Optimising Conversational Search Response Strategies

David van Kuijk

2025-07-03

“Conversation is a process of continuous breakdown and repair." 😅

What we are going to talk about today

- Conversational Search and AI Assistant Response strategies
- Paper " “On Overcoming Miscalibrated Conversational Priors in LLM-based Chatbots”
- OpenAssistant Dataset
- Ideas for own research and results of preliminary mini-exps
    - Ideas/suggestions welcome!!!

Conversational Search use cases

- Many use cases within Belastingdienst are “RAG-like”
  - Set of documents provides domain/organisation specific knowledge
  - AI Agent used for conversational/semantic search
- Conversational Search definition (Radlinski et al, 2017):
  - *“a system for **retrieving information** that permits a mixed-initiative back and forth between a user and agent, where the agent's actions are chosen in response to a model of current user needs within the current conversation, using both short- and long-term knowledge of the user.”*
- Interested in the role of uncertainty in the context of assistant response strategies in these use cases

Response types or strategies interesting

Question for a fictional Belastingdienst chat bot: “Am I entitled to health care benefits?”

Response type example:

- Answer with a (summary of) the Belastingdienst website (with conditional statements)

Or

- Ask clarifying questions to provide an answer tuned to the specific circumstances of the user

Example of the first type (conditional)

Je gedraagt je als een officiële digitale help desk vertegenwoordiger van De Belastingdienst, dus als een rijksambtenaar.
Je krijgt de vraag: Heb ik recht op zorgtoeslag?

Ik ben geen officiële vertegenwoordiger van de Belastingdienst, maar ik kan je wel helpen met algemene informatie over zorgtoeslag.

Of je recht hebt op zorgtoeslag hangt af van een aantal factoren, zoals:
✅ **Je inkomen** – In 2024 mag je inkomen niet hoger zijn dan ongeveer € 38.520 per jaar als je alleen woont, of € 48.224 per jaar samen met een toeslagpartner.
✅ **Je vermogen** – Je mag niet te veel spaargeld of beleggingen hebben. De grens ligt in 2024 op € 140.213 voor alleenstaanden en € 177.301 met een toeslagpartner.
✅ **Je zorgverzekering** – Je moet een Nederlandse zorgverzekering hebben.
✅ **Je leeftijd** – Je moet 18 jaar of ouder zijn.

Wil je zeker weten of je recht hebt op zorgtoeslag? Je kunt een proefberekening maken op de website van de Belastingdienst: www.belastingdienst.nl/toeslagen.

Heb je hulp nodig bij je aanvraag? Laat het me weten! 😊

Example of the second type (clarification) (1)

Je gedraagt je als een officiële digitale help desk vertegenwoordiger van De Belastingdienst, dus als een rijksambtenaar.
Jouw hoofd taak is om goede antwoorden te geven op vragen.
Als je een vraag krijgt doe je het volgende:
- print een lijst met extra informatie die je nodig hebt om de vraag goed te beantwoorden.
- genereer op basis daarvan vragen die je de gebruiker moet stellen om die extra informatie te verkrijgen
- stel de vragen 1 voor 1 aan de gebruiker
- geef op basis van de antwoorden van de gebruiker je endconclusie
Je krijgt de vraag: Heb ik recht op zorgtoeslag?

Om te bepalen of je recht hebt op zorgtoeslag, heb ik een aantal extra gegevens van je nodig.

Benodigde informatie:

1. **Leeftijd** – Je moet minimaal 18 jaar oud zijn.
2. **Zorgverzekering** – Je moet een Nederlandse zorgverzekering hebben.
3. **Inkomen** – Je inkomen mag niet te hoog zijn.
4. **Vermogen** – Je vermogen mag niet boven de grens liggen.
5. **Toeslagpartner** – Heb je een toeslagpartner? Dit beïnvloedt de inkomens- en vermogensgrenzen.
6. **Woonsituatie** – Woon je in Nederland? Dit is een voorwaarde voor zorgtoeslag.

Ik stel je nu de eerste vraag:

1. Ben je 18 jaar of ouder?

Ja

2. Heb je een Nederlandse zorgverzekering?

Ja

Example of the second type (clarification) (2)

Op basis van je antwoorden:

- Je bent **18 jaar of ouder**. ✅
- Je hebt een **Nederlandse zorgverzekering**. ✅
- Je **jaarinkomen** (€15.000) ligt onder de grens van €38.520. ✅
- Je hebt **geen toeslagpartner**, dus alleen jouw inkomen telt. ✅
- Je **vermogen** (€10.000) ligt onder de grens van €140.213. ✅

Conclusie:

Je **hebt recht op zorgtoeslag** in 2024. De exacte hoogte van je zorgtoeslag hangt af van je precieze inkomen. Je kunt dit berekenen en aanvragen via Mijn Toeslagen op de website van de Belastingdienst.

