DaSH 2024

Data Science with Human-in-the-Loop

Proceedings of the DaSH Workshop at NAACL 2024

June 20, 2024

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
317 Sidney Baker St. S
Suite 400 - 134
Kerrville, TX 78028
USA
Tel: +1-855-225-1962
acl@aclweb.org

ISBN 979-8-89176-101-8

Introduction

We are delighted to welcome to you DaSH 2024, the Fifth Workshop on Data Science with Human-in-the-loop at NAACL 2024!

The aim of this workshop is to stimulate research on the cooperation between humans and computers within the broad area of natural language processing, including but not limited to information extraction, information retrieval and text mining, machine translation, dialog systems, question answering, language generation, summarization, model interpretability, evaluation, fairness, and ethics. We invite researchers and practitioners interested in understanding how to optimize human-computer cooperation and how to minimize human effort along an NLP pipeline in a wide range of tasks and applications.

We hope to bring together interdisciplinary researchers from academia, research labs and practice to share, exchange, learn, and develop preliminary results, new concepts, ideas, principles, and methodologies on understanding and improving human-computer interaction in natural language processing. We expect the workshop to help develop and grow a strong community of researchers who are interested in this topic and to yield future collaborations and scientific exchanges across the relevant areas of computational linguistics, natural language processing, data mining, machine learning, data and knowledge management, human-machine interaction, and intelligent user interfaces. We are thankful to IBM research for sponsoring the workshop and best paper awards.

We hope you have a wonderful time at the workshop.

Cheers!

DaSH 2024 Organizers
Eduard Dragut, Temple University
Yunyao Li, Adobe
Lucian Popa, IBM Research
Shashank Srivastava, UNC Chapel Hill
Slobodan Vucetic, Temple University

Organizing Committee

Organizers

Eduard Dragut, Temple University
Yunyao Li, Adobe
Lucian Popa, IBM Research - Almaden
Shashank Srivastava, University of North Carolina at Chapel Hill
Slobodan Vucetic, Temple University

Program Committee

Chairs

Eduard Dragut, Temple University
Yunyao Li, Adobe
Lucian Popa, IBM Research - Almaden
Shashank Srivastava, University of North Carolina at Chapel Hill
Slobodan Vucetic, Temple University

Program Committee

Arjun Bhalla, Bloomberg
Eleftheria , University of Maryland
Zhijia Chen, Temple University
Aritra Dasgupta, New Jersey Institute of Technology
Rotem Dror, University of Haifa
Varun Embar, Apple
Shivali Goel, Columbia University
Sairam Gurajada, Megagon
Maeda Hanafi, IBM
Lihong He, IBM
Farnaz Jahanbakhsh, MIT
Eser Kandogan, Megagon
Edith Law, University of Waterloo
Akash Maharaj, Adobe
Yiwen Sun, Apple
Yuan Tian, Purdue University

Keynote Talk

Show It or Tell It? Text, Visualization, and their Combination

Marti Hearst
University of California, Berkeley

Abstract: In this talk, Dr. Marti Hearst will share observations about the role of language in information visualization. I will pose questions such as: how do we decide what to express via language vs via visualization? How do we choose what kind of text to use when creating visualizations, and does that choice matter? Does anyone prefer text over visuals, under what circumstances, and why?

Bio: Dr. Marti Hearst is the Interim Dean of the School of Information and a Professor at UC Berkeley in the School of Information and the Computer Science Division. Her research encompasses user interfaces with a focus on scientific document understanding, information visualization with a focus on text, and computational linguistics. She is the author of Search User Interfaces, the first academic book on that topic. She is past President of the Association of Computational Linguistics, an ACM Fellow, a member of the CHI Academy, a SIGIR Fellow, and ACL Fellow, and has received four Excellence in Teaching Awards.

Keynote Talk

Show Reasoning Myths about Language Models: What is Next?

Dan Roth
University of Pennsylvania and Amazon

Abstract: The rapid progress made over the last few years in generating linguistically coherent natural language has blurred, in the minds of many, the difference between natural language generation, understanding, and the ability to reason with respect to the world. Nevertheless, robust support of high-level decisions that depend on natural language understanding, and that require dealing with "truthfulness" are still beyond our capabilities, partly because most of these tasks are very sparse, often require grounding, and may depend on new types of supervision signals.
Dan will discuss some of the challenges underlying reasoning and argue that we should focus on LLMs as orchestrators – coordinating and managing multiple models, applications, and services, to execute complex tasks and processes. I will discuss some of the challenges and present some of our work in this space, focusing on supporting task decomposition and planning.

Bio: Dan Roth is the Eduardo D. Glandt Distinguished Professor at the Department of Computer and Information Science, University of Pennsylvania, a VP/Distinguished Scientist at AWS AI, and a Fellow of the AAAS, the ACM, AAAI, and the ACL. In 2017 Roth was awarded the John McCarthy Award, the highest award the AI community gives to mid-career AI researchers. Roth was recognized "for major conceptual and theoretical advances in the modeling of natural language understanding, machine learning, and reasoning." Roth has published broadly in machine learning, natural language processing, knowledge representation and reasoning, and learning theory. He was the Editor-in-Chief of the Journal of Artificial Intelligence Research (JAIR), has served as the Program Chair for AAAI, ACL and CoNLL. Prof. Roth received his B.A Summa cum laude in Mathematics from the Technion, Israel, and his Ph.D. in Computer Science from Harvard University in 1995.

Keynote Talk

Training Social Skills via Human-AI Collaboration

Diyi Yang
Stanford University

Bio: Diyi Yang is an assistant professor in the Computer Science Department at Stanford University. Her research focuses on human-centered natural language processing and computational social science. She is a recipient of the Microsoft Research Faculty Fellowship (2021), NSF CAREER Award (2022), ONR Young Investigator Award (2023), and Sloan Research Fellowship (2024). Her work has received multiple paper awards or nominations at top NLP and HCI conferences (e.g., ACL, EMNLP, SIGCHI, and CSCW).

Keynote Talk

Model-Aided Human Annotation at Scale

Hadas Kotek
Apple

Bio: Dr. Hadas Kotek is a senior data scientist on the Siri Natural Language Understanding team at Apple. She earned a PhD in Linguistics from MIT and previously held faculty positions at McGill University, New York University, and Yale University. Dr. Kotek develops methodologies for measuring the accuracy and efficiency of data annotation at scale, as well as the safety, robustness, and diversity of the resulting datasets and models, leveraging cross-functional teams to support innovative, product-centric research. Her most recent research is in the domains of model-in-the-loop annotation, ethical AI, and the efficacy of Large Language Models. In Fall 2023, she taught a full-semester seminar on Large Language Models at MIT, where she is currently a Research Affiliate.

