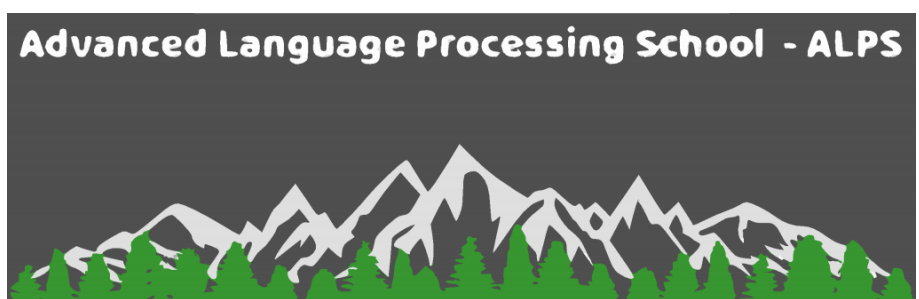
ELITR-Agent-Bench: An Agent Benchmark for Meeting Transcripts

Mohamed Salim Aissi, Song Duong, Florian Le Bronnec, Philippe Martin, Laurent Besacier

April 9, 2025



Extracting tools from audio transcripts

Audio transcripts

- **What we have:** Very long audio transcripts ($\approx$ 1000 lines).
- In these transcripts a lot of **actions** are discussed:
  - Send emails,
  - Adding events in calendar,
  - Sharing a file
  - etc.

ELITR-Bench

Dataset: ELITR-Bench: A Meeting Assistant Benchmark for Long-Context Language Models (Thonet et al., COLING 2025).

> ***PERSON9:*** *So that at least I should be able to prepare the call link for the call for all the participants so that at least we spend this kind of task. Just pretty—*
>
> ***PERSON10:*** *I think [PERSON7] is responsible, so we actually need to book a Zoom meeting slot. And I think [PERSON7] is responsible for that.*
>
> ***PERSON9:*** *Ok.*
>
> ⟨*censored*⟩
> ⟨*laugh*⟩ *Responsibility.*
> ***PERSON10:*** ⟨*censored*⟩
> ***PERSON9:*** *Ok.*

Adding tools

- **Goal:** Extract the actions that can be executed by a **LLM Agent**.
- Enrich the dataset with tool use, to benchmark tool use abilities of LLMs.

PERSON9: *So that at least I should be able to prepare the call link for the call for all the participants so that at least we spend this kind of task. Just pretty—*

PERSON10: *I think [PERSON7] is responsible, so we actually need to book a Zoom meeting slot. And I think [PERSON7] is responsible for that.*

PERSON9: *Ok.*

⟨*censored*⟩
⟨*laugh*⟩ *Responsibility.*
PERSON10: ⟨*censored*⟩
PERSON9: *Ok.*

Task extraction

We will focus on the task extraction problem. **Objective:** Extract actionable, non-technical follow-up actions from a meeting transcript.

Examples of actions:

- Drafting or sending emails
- Organizing meetings (booking platforms/resources)
- Confirming availability or time zones
- Assigning actions to relevant people with deadlines

- **Assigned to:** `PERSON7`
- **Description:** Send a Zoom Meeting invitation.
- **Supporting Evidence:**
  PERSON10: *I think [PERSON7] is responsible, so we actually need to book a Zoom meeting slot. And I think [PERSON7] is responsible for that.*

Why is it a difficult problem?

Using a strong LLM

Goal: Do this data-augmentation with a **long-context LLM**.

Why is it a difficult problem?

Using a strong LLM

Goal: Do this data-augmentation with a **long-context LLM**.

A difficult problem

- Transcripts are **very long**.
- Long-contexts LLMs are actually not that performing.
- Extracting actions is a mix of **extractive and abstractive** tasks.
- **No supervision**.
- We need some **guarantees** that the extraction has been well performed.

Why is it a difficult problem?

Using a strong LLM

Goal: Do this data-augmentation with a **long-context LLM**.

A difficult problem

- Transcripts are **very long**.
- Long-contexts LLMs are actually not that performing.
- Extracting actions is a mix of **extractive and abstractive** tasks.
- **No supervision**.
- We need some **guarantees** that the extraction has been well performed.

Conclusion:

- We expect a lot of incorrect behavior, even from strong LLMs.
- A lot of **prompt-engineering**.

