0.884
with Self-Critic	0.885	0.759	0.880

Table 1: Clustering Performance Results on Ubuntu IRC with NMI, ARI, and Shen-F. TRANSITION-BASED results from Liu et al. (2020). BI-CL results from Huang et al. (2022). PTR-NET results from Yu and Joty (2020).

The response prompt scheme significantly influences LLM performance on this task. While we are able to exceed previous state-of-the-art benchmarks with the Direct Assignment prompting scheme, the Best Response method falls short, especially for GPT-4o-mini, which underperforms all prior results we include.

Our self-critic technique also increases performance across all metrics except for Direct Assignment with GPT-4o-mini. However, the effect size is quite small and does not conclusively demonstrate the utility of our self-critic method.

4.2 Movie Dialogue

As in the Ubuntu IRC results, our Direct Assignment method with both GPT-4o and GPT-4o-mini outperforms the previous state-of-the-art by a wide margin, as presented in Table 2. In line with previous literature, we find that our methods underperform on the Movie Dialogue dataset compared to the Ubuntu IRC dataset. We hypothesize this is largely due to (1) modality differences, where movie scenes are made more ambiguous in text via the removal of disambiguating visual/scene information, and relatedly, (2) lack of specific discourse structures present in naturally-interleaved conversations like explicit addressee mention and related strategies taken by speakers who know they are participating in a forum where multiple conversations are taking place. Overall, we significantly advance the state-of-the-art performance on this task and dataset, with the best performing configuration (Direct Assignment GPT-4o with self-critic) demonstrating a relative ARI increase of +107% (0.382 → 0.793) over the previous best approach.

Movie Dialogue			
Method	**NMI**	**ARI**	**Shen-F**
TRANSITION-BASED	0.358	0.255	0.650
BI-CL	0.575	0.382	0.747
Best Response (4o)	0.292	0.222	0.653
with Self-Critic	0.665	0.528	0.791
Best Response (4o-mini)	0.154	0.077	0.581
with Self-Critic	0.166	0.077	0.586
Direct Assignment (4o)	**0.813**	**0.747**	**0.892**
with Self-Critic	**0.841**	**0.793**	**0.911**
Direct Assignment (4o-mini)	0.743	0.615	0.802
with Self-Critic	0.751	0.636	0.819

Table 2: Clustering Performance Results on Movie Dialogue. TRANSITION-BASED results from Liu et al. (2020). BI-CL results from Huang et al. (2022).

4.3 Human Baseline

To compare the performance of our methods to a human baseline, we recruit 5 volunteers to hand-label a random subset of 5 sessions from the Ubuntu IRC test set and 5 sessions from the Movie Dialogue test set. Annotators had no previous experience with the datasets or project but were research scientists in related fields from our informal social networks. Participants are provided the same instructions as the language models, except instead of iterating through the utterances sequentially, we allow them to view the whole context of the conversation.

We find that our disentanglement methods are out-performed by humans across most metrics on our sample of 5 sessions from Ubuntu IRC and 5 sessions from Movie Dialogue. This performance gap is greater when considering median performance due to an outlier labeler. Moreover, humans do not perform as well on the Movie dataset compared to Ubuntu IRC, which is in line with our LLM findings. Overall, the LLM method performs well but does not exceed human-level performance.

Ubuntu IRC			
Method	**NMI**	**ARI**	**Shen-F**
Direct Assignment (4o)	0.914	0.739	0.873
with Self-Critic	0.916	0.742	0.875
Direct Assignment (4o-mini)	0.918	0.788	0.890
with Self-Critic	0.909	0.777	0.937
Human Baseline (mean)	0.886	0.803	0.875
Human Baseline (median)	**0.969**	**0.940**	**0.948**
Movie Dialogue			
Method	**NMI**	**ARI**	**Shen-F**
Direct Assignment (4o)	0.777	0.737	0.892
with Self-Critic	0.809	0.785	**0.894**
Direct Assignment (4o-mini)	0.734	0.618	0.796
with Self-Critic	0.706	0.599	0.790
Human Baseline (mean)	0.812	0.717	0.806
Human Baseline (median)	**0.869**	**0.819**	0.867

Table 3: Comparison to Human Baseline on Random subsets of Ubuntu IRC and Movie Dialogue.