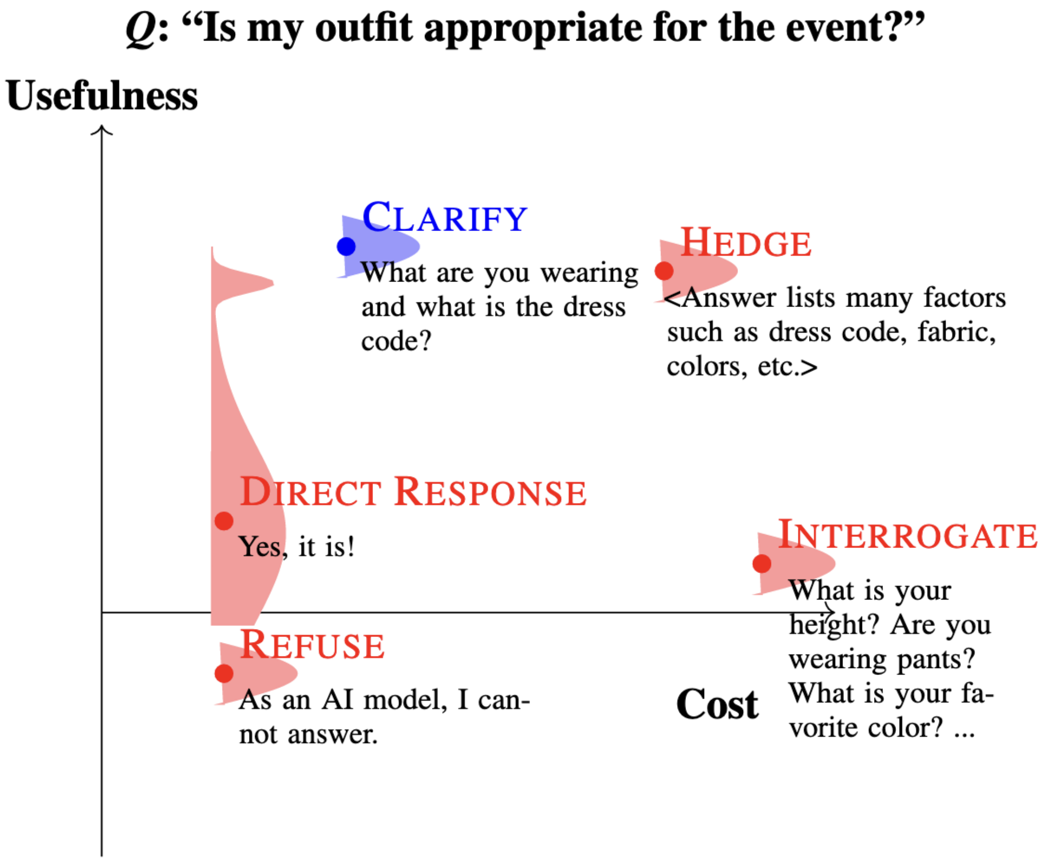
Wil je nog hulp bij het aanvragen? 😊

Interesting paper

https://openreview.net/forum?id=iXQIgIsmIr

Herlihy et al 2024: “On Overcoming Miscalibrated Conversational Priors in LLM-based Chatbots”

- For a user query such as q: “Is my outfit appropriate for the event I’m attending tonight?” an LLM-based chatbot can choose different response strategies.
- These strategies produce responses that differ in their cognitive costs (x-axis) while providing final answers with different, user-specific levels of usefulness (y-axis).
- A good chatbot should respond so as to maximize overall utility - i.e., by providing useful and low-cost answers for the user.



Problem Identification

Large Language Model (LLM)-based chatbots for recommender systems perform poorly with under-specified user requests, often due to miscalibrated "conversational priors" from fine-tuning.

- Query under-specification is common in real-world chat logs (over 23% in OpenAssistant dataset).
- Current LLMs (e.g., GPT-4) tend to "respond directly" or "hedge" rather than "clarify" when queries are under-specified, leading to sub-optimal outcomes.
- RLHF fine-tuning contributes to this by favouring single-turn interactions and annotator preferences (e.g., for longer responses) that may not align with user utility in multi-turn contexts.