Naive prompting (GPT-4o)

Naive prompt: *Extract all non-technical, actionable, and specific follow-up actions from this meeting transcript that can be handled by an LLM Agent.*

- Common error: Misclassifying **DONE** actions as **TODO**.
- **Example:**
  - **Task:** Contact [PERSON2]
  - **Assigned:** [PERSON10]
  - **Excerpt:** "I have sent an e-mail to [PERSON2]..."
- Filtering error: Keeping **technical** actions that can hardly be handled by an LLM Agent.
- **Example:**
  - **Task:** Select and test suitable demo video
  - **Assigned:** [PERSON5] & [PERSON11]
  - **Excerpt:** "Please try to identify the good video file suitable for this kind of demo."

Naive prompting: analysis

Naive prompt: *Extract all non-technical, actionable, and specific follow-up actions from this meeting transcript that can be handled by an LLM Agent.*

- Common error: Misclassifying **DONE** actions as **TODO**.
- **Example:**
  - **Task:** Contact [PERSON2]
  - **Assigned:** [PERSON10]
  - **Excerpt:** "I have sent an e-mail to [PERSON2]..."
- Filtering error: Keeping **technical** actions that can hardly be handled by an LLM Agent.
- **Example:**
  - **Task:** Select and test suitable demo video
  - **Assigned:** [PERSON5] & [PERSON11]
  - **Excerpt:** "Please try to identify the good video file suitable for this kind of demo."

✓ What works

- Most of the extracted actions are correct.
- Models seems quite exhaustive.
- Asking the model to output supporting excerpts helps in faithfulness.

✗ What doesn't work

- Smaller LLMs like Mistral were completely off-course.
- Larger LLMs are not yet precise enough.
- Though, detecting errors is quite easy.

Proposed approach

Use a **pipeline** of prompts, with successive refinement / filtering.

1. **Extract as most actions as possible** (simply asking the model to extract all follow-up actions, without much specifications).
2. **Filter the actions** that are actually **TODO**.
3. **Select the actions that are actually actionnable**, by providing a finer description of what should be extracted.

Quality guarantees

What we are interested in

- **Recall:** All mentioned actions should be retrieved.
- **Precision:** All extracted follow-up actions should be:
  - Non-technical.
  - Easily handled by an LLM Agent.
  - Supported by the input transcript.

Chunked self-assessement

Fine-grained evaluation

1. Iterate over the transcript's chunks.
2. For each chunk, ask the LLM to label which action is supported by the chunk.
3. Count how many actions have been labeled as "supported" over the total.

- **Easy individual evaluations**.
- **Mitigate long-contexts** processing inaccuracies.
- Serves as a **good proxy for quality**.

Evaluation metrics

Automatic metrics

- Number of actions.
- Number of assigned persons.
- Does the excerpt matches the input?

LLM metrics

- Chunk assessement.

Multi-model assessement

Inter-model agreement

Idea: If **several strong models agree** on the extracted actions, then the extraction has higher chances to be correct.

- Gather the results from several models.
- Find similar actions by cross-referencing the excerpts, the assigned person.
- Compute the overlap between the extractions.

Results

Actions extraction

- **LangChain Pipeline**, using either open-models or proprietary ones.
- **Structured outputs** to ensure easy parsing and analysis.

✓ Consistent formatting.

✓ Easy parsing.

Evaluation

Quality evaluation

- **LangChain Pipeline**.
- **Structured outputs**.
- **Task 1**
  - **Assigned to:** PERSON6
  - **Description:** Send an email to [PERSON10] with the meeting address
  - **Excerpt:** "So we are expecting [PERSON10] today?"
  - **Supported by chunk:** **No**
- **Task 2**
  - **Assigned to:** PERSON3
  - **Description:** Email [PERSON10] to confirm the meeting address
  - **Excerpt:** "Yes, he has just written an email asked for the address now are we"
  - **Supported by chunk:** **Yes**

✓ Easy task.
✓ Easy parsing.
✓ Consistent formatting.

Conclusion / Future Work

Improve actions extraction

- Define exact **actions/tools list**.
- **Improve the prompts** based on a complete evaluation.
- Explore **"model ensembling"**.

Evaluation

- Provide a limited set of **human annotations** (for Recall assessement).

Conclusion / Future Work

Improve actions extraction

- Define exact **actions/tools list**.
- **Improve the prompts** based on a complete evaluation.
- Explore **"model ensembling"**.

Evaluation

- Provide a limited set of **human annotations** (for Recall assessement).

Thank you!