Table of Contents

Program

Thursday, June 20, 2024

09:00 - 09:10 *Opening Remarks*

09:10 - 10:00 *Keynote Talk 1*

10:00 - 10:30 *Invited Talk 1*

10:30 - 11:00 *Coffee Break*

11:00 - 12:15 *Workshop Papers*

12:15 - 12:30 *Invited Paper (from Findings of NAACL'24)*

12:30 - 14:00 *Lunch Break*

14:00 - 14:50 *Keynote Talk 2*

14:50 - 15:20 *Invited Papers (from Findings of NAACL'24)*

15:20 - 16:00 *Coffee Break*

16:00 - 16:30 *Invited Talk 2*

16:30 - 17:15 *Panel Discussion*

17:15 - 17:30 *Conclusion of Workshop (Open Discussions)*

APE: Active Learning-based Tooling for Finding Informative Few-shot Examples for LLM-based Entity Matching

Kun Qian[*] and **Yisi Sang** and **Farima Fatahi Bayat**[†] and **Anton Belyi** and **Xianqi Chu**
Yash Govind and **Samira Khorshidi** and **Rahul Khot** and **Katherine Luna** and **Azadeh Nikfarjam**
Xiaoguang Qi and **Fei Wu**[‡] and **Xianhan Zhang** and **Yunyao Li**[§]
kunqian, yisi_sang, f_fatahibayat, a_belyy, xchu23, yash_govind, samiraa
r_khot, kluna, anikfarjam, xiaoguang_qi, fwu7, xianhan_zhang, yunyaoli@apple.com

Abstract

Prompt engineering is an iterative procedure often requiring extensive manual effort to formulate suitable instructions for effectively directing large language models (LLMs) in specific tasks. Incorporating few-shot examples is a vital and effective approach to providing LLMs with precise instructions, leading to improved LLM performance. Nonetheless, identifying the most informative demonstrations for LLMs is labor-intensive, frequently entailing sifting through an extensive search space. In this demonstration, we showcase a human-in-the-loop tool called **APE** (Active Prompt Engineering) designed for refining prompts through active learning. Drawing inspiration from active learning, **APE** iteratively selects the most ambiguous examples for human feedback, which will be transformed into few-shot examples within the prompt. Demo recording can be found with the submission or be viewed at https://youtu.be/OwQ6MQx53-Y.

1 Introduction

Prompt engineering typically serves as the initial step when developing LLM-based applications because it is a relatively fast process and requires fewer technical skills than fine-tuning. Prompt engineering involves crafting and optimization of instructions provided to LLMs. These prompts need to be carefully designed to direct the behavior of LLMs towards performing specific tasks or generating desired outcomes (Liu et al., 2023). While LLMs (e.g., ChatGPT and GPT-4) show impressive capabilities for zero-shot tasks without prior training, their performance can be further enhanced by integrating clear and informative few-shot demonstrations alongside the prompts (White et al., 2023). These demonstrations not only guide the LLMs but also provide examples that contribute to more accurate and contextually relevant outputs, especially for ambiguous cases.

Prompt engineering is a dynamic and iterative process that typically consists of the following stages: (1) *Task Description*: clearly outline the intended task for LLMs, (2) *Few-shot Demonstration*: provide a small number of concrete and helpful demonstrations to illustrate the precise semantics of the task, (3) *Task Input and Completion Request*: present the actual task input and request an LLM completion. For all three steps, minor prompt rephrasing is typically needed, but this task is relatively light and does not require many iterations. However, choosing informative few-shot demonstrations can be a labor-intensive and time-consuming process due to the large search space of the problem. For instance, to identify only 3 demonstration examples out of 100 examples, there are 970,200 (i.e., $100 \times 99 \times 98$) different combinations, a daunting manual task.

Identifying representative and ambiguous examples to enhance the performance of machine learning models is a well-established subject within the active learning community. We can view the few-shot example identification as an active learning problem, where the goal is to find the most informative examples to be included in the prompt to help improve LLMs' performance. Recently, (Diao et al., 2023) proposed the idea of using various active learning sampling strategy to identify few-shot examples prompt engineering. Our work follows the same direction with the main focus being building an interactive tool (with an intuitive user interface) that identifies the most informative few-shot examples through simple human interaction.

In this paper, we present **APE** (**A**ctive **P**rompt **E**ngineering), an intuitive and intelligent prompt engineering tool that iteratively identifies the most informative and ambiguous examples for which a given LLM will likely make a mistake, and then

[*]Work done while working at Apple
[†]Work done while interning at Apple
[‡]Work done while interning at Apple
[§]Work done while working at Apple

Proceedings of the Fifth Workshop on Data Science with Human-in-the-Loop (Language Advances), pages 1–3

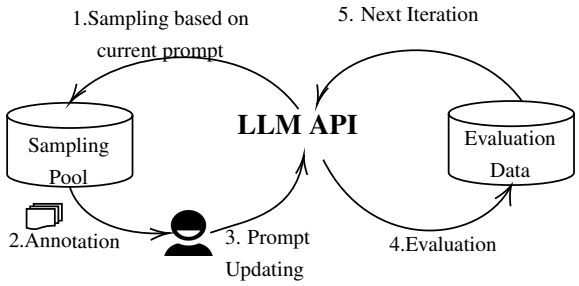

Figure 1: System Overview

provide them in the prompt as few-shot examples after seeking human annotation. Unlike (Diao et al., 2023), which focuses on the backend algorithm services, our goal is to hide the technical details by a carefully designed graph user interface so that we can have a usable tool that truly harnesses the power of active learning.

2 Methodology

The main goal of **APE** is to identify a handful of informative few-shot examples that can boost an LLM's performance. As an active learning tool, it follows the iterative procedure outlined in Figure 1, involving interaction with both a human user and the LLM API for prompt engineering.

The best way to understand **APE** end to end is to watch the video demo of the tool (see the link in the abstract). At a high level, in each iteration, we start with sampling informative examples based on the prompt of the current iteration, which includes applying a user-configured sampling strategy to let the LLM choose the ambiguous examples from the user-provided sampling pool. Next, users annotate the selected examples, potentially including explanations for Chain-of-Thought-style prompting. These newly annotated examples are then used to update the prompt. Lastly, the new prompt is evaluated against evaluation data to report its performance.

The core of the active learning process is the sampling strategy; for simplicity, we will use entity matching, a classic binary classification task, to illustrate the sampling methodology behind **APE**. Given a set $P = \{p_1, \ldots, p_m\}$ of entity pairs, where p_i consists of a pair $\langle e_1^i, e_2^i \rangle$ of entities, the task of entity matching is to learn a binary classifier $f : \langle e_1^i, e_2^i \rangle \rightarrow \{0, 1\}$. In this case, the binary classifier is the LLM in consideration, and the behavior of the classifier is dynamically controlled by the prompts created by **APE**. In our demo video, we used the DBLP-Scholar dataset sampled from (Köpcke et al., 2010) to illustrate **APE**.

2.1 Active Sampling Strategy

The core of **APE** is to find the most informative examples for human annotation to boost the performance of the LLMs. LLMs can be considered excellent student models that can learn effectively from examples. Inspired by active learning, we proposed identifying examples that LLMs are uncertain about to be used as few-shot examples for in-context learning. While both task-specific sampling strategies and task-agnostic sampling strategies can be integrated with **APE**, due to limited space, we focus on the task-agnostic approaches because they allow **APE** to be easily applied to a wide range of problems. In this demo, we introduce two task-agnostic strategies: (1) random-based sampling and (2) self-consistency-based sampling.

Random-based. We randomly select k examples (no replacement) from the sampling pool in each active iteration. Random sampling is simple and fast, and it would work reliably well for many simple tasks. However, for more sophisticated tasks where zero-shot LLMs do not perform well, the chance that random sampling would find informative examples to boost LLMs' performance is low.