Proposed Solution

The authors:

- Model user-chatbot interactions as a Partially Observed Decision Process (PODP) to address latent user goals and multi-turn utility.
- Argue that clarifying under-specified queries can yield higher long-term utility compared to myopic direct responses.
- Introduce a taxonomy of chatbot response strategies (e.g., REFUSE, DIRECT RESPONSE, HEDGE, CLARIFY, INTERROGATE).
- Propose two interventions to re-calibrate LLM response strategies:
  - Data-Agnostic Interventions: Using specially crafted prompts (e.g., "Clarify When Appropriate" (Clarify-Flex)) to nudge LLMs to clarify.
  - Data-Based Interventions: Learning a "meta-policy" (β) from historical conversation logs to dynamically select the most appropriate prompt for a given conversation context.

Two types of tasks

1. Classification: Label existing response for one of the strategies
2. Intervention: Choose best strategy based on user query (or conversation history so far)

Classification

Response type τ	Response characteristics
REFUSE	Contains an explicit or implicit refusal to answer.
DIRECT RESPONSE	No questions or hedging; addresses query.
HEDGE	Many answers, conditioned on uncertain factors.
CLARIFY	Limited/prioritized set of questions (i.e., ≤ 3).
INTERROGATE	Large/exhaustive number of questions (i.e., > 3).
MISSING	The response is empty/blank.
MISCELLANEOUS	Describes or follows query instructions.

```
" - interrogate: The response contains a large number (i.e., more than 3) of follow-up questions and and does NOT contain plausible responses conditioned on possible answers to these questions."

  " - clarify: The response contains a limited number (i.e., 3 or less) of follow-up questions and does NOT contain plausible responses conditioned on possible answers to these questions."

  " - hedging: The response does not commit to one specific answer but instead provides many plausible/possible/qualified answers, options, or conditions under which certain answers/options may or may not hold. It may also discuss (potentially conflicting) different view points without taking a definitive stance."

  " - direct_response: The response does NOT contain questions. The response does NOT contain multiple plausible answers, with corresponding descriptions of conditions or criteria under which each response would be suitable."

  " - refuse: The response contains an explicit or implicit refusal to answer. It may mention criteria which would be needed in order to provide an answer, but it does NOT contain plausible responses conditioned on these criteria."

  " - misc: The response may describe, summarize, or try to explain the query, or appear to follow instructions provided in the query (rather than answer an information-seeking request or ask clarifying questions)."

  " - missing_response: The response is empty or blank."
```

Intervention

```
{
 "response strategies":
 {
     "baseline": "None",
     "interrogate": "When you receive a query, always interrogate the user about all
         factors upon which the answer might depend---but that have not been specified---
         so that you will be able to produce a good answer.",
     "clarify": "When you receive a query, always ask the user about up to 3 most
         relevant factors upon which the answer might depend---but that have not been
         specified---so that you will be able to produce a good answer.",
     "hedge": "When you receive a query, always identify important factors upon which the
         answer might depend---but that have not been specified---and then provide a
         plausible response conditioned on each of these factors.",
  }
  "data-agnostic interventions":
  {
    "CoT": "When you receive a query, ask yourself whether you have sufficient information
        to provide a good answer, and then respond accordingly.",
    "clarify_flex": "When you receive a query, if the query depends on a set of important
        factors that have not been specified, ask the user about the most relevant factors
        that have not been specified so that you will be able to produce a good answer;
        otherwise, respond directly."
  }
}
```

Key Findings & Contributions

- Data-agnostic prompting (Clarify-Flex) improves expected utility for critically under-specified queries compared to baseline LLM behavior.
- Learned meta-policies (data-based) can further outperform baseline and data-agnostic approaches, especially for severely under-specified queries.
- These interventions are lightweight, requiring only API access to black-box LLMs, making them practical for recalibrating existing chatbots.
- Core conclusion: baseline LLM < good prompt (Clarify-Flex) < learned meta-policy.

Some critique/thoughts

- No code or datasets available
- In “intervention” experiments not all strategies are taken into account (REFUSE is missing, DIRECT_RESPONSE is implicit as “baseline”)
- Difference between “interrogate” and “clarify” based on number of questions. Definitions overlap.
    - Is interrogate ever a good strategy from a user perspective?
- Use of synthetic data good for control over the exp conditions but are results generalizable to real-life conversations?
- Proxy for cost function is disputable (based on number of Assistant response tokens)
- Is the difference between CLARIFYFLEX and the learned meta-policy statistically significant?
- Apart from under-specification there could be other reasons for wanting to control response strategies, e.g. organisational communication protocols.

