

# ECLAIR: Enhanced Clarification for Interactive Responses in an Enterprise AI Assistant

John Murzaku<sup>\*1</sup>, Zifan Liu<sup>2</sup>, Vaishnavi Muppala<sup>2</sup>,  
Md Mehrab Tanjim<sup>3</sup>, Xiang Chen<sup>3</sup>, Yunyao Li<sup>2</sup>

<sup>1</sup>Stony Brook University

<sup>2</sup>Adobe

<sup>3</sup>Adobe Research

jmurzaku@cs.stonybrook.edu, {zifanl, mvaishna, tanjim, xiangche, yunyao}@adobe.com

## Abstract

Large language models (LLMs) have shown remarkable progress in understanding and generating natural language across various applications. However, they often struggle with resolving ambiguities in real-world, enterprise-level interactions, where context and domain-specific knowledge play a crucial role. In this demonstration, we introduce ECLAIR (**E**nhanced **CL**arification for **I**nteractive **R**esponses), a multi-agent framework for interactive disambiguation. ECLAIR enhances ambiguous user query clarification through an interactive process where custom agents are defined, ambiguity reasoning is conducted by the agents, clarification questions are generated, and user feedback is leveraged to refine the final response. When tested on real-world customer data, ECLAIR demonstrates significant improvements in clarification question generation compared to standard few-shot methods.

## Introduction

In human conversation, ambiguities are naturally detected and resolved through contextually appropriate clarification questions, integrating a combination of linguistic and paralinguistic cues. Large language models (LLMs) have made strides in this area, demonstrating some ability to resolve ambiguities through ambiguity detection and clarification question generation (Kuhn, Gal, and Farquhar 2022; Zhang and Choi 2023; Zhang et al. 2024). However, recent LLM-based approaches tend to rely on sequential pipelines where ambiguity is first detected and then a clarification question is generated. These methods treat ambiguity detection as a purely lexical issue, ignoring vital contextual and domain-specific information. This in turn significantly limits their applicability to real-world, enterprise scenarios.

Existing approaches are also limited by their focus on datasets that lack the complexity and nuances of actual industry applications. Many of the datasets are synthetic, failing to capture the range of ambiguities in real-world interactions. This limits their applicability to enterprise systems that require a sophisticated understanding of ambiguity.

To address these challenges, we introduce ECLAIR (**E**nhanced **CL**arification for **I**nteractive **R**esponses), a uni-

fied agentic framework designed for enterprise AI assistants. ECLAIR provides a unified conversational experience on the user side and offers better disambiguation capabilities.

Recent works on ambiguity detection and clarification question generation often use prompt-based LLM approaches, such as zero-shot or few-shot chain-of-thought (CoT) prompting (Kuhn, Gal, and Farquhar 2022; Deng et al. 2023; Zhang et al. 2024). Despite their successes on benchmark datasets, these approaches are inadequate for resolving real-world ambiguities in an enterprise domain setting. Our method aims to resolve these issues and is salient in two key aspects: 1) ECLAIR integrates ambiguity information and domain knowledge from multiple downstream agents, enhancing overall context-awareness in resolving ambiguities. 2) ECLAIR is developed and evaluated using a corpus of real user queries from the Adobe Experience Platform (AEP) AI Assistant (Bhambhri 2024).

## Framework Overview

The ECLAIR architecture (Figure 1) is built as a modularizable and extendable framework: the AI Assistant developers are able to fully define their custom ambiguity detection or grounding information agents. We describe a step-by-step sample interaction and what happens in our framework.

The framework takes a user query as input and generates a set of predefined agents for ambiguity detection. The agents collaboratively reason whether an ambiguity exists in the query. The agents that do detect an ambiguity then get added to a prompt, which then calls a LLM to decide whether clarification is needed or not. In our example, ECLAIR then generates a clarification question with two choices, where the user can then add their feedback.

In our specific use case, we define the following agents: 1) a generic sentence-level ambiguity detector that captures domain-agnostic ambiguities, 2) a product ambiguity detector that decides if the query refers to multiple products in Adobe AEP 3) an entity linking ambiguity detector that identifies text spans that can be linked to multiple entities in the database and 4) a concept graph module that identified tokens that are AEP-specific terminology.

## Demonstration

Our demo video showcases two salient use cases for our enterprise AI assistant on our UI deployment.

<sup>\*</sup>Work done while interning at Adobe.

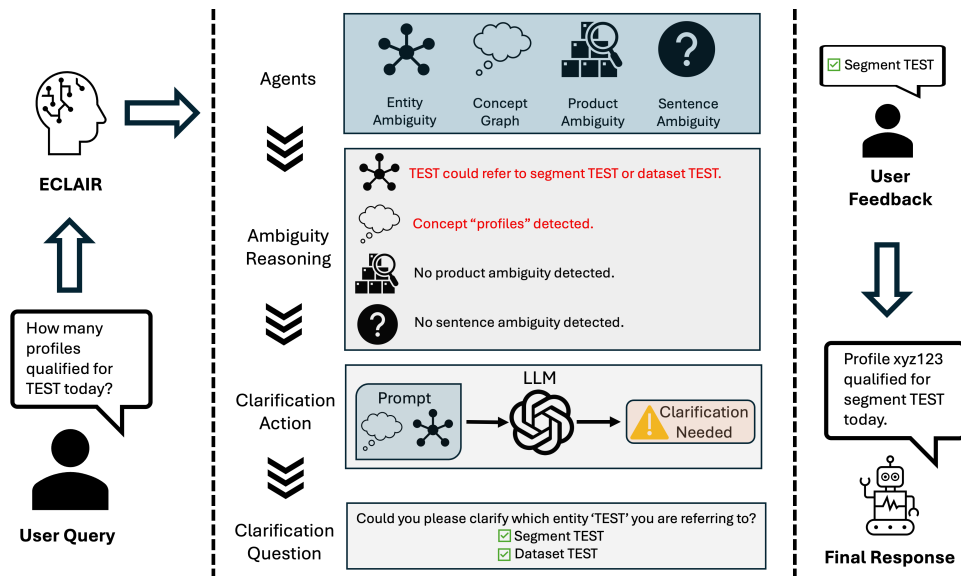


Figure 1: The architecture diagram of ECLAIR.