Self-consistency-based. To overcome the issue of random sampling, we support self-consistency-based sampling, a strategy inspired by self-consistency (Wang et al., 2023). The core idea is to either run multiple different prompts or the same prompt multiple times in the style of Chain-of-Thought (Wei et al., 2023), allowing the model to generate the final answers with multiple reasoning paths. The consistent answers (e.g., the majority answer) are then chosen as the final answer. A similar idea, known as query-by-committee (QBC)(Seung et al., 1992), has been heavily used in active learning to identify uncertain examples (Settles, 2009). QBC works by training a committee of k slightly different classifiers, e.g., five deep-learning-based classifiers with different architectures, and then let the committee make inferences over the same examples. The disagreement ratio of the committee is used as a proxy to quantify the uncertainty of the examples. The examples with high disagreement ratios are then sent for human annotation.

Our self-consistency-based strategy follows the same idea. Concretely, when selecting examples from the sampling pool, for every entity pair $\langle e_1, e_2 \rangle$, we run the same prompt m times, where

m is a hyperparameter that is usually a small number (in our case, 3). However, each run of the prompt would use a different temperature t, where t gradually grows from 0 to 1 depending on the number of runs. For instance, if $m = 3$, then the three runs of the prompt would have temperatures: $0, 0.5, 1.0$, respectively. Varying the temperature is a way to control the creativity and consistency of LLMs, and we use it to build a committee of slightly different LLMs for uncertain example sampling. Specifically for our entity matching demo scenarios, we collect the m binary labels for a given entity pair p, we then compute the label distributions of the m predictions. We denote the ratio of positive labels as $R^+(p)$ (i.e., $\frac{\#\ positive\ labels}{m}$), and obviously the ratio of negative labels would be $1 - R^+(p)$. With that, we can then compute the label distribution entropy $H(p)$ as follows:

$$-R^+(p)\log R^+(p) - (1-R^+(p))\log(1-R^+(p))$$

the entropy can be viewed as a proxy for uncertainty, and the higher the entropy value, the higher the uncertainty. We then select the examples with the top-k entropy (breaking tie arbitrarily) for human annotations. The annotated examples will be included as new few-shot examples. Note that varying temperatures is for sampling mode only, we set the temperature to zero during prompt evaluation.

Incremental or Fixed Sampling. We offer both incremental sampling and fixed sampling. Incremental sampling accumulates examples labeled in each iteration to form the final few-shot demonstrations. In contrast, fixed sampling selects a predetermined number of examples iteration without accumulating them to create the final prompt.

Human Annotation. By default, we only ask the annotator for the ground truth of a selected example, but for self-consistency-based, we also ask for an explanation of the label provided. Both settings are user-configurable.

3 Concluding Remarks

Due to limited space, we focus on the tooling aspect of **APE** in this demo paper, and are currently working on a research paper that will provide a comprehensive description of the system design, theoretical foundation underlying this optimization problem, and experimental evaluations.

References

Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. Active prompting with chain-of-thought for large language models.

Hanna Köpcke, Andreas Thor, and Erhard Rahm. 2010. Evaluation of entity resolution approaches on real-world match problems. *Proceedings of the VLDB Endowment*, 3(1-2):484–493.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Burr Settles. 2009. Active learning literature survey.

H. S. Seung, M. Opper, and H. Sompolinsky. 1992. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 287–294, New York, NY, USA. Association for Computing Machinery.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

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.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt.

Towards Optimizing and Evaluating a Retrieval Augmented QA Chatbot using LLMs with Human-in-the-Loop

Anum Afzal, Alexander Kowsik, Rajna Fani, Florian Matthes
School of Computation, Information and Technology
Technical University of Munich
{anum.afzal, alexander.kowsik, rajna.fani, matthes}@tum.de

Abstract

Large Language Models have found application in various mundane and repetitive tasks including Human Resource (HR) support. We worked with the domain experts of SAP SE to develop an HR support chatbot as an efficient and effective tool for addressing employee inquiries. We inserted a human-in-the-loop in various parts of the development cycles such as dataset collection, prompt optimization, and evaluation of generated output. By enhancing the LLM-driven chatbot's response quality and exploring alternative retrieval methods, we have created an efficient, scalable, and flexible tool for HR professionals to address employee inquiries effectively. Our experiments and evaluation conclude that GPT-4 outperforms other models and can overcome inconsistencies in data through internal reasoning capabilities. Additionally, through expert analysis, we infer that reference-free evaluation metrics such as G-Eval and Prometheus demonstrate reliability closely aligned with that of human evaluation.

1 Introduction

In recent years, incorporating Artificial Intelligence (AI) into various sectors has led to significant improvements in automated systems, particularly in customer service and support. Since the offset of Large Language Models (LLMs), more companies are now incorporating Natural Language Processing (NLP) techniques to minimize the need for human support personnel, especially domain experts (Shuster et al., 2021). With a chatbot providing accurate and comprehensive responses promptly, domain experts can redirect their focus towards higher-value tasks, leading to potential cost savings and improved productivity within the HR department. Moreover, an effective chatbot can play a pivotal role in enhancing overall employee satisfaction and engagement by delivering timely and relevant assistance.

To this end, we worked with a SAP SE on developing an HR chatbot to evaluate the potential of LLMs on industrial data. We used domain experts as *a human-in-the-loop* through various iterations of LLM-centric development such as dataset collection, prompt optimization, and most importantly the evaluation of model outputs.

The well-known Retrieval Augmented Generation (RAG) (Lewis et al., 2021) approach is ideal for this use case as it allows the model to produce more grounded answers, hence reducing hallucinations. We optimized different modules of the standard RAG pipeline such as the retriever and model prompts, while constantly incorporating feedback from the domain experts. While the retrieval accuracy of an LLM could still be assessed to a degree, the generative nature of LLMs makes evaluation of the generated output quite challenging. To overcome this, we explored the effectiveness of both traditional reference-based and reference-free (LLM-based) automatic evaluation metrics while using human evaluation as a baseline.

We benchmark OpenAI's models in our experiments while using the open-source LongT5 (Guo et al., 2022) and BERT (Devlin et al., 2019) as a baseline. In essence, both the industry and the research community could benefit from our findings related to the retriever and the reliability of automatic evaluation metrics.

2 Corpus

The dataset used in the development of the HR chatbot was compiled using SAP's internal HR policies with the help of domain experts. While each sample forms a triplet consisting of a Question, Answer, and Context, additional metadata such as the user's region, company, employment status, and applicable company policies were also included. A snippet of such a sample is shown in Appendix A.4. The dataset was compiled using two separate sources

Proceedings of the Fifth Workshop on Data Science with Human-in-the-Loop (Language Advances), pages 4–16

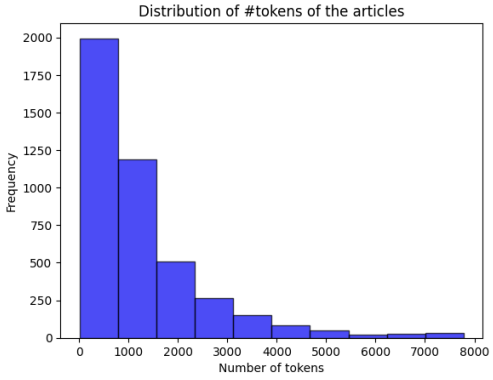

Figure 1: Distribution over the number of tokens of all unique articles in our HR dataset.

to have a mix of a gold dataset (FAQ dataset) and a user-utterance dataset (UT dataset). Both datasets follow the same structure and differences exist in the distribution of the questions. We extracted all unique HR articles to form a knowledge base for answering new user questions. Additionally, an evaluation set of 6k samples was used to evaluate both the retriever and the chatbot as a whole.