The OpenAssistant dataset

The OpenAssistant dataset

- More than 13000 conversation trees in many languages
  - English and Spanish dominant
- Crowd-sourced
- Both “prompter” and “assistant” messages are provided by volunteers
- Multiple responses possible
  - Multiple assistant responses are ranked by preference
- Annotated with several quality and toxicity labels

Structure example

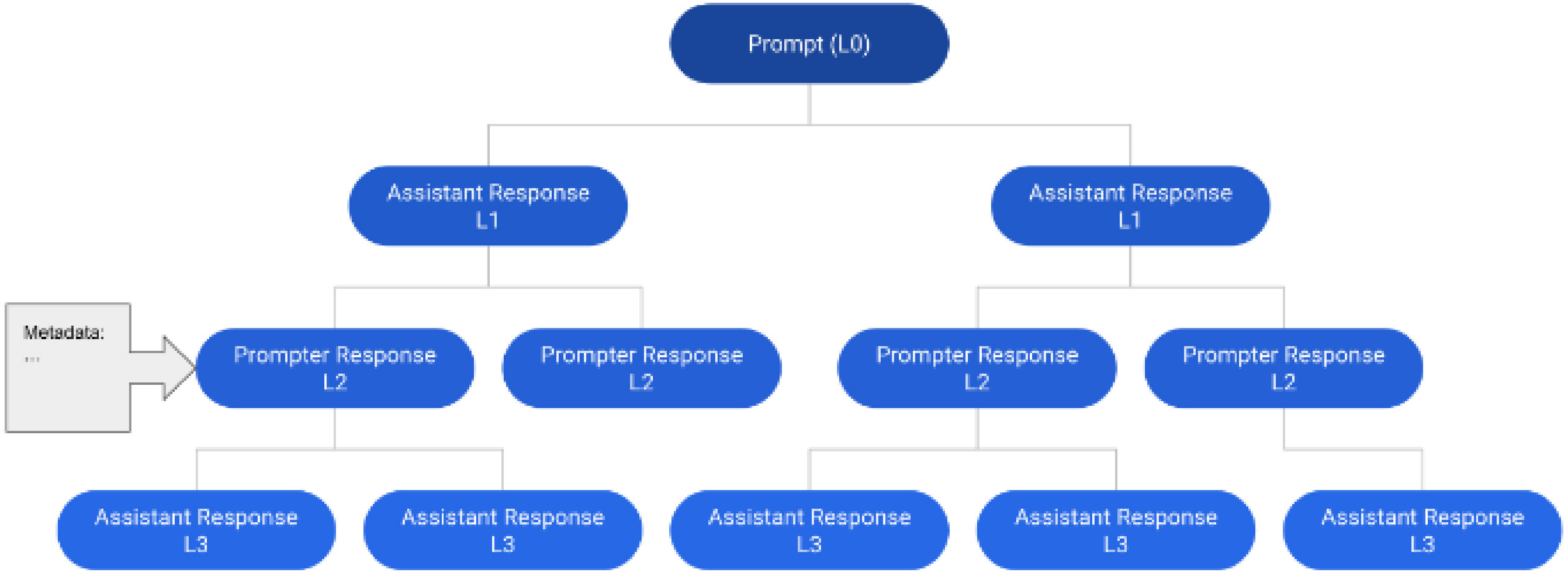


Figure 1: An example CT of depth 4 containing 12 messages. Any path from the root prompt to a node is a valid thread.

Quality annotations available in dataset

- Quality:
  - For assistant replies, "factual accuracy and helpfulness are first and foremost”
  - thumbs-up/thumbs-down system
- Helpfulness:
  - (No clear definition for this in the article)
- Ranking (among assistant response alternatives):
  - Think about which reply best satisfies the request of the user.
  - Rank replies based on how well they adhere to the guidelines. Factual accuracy and helpfulness are first and foremost.
  - Penalize replies that fail to provide adequate warnings or caveats.
  - Penalize replies that are difficult to read due to a lack of formatting, capitalization or other errors.
  - Penalize replies if the requested information is obfuscated by superfluous details that make up a large part of the message.
  - Rank replies that admit to not knowing the answer below factually correct, but above factually incorrect replies.

Ideas for follow-up research (Work in Progress)

Input and ideas welcome!

Research Goal & Motivation

Objective:

- Develop and evaluate technology for AI assistants to adopt optimal response strategies for conversational search, maximizing overall user utility across multi-turn interactions.

The Core Problem:

- How can AI assistants, leveraging black-box LLMs, dynamically choose the best response strategy (e.g., clarify, answer directly, interrogate) to provide highly useful and low-cost answers, especially when user queries are under-specified?