5 Conclusion

In this work, we present the first exploration of frontier large language models on the task of conversation disentanglement. We develop an iterative, zero-shot prompting scheme for GPT-4o that is able to exceed previous state-of-the-art performance obtained via earlier pre-trained language models such as BERT. Additionally, we demonstrate that implementing an iterative self-critic procedure provides modest performance gains for most LLM conditions. The task of conversation disentanglement leverages both local and global information, requiring the synthesis of informative cues at the utterance level and at the larger level of discourse structure. This work presents evidence that LLMs are well-positioned to excel at this task and represents a baseline for future development and research.

6 Limitations

One potential limitation of this research is our reliance on restrictive structured output schema and standard decoding methods. Strict constraints on structured output, such as output schema restrictions, can cause a significant decrease in LLM performance on reasoning tasks (Tam et al., 2024). This performance decrease can be avoided by generating responses on reasoning-based tasks first in natural language and then casting to a structured output via deterministic methods or an oracle model.

Another potential concern is data memorization of the test materials resulting in artificially inflated performance that does not transfer to unseen or future data. Both GPT-4o and GPT-4o-mini have reported knowledge cutoffs of October 2023 (OpenAI, 2024) and may very well include elements of the Movie Dialogue and Ubuntu IRC testsets in their training data. This concern can be addressed in future work by benchmarking performance on novel, unseen datasets or synthetic data.

References

Kent Chang, Danica Chen, and David Bamman. 2023. Dramatic conversation disentanglement. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4020–4046, Toronto, Canada. Association for Computational Linguistics.

Micha Elsner and Eugene Charniak. 2008. You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of ACL-08: HLT*, pages 834–842, Columbus, Ohio. Association for Computational Linguistics.

Sai R. Gouravajhala, Andrew M. Vernier, Yiming Shi, Zihan Li, Mark S. Ackerman, and Jonathan K. Kummerfeld. 2023. Chat disentanglement: Data for new domains and methods for more accurate annotation. In *Proceedings of the 21st Annual Workshop of the Australasian Language Technology Association*, pages 112–117, Melbourne, Australia. Association for Computational Linguistics.

Jia-Chen Gu, Chongyang Tao, and Zhen-Hua Ling. 2022. Who says what to whom: A survey of multi-party conversations. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5486–5493. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Chengyu Huang, Zheng Zhang, Hao Fei, and Lizi Liao. 2022. Conversation disentanglement with bi-level contrastive learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2985–2996, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

Jonathan K. Kummerfeld, Sai R. Gouravajhala, Joseph J. Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros C Polymenakos, and Walter Lasecki. 2019. A large-scale corpus for conversation disentanglement. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3846–3856, Florence, Italy. Association for Computational Linguistics.

Tianda Li, Jia-Chen Gu, Zhen-Hua Ling, and Quan Liu. 2022. Conversation- and tree-structure losses for dialogue disentanglement. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 54–64, Dublin, Ireland. Association for Computational Linguistics.

Hui Liu, Zhan Shi, Jia-Chen Gu, Quan Liu, Si Wei, and Xiaodan Zhu. 2020. End-to-end transition-based online dialogue disentanglement. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3868–3874. International Joint Conferences on Artificial Intelligence Organization. Main track.

Hui Liu, Zhan Shi, and Xiaodan Zhu. 2021. Unsupervised conversation disentanglement through co-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2345–2356, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xinbei Ma, Zhuosheng Zhang, and Hai Zhao. 2022. Structural characterization for dialogue disentanglement. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 285–297, Dublin, Ireland. Association for Computational Linguistics.

OpenAI. 2024. API Pricing. `https://openai.com/api/pricing/`. Accessed: 2024-11-27.

OpenAI. 2024. Chatgpt: Language model by openai. `https://platform.openai.com/docs/api-reference/chat/create`. Accessed: 2024-12-01.

Dou Shen, Qiang Yang, Jian-Tao Sun, and Zheng Chen. 2006. Thread detection in dynamic text message streams. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, page 35–42, New York, NY, USA. Association for Computing Machinery.

Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. 2024. Let me speak freely? a study on the impact of format restrictions on large language model performance. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1218–1236, Miami, Florida, US. Association for Computational Linguistics.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Trans. Knowl. Discov. Data*, 18(6).

Tao Yu and Shafiq Joty. 2020. Online conversation disentanglement with pointer networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6321–6330, Online. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2024. A survey of large language models. *Preprint*, arXiv:2303.18223.

Rongxin Zhu, Jey Han Lau, and Jianzhong Qi. 2021. Findings on conversation disentanglement. In *Proceedings of the 19th Annual Workshop of the Australasian Language Technology Association*, pages 1–11, Online. Australasian Language Technology Association.