2.1 Dataset Collection

FAQ Dataset (N≈48k): This is a collection of potential questions, along with their corresponding articles and gold-standard answers. It is carefully created and curated by domain experts based on the company's internal policies.

UT Dataset (N≈41k): This is a collection of real user utterances (UT) gathered from previous iterations of the chatbot. Inspired by a semi-supervised learning approach, a simplistic text-matching approach was implemented that mapped each user query to a question from the FAQ dataset. The chatbot logs from this development cycle were inspected and corrected by the domain experts.

2.2 Dataset Statistics

Figure 1 shows that the majority of the articles in our dataset have under 4k tokens. Hence, they can easily fit into the context window of OpenAI models. As displayed in **??**, the most asked questions in the dataset revolve around payslips, leave days of any kind, and questions regarding management.

3 Methodology

In general, the HR chatbot follows the standard RAG pipeline with optimizations done on individual modules with the help of domain experts as shown in Figure 2. The methodology illustrates various parts of the chatbot pipeline that are influenced by a human-in-the-loop and is further discussed in Appendix B.

3.1 Retriever

We compiled a comprehensive knowledge base of all possible HR articles occurring in the whole dataset as the basis for retrieval, resulting in roughly 50k unique articles. Given a user utterance, the goal of the retriever is to find the most relevant article from the collection. While the technical details for each retriever may differ, in general, they are both embedding-based. Technical details of the Retriever module are discussed in Appendix D.1.

Moreover, we developed extensive filter functionalities, ensuring that the vector search only considers articles relevant to the user, like their country, region, or employment status as shown in Table 4. For example, from the top retrieved articles, we filter them to only keep the ones that are applicable to the employee and then pick the article with the maximum similarity score from the filtered list.

3.1.1 Dense Passage Retriever (BERT)

Dense Passage Retriever (DPR) fine-tunes *bert-base-uncased* embedding to generate a model that given a user query, retrieves the most relevant article from a set of documents. The dataset used for training was processed to contain questions paired with their respective gold answers, as well as positive and negative contexts for each question. A triplet loss function (Hoffer and Ailon, 2018) was used for training such that the relevant article served as the positive context, with two random articles from the entire dataset providing the negative contexts. This retriever is used in the framework with the fine-tuned LongT5 model and also serves as a baseline for evaluating the OpenAI retriever.

3.1.2 Vector Search (OpenAI)

The OpenAI Retriever is plain vector search, that utilizes the *text-embedding-ada-002* embedding model by OpenAI to generate embeddings for each article, followed by using similarity search to find the relevant article. To further enhance retrieval accuracy, we implemented various **Query Transformation** techniques[1] (Cormack et al., 2009a). These methods alter the user query into a different

[1] https://docs.llamaindex.ai/en/stable/optimizing/advanced_retrieval/query_transformations/

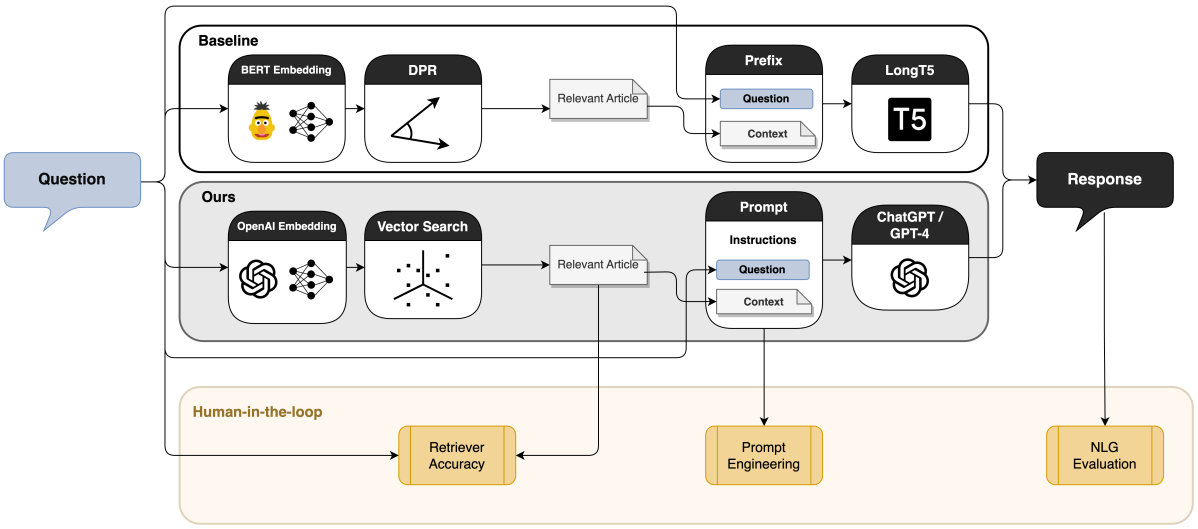

Figure 2: Block diagram of the methodology introduced in our paper, illustrating baseline and Open AI models, highlighting the role of the human-in-the-loop during development

representation using LLMs before the embedding model computes the query vector. The following three query transformation methods were explored and evaluated:

1) Intended Topics: Inspired by Ma et al. (2023), the user question is sent to an LLM with the instruction to return a list of three intended topics of the question, which are then embedded instead of the user question.

Example: *How to request a parental leave?* → *parental leave, childcare leave, maternity leave*

2) HyDE (Hypothetical Document Embeddings): In this method introduced by Gao et al. (2022), the user question is transformed by an LLM into three distinct excerpts from potential HR articles answering the original question. These parts are then embedded instead of the user question itself. This approach leads to query embeddings that are very close to the article embeddings, because of the very similar content.

Example: *How to request a parental leave?* → *To request parental leave, please submit...*, *If you wish to request...*, ...

3) Multi-Query: This method[2] employs LLMs to generate multiple variations of a user's question varying in length and phrasing but maintaining the same meaning and intent as the original question. We then embed each of these variants individually. Along with the embedded original question, we perform a vector search for each query, combining the results using Reciprocal Rank Fusion (Cormack et al., 2009b). Additionally, we include queries from the *Intended Topics* and *HyDE* methods.

Example: *parental leave request?* → *How can I request a parental leave?*, *Where can I apply for parental leave?*, ...

3.2 NLG Module

3.2.1 LongT5 (Fine-tuning driven)

We fine-tuned LongT5 (Guo et al., 2022), employing the local-attention-based variant[3], which consists of 296 million trainable parameters. This model was fine-tuned on a combination of the FAQ dataset and UT dataset for a generative question-answering task. To limit computational requirements, we fine-tuned it on a context window of 7168 tokens, retaining approximately ∼86K samples from the original dataset to avoid truncation.