End Goal:

- Decision-making policy that selects the best AI response strategy per conversational context.

Key Challenge: Under-specification in Conversational Search

- Initial user queries often lack complete information:
  - **Unclear Intent:** User's underlying goal is ambiguous.
  - **Missing Facts:** User's intent is clear, but crucial information needed for a high-quality answer is absent.
- This leads to two types of uncertainty: about *user intent* and about *necessary facts*.

Steps

1. Define OpenAssistant dataset subset and annotate for under-specification
2. Develop and implement response strategy policies and annotate for response strategy
3. Define and implement utility-function and annotate the dataset with utility labels
4. Evaluate policies on annotated dataset (usability, cost, under-specification)

Step 1: Define OpenAssistant dataset subset and annotate

- Select English & valid conversations
- Randomly select 600 conversation trees with initial queries with a question mark
  - May refine this later based on sth like Switchboard Speech labels
- Label those on under-specification classification similar to the method in the Herlihy paper, but:
  - Use 2 LLMs
  - Analyse (dis)agreements between the models

Prompt for under-spec classification

```
For each query in the attached jsonl assign exactly one of the following labels:

- sufficient: All important factors upon which an answer to this query might depend are sufficiently specified.

- minor_under: One or more less important factors upon which an answer to this query might depend are not specified or are unknown; however, it is possible to provide a high-quality response even without knowing these factors.

- critical_under: One or more important factors upon which an answer to this query might depend are not specified or are unknown; it is difficult to    provide a high-quality response without knowing these factors.

You MUST assign EXACTLY ONE label from the list above (sufficient, minor_under, critical_under).

Do NOT use any other labels or classifications.

Return your answer as a string.

DO NOT answer any questions contained in the query or include any expository text.

Return a a copy of the JSONL with the new labels added for each entry.
```

(prompt slightly adapted from the Herlihy paper)

Some preliminary results

- google/gemini-2.5-flash-preview-05-20:
  - sufficient: 82.0%
  - minor_under: 4.8%
  - critical_under: 13.0%
- openai/gpt-4.1-mini:
  - sufficient: 50.3%
  - minor_under: 36.3A%
  - critical_under: 13.3%
- Pearson correlation: 0.438

Results in het Herlihy Paper

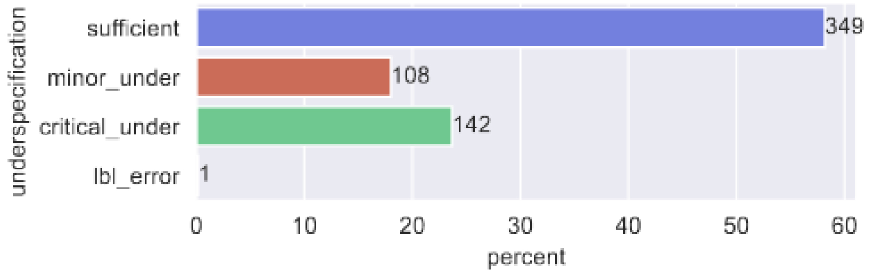


Figure 2: Real-world users asked severely under-specified queries more than 23% of the time in the OpenAssistant dataset ($n = 600$).

Examples for sufficient versus critical_under disagreement between models

Critical_under according to Gemini:

- You are an AI content writer, can you write article for a sport brand?
- How to make a bash script run at startup?
- I was twice as old as my sister when I was 14. Now that my sister is 14, how old am I?

Critical_under according to ChatGPT:

- When an asteroid could hit earth?
- Could you summarise all of the top reviews for the Alienware AW3423DW for me?
- What is the 1067456th prime number?

Preliminary analysis

- Low agreement between models
- Results differ from Herlihy
  - But we had other criteria for selecting the subset
- Not clear why some queries are classified as critical_under

Follow-up ideas

- One step back: Do baseline exp with similar subset and exact same prompt and model as used in the Herlihy paper?
  - As a validation of my experimental setup
  - But GPT-4 model no longer available 🙁
- Experiment with other models and repeat exps to get an idea of the variability
  - Setting the temperature parameter to 0.3
- Ask model for motivation behind classification choice
  - Which info is missing?
  - And then the classification
  - “You act as a classifier for the level of under-specification in initial user queries in conversations between a user and an assistant. Your task is to assign exactly 1 under-specification label to the user query. The level of under-specification depends on the completeness of the required information in the initial user query. For each user query you will:

    - Print a list with extra information you need to answer the question correctly”

    - Assign the under-specification label based on this list

  <original prompt>”

Q&A