**Use Case 1: Ambiguous Product** We show a case where a user’s product-related question can refer to multiple Adobe products. ECLAIR successfully captures that this query is ambiguous (from the product classifier agent), and asks a relevant clarification question.

**Use Case 2: Ambiguous Entity** We show cases where a user refers to ambiguous entities. ECLAIR successfully shows that there is entity ambiguity (through the entity linking agent), and generates a useful clarification question. The user then can clarify which specific entity type they are referring to in the interactive UI, and click their choice.

### Framework Evaluation

**Evaluation** We perform a preliminary evaluation on whether a clarification is needed or not. We evaluate this with standard Precision/Recall/F1 metrics.

**Dataset** Our evaluation set consists of a sample from real user queries in the AEP AI Assistant, which was annotated in-house for ambiguity detection. The dataset covers ambiguity types in three categories: contextual, where the context or reference to an object is underspecified; syntactic, where the sentence is malformed or incomplete, leading to indirect interpretation; and aleatoric, where a specific token is undefined or contains multiple possible meanings.

**Baseline** For our baseline, we prompt GPT 3.5 to generate clarification questions if needed. 10 human-crafted examples are provided in the prompt as the few shot examples. We also use GPT 3.5 in ECLAIR for a fair comparison.

**Results** Table 1 shows the results for ECLAIR compared to our few-shot baseline on deciding whether clarification question is needed or not. We first observe that ECLAIR achieves higher precision on “Clarification Needed”, but falls short on recall and F1. We emphasize that for our production setting, we are aiming to maximize precision (that

Model	Category	P	R	F1
Baseline	Clar. Needed	0.732	<b>0.833</b>	<b>0.779</b>
	Clar. Not Needed	0.333	0.214	0.261
	Avg	0.533	0.524	0.520
ECLAIR	Clar. Needed	<b>0.904</b>	0.635	0.746
	Clar. Not Needed	<b>0.438</b>	<b>0.808</b>	<b>0.568</b>
	Avg	<b>0.671</b>	<b>0.721</b>	<b>0.657</b>

Table 1: Comparison of Precision (P), Recall (R), and F1 Score for Baseline and ECLAIR models.

is, we do not want to over-clarify, and only ask clarifications when they are needed). Therefore, ECLAIR’s higher precision means a better and more natural user experience: clarifications are only asked when necessary. ECLAIR yields major improvements on the “Clarification Not Needed” label: we see a boost in precision, recall, and F1, with the largest boost occurring in recall. In turn, ECLAIR yields a overall F1 that is 13% higher than our few-shot method. This emphasizes that ECLAIR not only captures more information and provides a clearer picture of ambiguity, but also yields better metrics compared to standard few-shot methods.

### Conclusion

We present the ECLAIR framework and demonstrate its superior performance in ambiguity detection and clarification question generation on real-world data from the Adobe Experience Platform (AEP) AI Assistant. ECLAIR’s modular and unified approach not only outperforms traditional few-shot methods but also integrates seamlessly into enterprise AI systems, addressing practical challenges such as latency and precision. We showcase two major use-cases of ECLAIR in a live deployment of an enterprise AI assistant.

## Acknowledgments

We would like to thank Victor Soares Bursztyn, Cole Connelly, Nathanael Dereb, Rachel Hanessian, Jordyn Harrison, Sai Sree Harsha, Sai Jayakumar, Shun Jiang, Akash Maharaj, Danny Miller, Kun Qian, Pawan Sevak, Jordan Walker for their insightful discussions and valuable help throughout the development of this work. Their guidance and feedback greatly contributed to the quality of this research.

## References

- Bhambhri, A. 2024. New AI Integrations in Adobe Experience Platform. Accessed: 2024-04-23.
- Deng, Y.; Liao, L.; Chen, L.; Wang, H.; Lei, W.; and Chua, T.-S. 2023. Prompting and Evaluating Large Language Models for Proactive Dialogues: Clarification, Target-guided, and Non-collaboration. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 10602–10621. Singapore: Association for Computational Linguistics.
- Kuhn, L.; Gal, Y.; and Farquhar, S. 2022. Clam: Selective clarification for ambiguous questions with generative language models. *arXiv preprint arXiv:2212.07769*.
- Zhang, M. J.; and Choi, E. 2023. Clarify when necessary: Resolving ambiguity through interaction with lms. *arXiv preprint arXiv:2311.09469*.
- Zhang, T.; Qin, P.; Deng, Y.; Huang, C.; Lei, W.; Liu, J.; Jin, D.; Liang, H.; and Chua, T.-S. 2024. CLAMBER: A Benchmark of Identifying and Clarifying Ambiguous Information Needs in Large Language Models.