3.2.2 OpenAI Models (Prompt driven)

We used OpenAI's ChatGPT and GPT-4 to generate the answer to the user's query by passing both the user query and the retrieved article via a meaningful prompt. We conducted extensive prompt engineering to tailor the responses of the LLMs to the company's requirements for an HR chatbot. Prompt engineering was an iterative process that included our qualitative analysis and multiple small evaluations of 10-100 sample responses by the company's HR experts who served as the *human-in-the-loop*. We analyzed feedback from

[2] https://docs.llamaindex.ai/en/latest/examples/retrievers/reciprocal_rerank_fusion/

[3] https://huggingface.co/google/long-t5-local-base

these evaluation runs and addressed the main issues in the next iteration of the process to produce the final prompt shown in Table 5.

3.3 Evaluation Framework

For our analysis we employ Reference-based evaluation metrics such as BERTScore (Zhang et al., 2019), ROUGE (Lin, 2004), and BLEU (Papineni et al., 2002). We also explore the concept of using LLM as an evaluator, and finally, we assess the effectiveness of automated metrics by involving domain experts in a human-in-the-loop process.

3.3.1 Retriever Evaluation

Our primary evaluation metric for the retriever is accuracy, defined as the percentage of times the retriever returns the correct article for a given question.

3.3.2 Human Evaluation Setup

The domain experts who served as the human-in-the-loop brought a high level of precision and insight to the evaluation process. Apart from dataset curation, they also evaluated the performance of the retriever by verifying the correctness of the retrieved articles. After discussion with domain experts, we found four dimensions across which the quality of the model's output could be evaluated on a score between 1 - 5 following a 5-point Likert (Likert, 1932) scale. One domain expert evaluated 100 samples across the fine-tuned LongT5, ChatGPT and GPT-4 across *Readability*, *Relevance*, *Truthfulness*, and *Usability*.

3.3.3 Reference-based Metrics

In evaluating the effectiveness of reference-based metrics, we examine two distinct categories: N-gram-based and embedding-based metrics metrics.

N-gram based metrics: N-gram-based metrics, such as BLEU (Bilingual Evaluation Understudy) and ROUGE (Recall-Oriented Understudy for Gisting Evaluation), assess the similarity between the generated response and the ground truth answer by analyzing the overlap of n-grams.

Embedding-based metrics: Embedding-based metrics, such as BERTScore, leverage deep contextual embeddings from language models like BERT to assess the semantic similarity between generated and reference texts.

3.3.4 Reference-free Metrics

In the evolving landscape of Natural Language Generation evaluation, LLM-based metrics emerge as a compelling alternative, offering insights into model performance without the constraints of pre-defined reference responses. Details regarding the prompts used for these Reference-free metrics are present in Appendix C.

Prompt-based Evaluation: Prompt-based evaluation is at the forefront of NLG advancements, particularly with the utilization of LLMs (Li et al., 2024). Inspired by **G-Eval**, we followed the approach described by Liu et al. (2023) and tailored the prompts to be suitable for the evaluation of a question-answering task.

Tuning-based Evaluation: Nowadays, there is a significant shift toward leveraging open-source language models, such as LLaMA (Touvron et al., 2023), for fine-tuning purposes. We utilize **Prometheus** (Kim et al., 2023), which stands out for its fine-tuned evaluation capability, leveraging a large language model to perform nuanced analysis based on customized score rubrics (Li et al., 2024). This unique approach enables Prometheus to evaluate text generation tasks comprehensively, considering factors such as creativity, relevance, and coherence without relying on reference texts.

4 Results and Discussion

4.1 Dense Passage Retriever

As depicted in Table 2, surprisingly the BERT-based DPR significantly outperforms all new methods with a top-1 accuracy of 22.24%, whereas the OpenAI-based retriever only reaches a top-1 accuracy of 11.12%. Of the latter, the best performer is *Multi-Query*, with 10.92%, yet this still falls short of the *Basic* retriever (no query transformation). These results resonate with the findings of Weller et al. (2024), confirming that query transformations, do not always lead to better performance. Our understanding is that the retriever performs poorly mainly because of the noise attributed to the dataset. It is worth noting, that our dataset contains many variant articles for a given topic or question, with only small differences such as the region or the employee role. Hence, the incorrect article may still contain sufficient knowledge to address user queries. We confirmed these findings with our domain experts and elaborated on them further in Appendix A.3. Further results on up to top-5 articles are shared in Appendix E.1.

However, to assess the effectiveness of the newly implemented methods on a different dataset, we gathered 10k samples from CQADupStack English

	HR Test Dataset	Stackexchange English
Method	**top-1**	**top-1**
BERT-based DPR	**22.24%**	-
Basic	**11.12%**	69.5%
Intended Topics	9.33%	57.25%
HyDE	10.01%	65.91%
Multi-Query	10.92%	**71.31%**

Table 2: Retriever accuracy on the HR test data and the Stackexchange benchmark dataset for various retriever methods on top-1 retrieved articles

(Hoogeveen et al., 2015), a collection of English language questions and their top answers from the Stackexchange English forum. We used the same embedding model as the HR dataset to embed this new data and evaluated its top-1 accuracy. It can be observed that the *Intended Topics* method and *HyDE* both underperform compared to the *Basic* retriever. However, the *Multi-Query* method did produce a higher top-1 accuracy. During our experiments, we noticed that these methods are greatly influenced by the choice of query transformation prompts. For instance, when *HyDE* responses closely matched the desired replies, the accuracy was significantly higher. These methods also achieved higher accuracies than the *Basic* on other types of data, which indicates that the performance is also dependent on the type of data used. This might explain why these methods couldn't achieve higher accuracy on the HR dataset.

4.2 NLG Evaluation

We use the previously optimized DPRs with the top-1 article for our NLG Module consisting of ChatGPT, GPT-4 and fine-tuned LongT5 as shown in Figure 2. An overview of all evaluation scores highlighting model performance across several dimensions is summarized in Table 3.

Overall, GPT-4 shows clear domination in terms of generation capabilities for an HR chatbot. N-gram-based evaluation scores such as ROUGE and BLEU are quite low due to the generative nature of the (L)LMs, as the answer may contain words different than the reference answers. Nonetheless, these results establish GPT-4 as the leading model, effectively combining advanced language skills with the demands of content accuracy and user engagement. On the other hand, the fine-tuned LongT5's performance is observed to be inferior when benchmarked against the OpenAI models. This outcome is consistent with the anticipated advancements in LLMs, which are progressively outpacing the capabilities of fine-tuning-driven models. The performance of ChatGPT has been notably strong, trailing marginally behind GPT-4 in only a few scoring categories. Its close performance to GPT-4 raises important considerations for the trade-offs between computational efficiency and output quality.

Metric	**ChatGPT**	**GPT-4**	**LongT5**
Reference-based Evaluation			
BLEU Score	0.27	0.28	**0.41**
ROUGE-1	0.48	**0.52**	0.51
ROUGE-2	0.36	0.35	**0.43**
ROUGE-L	0.46	**0.50**	0.49
BERTScore_P	0.88	0.90	**0.91**
BERTScore_R	**0.96**	0.93	0.91
BERTScore_F1	0.90	**0.91**	0.90
Reference-free Evaluation (LLM-based)			
G-Eval: Relevance	4.03	**4.51**	3.17
G-Eval: Readability	4.26	**4.49**	3.52
G-Eval: Truthfulness	4.12	**4.80**	3.36
G-Eval: Usability	4.67	**4.79**	3.29
Prometheus: Relevance	3.25	**3.70**	2.83
Prometheus: Readability	3.07	**4.22**	3.73
Prometheus: Truthfulness	3.20	**3.75**	3.32
Prometheus: Usability	3.98	**4.32**	2.83
Domain Expert Evaluation			
Human Eval: Readability	4.31	**4.76**	4.02
Human Eval: Relevance	4.31	**4.67**	3.46
Human Eval: Truthfulness	4.09	**4.41**	3.67
Human Eval: Usability	3.32	**4.11**	2.59

Table 3: Average Evaluation Scores. BLEU (0 to 1), ROUGE (0 to 1) and BERTScore (-1 to +1) were computed on 200 samples, Prometheus (1 to 5) on 60 samples, and Domain Expert Evaluation (1 to 5) & G-Eval (1 - 5) on 100 samples.