Appendix

Prompt: Best Response for Ubuntu IRC

```
You will be provided a chat log where
each message has a unique ID, as well as
the next chat message in the sequence.
Respond with a JSON object that contains
the ID of the message the last chat
message is responding to/following up
to. Alternatively, if the last chat
message seems to be starting its own
conversation, respond with the chat
message's ID. If the last chat message
is just a system message, respond with
the chat message's ID.

Example: {"response_to": 1213}
Example: {"response_to": 6513}
Example: {"response_to": 2439}
```

Prompt: Best Response for Movie Dialogue

```
You will be provided a dialogue where
each message has a unique ID, as well as
the next utterance in the sequence.
Respond with a JSON object that contains
the ID of the message the last line of
dialogue is responding to/following up
to. Alternatively, if the last utterance
seems to be starting its own
conversation, respond with the
utterance's ID.

Example: {"response_to": 12}
Example: {"response_to": 651}
Example: {"response_to": 2439}
```

Prompt: Direct Assignment for Ubuntu IRC

```
You will be provided a set of
conversations extracted from a chat log,
as well as the next chat message in the
sequence. Respond with a JSON object
that contains the conversation ID of
the message the last chat message is
responding to/following up to.
Alternatively, if the last chat message
seems to be starting its own
conversation, respond with
{"conversation_id": 0}. If the last chat
message is just a system message,
respond with {"conversation_id": 0}.

Examples:
{"conversation_id": 0}
{"conversation_id": 11}
{"conversation_id": 7}
```

Prompt: Direct Assignment for Movie Dialogue

```
You will be provided a dialogue with
lines from entangled movie scripts. Your
goal is to assign each line to a cluster
representing which movie script it
belongs to. Respond with a JSON object
that contains the conversation ID of the
movie script it seems the last line of
dialogue belongs to. Alternatively, if
the last chat message seems to be
starting its own conversation, respond
with {"conversation_id": 0}.

Examples:
{"conversation_id": 0}
{"conversation_id": 11}
{"conversation_id": 7}
```

Prompt: Self-Critic Examine for Ubuntu IRC

```
Another agent has tried to disentangle
the conversations in an Ubuntu IRC chat
log. Your task is to decide which action
to take to make any necessary fixes.
These are the actions available to you:

`assign_utterance(utterance_id: int,
new_cluster_id: int)`
Given an utterance's ID and a cluster
ID, move the utterance to that cluster.

`create_conversation(utterance_id: int)`
Create a new conversation populated by
the specified utterance.

`finish()`
Finish the editing process. Run this
once you are content with the results.

- System messages are treated as
separate, unique conversations.

Think step-by-step to determine what the
next action should be. Make your final
decision clear at the end so that the
assigner can follow your instruction.
The final decision should be a single
action rather than multiple.
```

Prompt: Self-Critic Examine for Movie Dialogue

```
Another agent has tried to disentangle
the movie scripts ("conversations") in a
jumbled script. Your task is to decide
which action to take to make any
necessary fixes.
These are the actions available to you:

`assign_utterance(utterance_id: int,
new_cluster_id: int)`
Given an utterance's ID and a cluster
ID, move the utterance to that cluster.

`create_conversation(utterance_id: int)`
Create a new conversation populated by
the specified utterance.

`finish()`
Finish the editing process. Run this
once you are content with the results.

- An utterance belongs to a conversation
when it seems to potentially come from
the same script as other utterances in
that conversation.
- "Conversation" refers to a distinct
movie script. If two utterances are from
the same script, they should be in the
same conversation.

Think step-by-step to determine what the
next action should be. Make your final
decision clear at the end so that the
assigner can follow your instruction.
The final decision should be a single
action rather than multiple.
```

Prompt: Self-Critic Action

```
You will be provided a set of
conversations extracted from a chat log,
the next chat message in the sequence,
and an instruction on what action to
take.

`assign_utterance(utterance_id: int,
new_cluster_id: int)`
Given an utterance's ID and a cluster
ID, move the utterance to that cluster.

`create_conversation(utterance_id: int)`
Create a new conversation populated by
the specified utterance. new_cluster_id
should be set to 0.

`finish()`
Finish the editing process. Run this
once you are content with the results.
utterance_id and new_cluster_id should
be set to 0.

Your response should be a JSON object
with the following keys:

- `action`: The action to take. One of
assign_utterance, create_conversation,
or get_next_utterance.
- `utterance_id`: The ID of the
utterance to assign to a cluster.
- `cluster_id`: The ID of the cluster to
assign the utterance to.

You should only respond with a JSON
object and nothing else. Your response
should formalize what the instruction
says, not what you think the best option
is.
```