4.3 Correlation Analysis

Inspired by Zhong et al. (2022), we assessed the reliability of the evaluation score using Spearman (Myers and Sirois, 2004) and Kendall (Abdi, 2007) correlation coefficients in Table 9.

Human Evaluation & Reference-based Metrics Due to its limited innovation, LongT5 typically produces text with fewer novel sentences, resulting in more favorable scores from n-gram-based metrics like BLEU and ROUGE. The analysis of GPT-3.5 and GPT-4, in particular, illuminates a significant gap between automated metrics and human judgment. As these models generate more varied and longer sentences, their outputs increasingly diverge from the patterns recognized by word-overlap metrics, such as BLEU and ROUGE. For instance, GPT-4's BLEU score correlation marks a clear disconnect, indicating that as text generation becomes more complex, the less effective traditional metrics are in evaluating it. This discrepancy calls into question the reliance on current automated metrics

for assessing the creativity and nuance of outputs from advanced language models, highlighting the need for more sophisticated evaluation frameworks that can better align with human judgment.

Human Evaluation & Reference-free Metrics Despite similar average scores between Reference-free metrics and Domain Expert evaluations shown in Table 3, their correlations are low. Since these methods measure linear and ordinal relationships, similar averages in evaluations do not imply a strong correlation as depicted in Table 9.

Overall, while Prometheus and G-Eval both serve as proxies for human evaluation, their effectiveness varies by model and evaluation criteria. While G-Eval excels in assessing truthfulness, its capability in evaluating readability and usability lags behind. Prometheus on the other hand, outperforms G-Eval in assessing usability across all models. However, G-Eval shows a steadier performance across different models, particularly with LongT5, suggesting its robustness in accurate evaluations. Both metrics show weak alignment in assessing readability, reflecting the inherent challenge of one LLM evaluating another's ability to produce easily understandable text.

Additionally, LLM-based metrics sometimes fail to align with human judgment, particularly when answers or instructions involve unfamiliar HR terms or sensitive information. Notably, OpenAI models' novel answers exhibit lower human correlation compared to LongT5, which provides answers more similar to the golden response.

5 Related Work

Previously, domain-specific chatbots meant for a specific task were designed using conversational AI frameworks like RASA (Bocklisch et al., 2017). Latest advancements in NLP have shifted focus towards employing and optimizing LLM-based RAG (Gao et al., 2024b). Chen et al. (2023) experiment with ChatGPT and several other open-source models like Vicuna to benchmark their capabilities in RAG, and Wang et al. (2023) use a smaller secondary domain-specific model to assist a bigger LLM on a domain-specific question answering task on industrial data. Recent studies have explored various retrieval methods, including dense vector retrieval (Karpukhin et al., 2020a), sparse retrieval (Robertson et al., 2004, 2009), and hybrid approaches (Guu et al., 2020a), to improve the relevance and diversity of retrieved documents. Guu et al. (2020b) uses various RAG techniques to ensure that chatbot responses are based on relevant HR policies, leading to accurate and helpful user support.

Given the diverse distribution of the text generated by LLMs, conventional metrics are not suitable for its evaluation (Wei et al., 2021; Belz and Reiter, 2006; Novikova et al., 2017). Consequently, a lot of follow-up research has come up in the area of NLG Evaluation (Gao et al., 2024a; Li et al., 2024). Specifically focusing on RAG, Es et al. (2024) released a Framework for the automatic evaluation of generated output using LLM-based metrics with a focus on faithfulness. A similar approach is followed by Saad-Falcon et al. (2023) in their framework ARES which also evaluates the performance of RAG systems over relevance and faithfulness by fine-tuning a lightweight LM judge.

6 Conclusion

By optimizing retrieval techniques and benchmarking state-of-the-art LLMs with the help of domain experts, we show how LLM-based applications could benefit from a domain expert as human-in-the-loop within various iterations of the development. Even though our optimizations on the OpenAI-based retriever show minor improvements, the accuracy remains quite low due to the poor quality of the evaluation dataset. Nonetheless, both ChatGPT and GPT-4 show competence when addressing the user query. This hints that the internal reasoning capabilities and domain knowledge of these LLMs are strong enough to overcome the knowledge in the *supposed incorrect article*. This also suggests that, given the nature of the dataset used, the accuracy metric used for the evaluation of the retriever is not a good measure of its performance. We employed and studied a range of evaluation metrics and concluded that in contrast to traditional evaluation approaches such ROUGE & BERTScore, LLM-based metrics such as Prometheus and G-Eval come very close to human evaluation on average. Nonetheless, our findings reiterate the importance of human judgment, particularly in use cases that require an understanding of a specific domain.

Acknowledgements

The work outlined in this paper is part of a research project between the Technical University of Munich and SAP SE under SAP@TUM Col-

laboration Lab. The authors would like to thank Patrick Heinze, Christopher Pielka, Albert Neumueller, Darwin Wijaya from the SAP IES as well as the Domain Experts from the Human Resource department for their continued support.

Limitations

In our experiments, we mostly worked with OpenAI models which are closed-source and hence raise concerns of privacy. Additionally, their large sizes inhibited fine-tuning as they required extensive hardware. Fine-tuning open source and smaller models tailored to HR-specific contexts could further improve response accuracy and relevance. Additionally, since we worked with only one domain expert for the evaluation of the generated answers, the human evaluation might be biased. Because of the data protection concerns with the associated dataset, we cannot make the dataset open source. We employed basic filtering techniques to include user-specific information and context, more advanced approaches could be explored to include this information into the LLM prompt.

Ethics Statement

Throughout our experiments, we strictly adhere to the ACL Code of Ethics. The dataset used for our research was anonymized to not include any personal information. We employed in-house domain experts, who receive a full salary for evaluation for generated summaries. They were informed about the task and usability of data in the research. Their annotations were stored in an anonymized fashion, mitigating any privacy concerns. Through our fine-tuning strategies, no additional bias was introduced into the models, other than what might already be part of the dataset. The goal of the research was to optimize an LLM-centric chatbot with the help of a human-in-the-loop. The results and discussions in this paper are meant to further promote research in LLM-based development, bridging the gap between academia and application.

References

Hervé Abdi. 2007. The kendall rank correlation coefficient. *Encyclopedia of measurement and statistics*, 2:508–510.

Anja Belz and Ehud Reiter. 2006. Comparing automatic and human evaluation of nlg systems. In *11th conference of the european chapter of the association for computational linguistics*, pages 313–320.

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2023. Benchmarking large language models in retrieval-augmented generation.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009a. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759, New York, NY, USA. Association for Computing Machinery.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009b. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*.

Mingqi Gao, Xinyu Hu, Jie Ruan, Xiao Pu, and Xiaojun Wan. 2024a. Llm-based nlg evaluation: Current status and challenges. *arXiv preprint arXiv:2402.01383*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2024b. Retrieval-augmented generation for large language models: A survey.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. LongT5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736, Seattle, United States. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020a. Realm: Retrieval-augmented language model pre-training.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020b. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Elad Hoffer and Nir Ailon. 2018. Deep metric learning using triplet network.

Doris Hoogeveen, Karin Verspoor, and Timothy Baldwin. 2015. Cqadupstack: A benchmark data set for community question-answering research.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020a. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020b. Dense passage retrieval for open-domain question answering.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. 2023. Prometheus: Inducing fine-grained evaluation capability in language models. *arXiv preprint arXiv:2310.08491*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Zhen Li, Xiaohan Xu, Tao Shen, Can Xu, Jia-Chen Gu, and Chongyang Tao. 2024. Leveraging large language models for nlg evaluation: A survey. *arXiv preprint arXiv:2401.07103*.

Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Xinbei Ma, Yeyun Gong, Pengcheng He, hai zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Leann Myers and Maria J Sirois. 2004. Spearman correlation coefficients, differences between. *Encyclopedia of statistical sciences*, 12.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for nlg. *arXiv preprint arXiv:1707.06875*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Stephen Robertson, Hugo Zaragoza, and Michael Taylor. 2004. Simple bm25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 42–49.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2023. Ares: An automated evaluation framework for retrieval-augmented generation systems.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Zezhong Wang, Fangkai Yang, Pu Zhao, Lu Wang, Jue Zhang, Mohit Garg, Qingwei Lin, and Dongmei Zhang. 2023. Empower large language model to perform better on industrial domain-specific question answering. *arXiv preprint arXiv:2305.11541*.

Wei Wei, Bo Dai, Tuo Zhao, Lihong Li, Diyi Yang, Yun-Nung Chen, Y-Lan Boureau, Asli Celikyilmaz, Alborz Geramifard, Aman Ahuja, et al. 2021. The first workshop on evaluations and assessments of neural conversation systems. In *The First Workshop on Evaluations and Assessments of Neural Conversation Systems*.

Orion Weller, Kyle Lo, David Wadden, Dawn Lawrie, Benjamin Van Durme, Arman Cohan, and Luca Soldaini. 2024. When do generative query and document expansions fail? a comprehensive study across methods, retrievers, and datasets. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1987–2003, St. Julian's, Malta. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. *arXiv preprint arXiv:2210.07197*.

A Dataset

A.1 Dataset Collection

FAQ Dataset: The internal HR policies of the company consist of Wiki articles, where each article contains a description text followed by some frequently asked questions. The FAQ dataset was constructed by the domain articles by compiling all the FAQ questions from all articles. Each FAQ question is in the form of a triplet where the context is the original Wiki article the question was derived from. **UT Dataset:** The user utterance (UT) dataset was compiled using the user utterances collected from the chatbot logs. To reduce the manual labeling effort, a simple text-matching approach was deployed that mapped each user query to one of the questions from the FAQ dataset. The respective answers and context of the matched question were used to create the triplets that form the UT dataset.

A.2 Dataset Pre-processing

We cleaned the dataset using regular expressions and with the help of LLMs. This involved removing unnecessary formatting like HTML tags, leading or trailing white spaces and newline characters, and removing some wasteful markdown annotations without text. This process thus reduced the number of tokens in each document. Some of the documents were too long to fit into the LLM's context window, so we excluded them from our analysis.

A.3 Dataset Challenges

We discovered that our dataset contains multiple articles answering most questions. These articles differ in a few characters, often in an unequal amount of whitespaces, or a few exchanged words, or even entire sections not present in other articles. This situation leads to multiple slightly different versions of the same article present in the dataset, all linked to similar questions. Consequently, the retriever often retrieves very relevant articles that do not exactly match the gold standard article but are a slightly different version.

To address this, we implemented an evaluation method measuring the Levenshtein distance between the retrieved article and the gold article. If this distance is below a threshold of 100, we consider it a successful retrieval. However, this approach does not match articles with varying sections, as the Levenshtein distance is much higher, and we didn't want to risk matching incorrect articles by increasing the threshold. All of the results in Table 2 are using this evaluation method.

As the DPR is fine-tuned on the dataset, which likely has a strong imbalance in the counts of different article versions, it tends to favor the most common version. This bias contributes to its higher accuracy, as the retriever fetches the correct article more often than not.

A.4 Dataset Example

Table 4 shows an example sample from the FAQ dataset representing the training triplet along with all metadata.

DATA TRIPLET
Question: How can I apply for half a day of holiday? **Answer:** Unfortunately, vacation days in your country can only be taken as full days. **Context:** {Relevant Article}
META DATA
User Role: Employee **Name of KBA:** Vacation **Company Name:** {Company Name} **Company Code:** {Company Code} **Region:** {Region} **Country Code:** {Country Code} **FAQ Category:** {FAQ Category} **Process ID:** {Process ID} **Service ID:** {Process ID}

Table 4: HR Dataset Sample

B Human-in-the-Loop

As shown in Figure 2, the domain experts are involved in various parts of the development cycle explained below:

Dataset Collection: The domain experts play a big role in the compilation and quality control of the datasets used in this paper

Prompt Optimization: The domain experts evaluated answers generated by models on various prompt versions. They also provided guidelines the chatbot should follow when addressing the user query which is reflected in the final prompt displayed in Table 5.

Evaluation: Domain experts also served as the human annotators for the answers generated by (L)LMs which helped us assess the quality of an-

swers as well as study the effectiveness of automatic evaluation scores.

C Prompts Samples

In this section, we provide the extensive list of prompts used for the OpenAI Models for the Chatbot Pipeline, as well as the prompts used for the LLM-based Metrics.

C.1 Prompts used for OpenAI models

The optimized prompt used for ChatGPT and GPT-4 during our experiments is shown in Table 5.

C.2 G-Eval Evaluation Metric Prompt

The evaluation prompt used for the Readability Criteria is shown in Table 6. The prompts for other criteria (Truthfulness, Usability, Relevance) follow similar instructions as the one shown for the Readability prompt.

C.3 Prometheus Evaluation Metric Prompt

The prompt for the Prometheus Evaluation Metric outlined in Table 7 was based on the official paper's guidelines (Kim et al., 2023) for Feedback Collection. This specific prompt illustrates the Readability Criteria and was similarly adapted for other criteria such as Truthfulness, Relevance, and Usability. In general, both LLM-based metrics follow similar evaluation criteria in the prompts.

D Technical Details

D.1 Retriever

It is worth noting that we embed the whole article and do not perform chunking. As shown in Figure 1, these articles are quite long. To cater to the limited context window of the models, we opt for the top-1 article to be passed as context. This also makes sense for our use case as the dataset is designed such that the answer to any given HR question usually exists in only one article.

D.2 Dense Passage Retriver Training

Dense Passage Retriever (DPR) (Karpukhin et al., 2020b) powered by Haystack[4] uses the *bert-base-uncased* embedding model by *google-bert*, openly available on HuggingFace. DPR training aims to generate a model that creates embeddings where the question embedding closely aligns with the relevant context embedding. During retrieval, the user query is processed through the previously trained retriever, producing a query vector in the same embedding space as the articles. This query vector is then compared to all article vectors within the vector store using cosine similarity. The top-k articles belonging to the embeddings with the highest cosine similarities are returned.

D.3 LongT5 Fine-tuning

During fine-tuning of the LongT5 models, the training process was configured with a learning rate of 1e-4 and a batch size of 8, spanning 5 epochs.

E Results and Evaluation

Throughout our research, we encountered several challenges that warrant attention. The variability in retrieved articles due to slight differences in content or formatting posed complexities in evaluating retrieval accuracy and ensuring consistency in response generation. Addressing this challenge may require further refinement of the retrieval mechanism or additional preprocessing steps to standardize the retrieved content.

E.1 Retriever

The accuracy of both DPR on the top-1, top-2, top-3, and top-5 articles on both retrievers is shown in Table 8. As expected, the accuracy of the retriever module increases as the value of k is increased. However, we are limited to including only top-1 articles because the articles are quite long and more samples may not fit in the model's context window. The BERT-based DPR model still significantly outperforms all new methods with a top-1 accuracy of 22.24% and a top-5 accuracy exceeding 40%. The new retriever, in comparison, only reaches a top-1 accuracy of 11.12% and a top-5 accuracy of 18.53% on the same dataset. These results in general are quite underwhelming and mainly attributed to the dataset challenges described in Appendix A.3.

DPR	top-1	top-2	top-3	top-5
BERT-based	22.24%	30.03%	35.08%	**40.06%**
OpenAI-based	11.12%	15.06%	16.82%	**18.53%**

Table 8: Retriever Accuracy on the HR test dataset for various values of k on the HR Dataset. The OpenAI-based DPR uses the *Basic* method.

[4] https://haystack.deepset.ai/

SYSTEM PROMPT

You are an HR chatbot for SAP SE and you provide truthful and concise answers to employee questions based on provided relevant HR articles.
1. Stay very concise and keep your answer below 150 words.
2. Do not include too much irrelevant information unrelated to the posed question.
3. Keep your response brief and on point.
4. Include URLs from the relevant article if it is important to answer the question.
5. If the answer applies to specific labs/countries/companies, include this information in your response.
6. Refer to the employee directly as "you" and not indirectly as "the employee".
7. If the provided HR article does not include the answer to the question, tell the employee to create an HRdirect ticket.
8. Answer in a polite, personal, user-friendly, and actionable way.
9. Never make up your response! If you do not know the answer to the question, just say so and ask the user to create an HRdirect ticket!

USER PROMPT

Question: {question}
Relevant Article: {article}

Table 5: Chatbot Prompt for OpenAI Models

SYSTEM PROMPT

You will be given a generated answer for a given question. Your task is to act as an evaluator and compare the generated answer with a reference answer on one metric. The reference answer is the fact-based benchmark and shall be assumed as the perfect answer for your evaluation. Please make sure you read and understand these instructions very carefully. Please keep this document open while reviewing, and refer to it as needed.
Evaluation Criteria: {criteria}
Evaluation Steps: {steps}

USER PROMPT

Example: {example}
Question: {question}
Generated Answer: {generated_answer}
Reference Answer: {reference_answer}
Evaluation Form: Please provide your output in two parts separate as a Python dictionary with keys rating and explanation. First the rating in an integer followed by the explanation of the rating.
{metric_name}

METRIC SCORE CRITERIA

{The degree to which the generated answer matches the reference answer based on the metric description.}
Readability(1-5) - Please rate the readability of each chatbot response. This criterion assesses how easily the response can be understood. A response with high readability should be clear, concise, and straightforward, making it easy for the reader to comprehend the information presented. Complex sentences, jargon, or convoluted explanations should result in a lower readability score.

METRIC SCORE STEPS

{Readability Score Steps}
1. Read the chatbot response carefully.
2. Assess how easily the response can be understood. Consider the clarity and conciseness of the response.
3. Consider the complexity of the sentences, the use of jargon, and how straightforward the explanation is.
4. Assign a readability score from 1 to 5 based on these criteria, where 1 is the lowest (hard to understand) and 5 is the highest (very easy to understand).

Table 6: G-Eval Prompt Example for Readability Criteria

SYSTEM PROMPT

Task Description: An instruction (might include an input inside it), a response to evaluate, a reference answer that gets a score of 5, and a score rubric representing an evaluation criterion is given.
2. After writing a feedback, write a score that is an integer between 1 and 5. You should refer to the score rubric.
3. The output format should look as follows: Feedback: [write a feedback for criteria] [RESULT] [an integer number between 1 and 5].
4. Please do not generate any other opening, closing, and explanations.

Question to Evaluate: {instruction}
Response to Evaluate: {response}
Reference Answer (Score 5): {reference answer}
Score Rubrics: {criteria description}
Score 1: {Very Low correlation with the criteria description}
Score 2: {Low correlation with the criteria description}
Score 3: {Acceptable correlation with the criteria description}
Score 4: {Good correlation with the criteria description}
Score 5: {Excellent correlation with the criteria description}
{criteria description}: Readability(1-5) - Please rate the readability of each chatbot response. This criterion assesses how easily the response can be understood. A response with high readability should be clear, concise, and straightforward. Complex sentences, jargon, or convoluted explanations should result in a lower readability score.

Table 7: Prometheus Prompt Example for Readability Criteria

E.2 Correlation between Automatic Evaluation and Domain Expert Evaluation

Table 9 shows the individual across for correlation of each evaluation metric with human evaluation across LongT5, ChatGPT, and GPT-4. The low correlation coefficients are a consequence of the Spearman and Kendall methods, which analyze the linear and ordinal relationships between variables by comparing each set of scores. When these methods detect divergent scores between two evaluations, it leads to a reduced correlation coefficient, indicating a disproportion that is not apparent when considering the average scores alone.

Criteria	LongT5		ChatGPT		GPT-4	
	Spearman ρ	Kendall τ	Spearman ρ	Kendall τ	Spearman ρ	Kendall τ
BLEU	**0.459**	0.337	0.345	0.263	0.146	0.116
ROUGE-1	**0.435**	0.321	0.364	0.284	0.113	0.091
ROUGE-2	**0.462**	0.341	0.332	0.258	0.056	0.044
ROUGE-L	**0.433**	0.324	0.353	0.274	0.093	0.075
BERTScore_P	**0.457**	0.347	0.304	0.234	0.156	0.122
BERTScore_R	**0.466**	0.305	0.085	0.064	-0.022	-0.018
BERTScore_F1	**0.455**	0.332	0.246	0.192	0.097	0.077
G-Eval						
Usability	0.675	0.584	0.217	0.198	0.346	0.327
Relevance	0.569	0.499	0.339	0.304	0.325	0.306
Readability	0.208	0.181	0.395	0.373	0.139	0.137
Truthfulness	**0.726**	0.651	0.694	0.667	0.452	0.432
Prometheus						
Usability	**0.723**	0.675	0.386	0.351	0.516	0.495
Relevance	0.467	0.439	0.419	0.371	0.382	0.357
Readability	0.493	0.468	0.378	0.358	0.225	0.213
Truthfulness	0.541	0.521	0.439	0.402	0.454	0.427

Table 9: Correlations between Automated Metrics and Human Evaluation across Models

Evaluation and Continual Improvement for an Enterprise AI Assistant

Akash V. Maharaj, Kun Qian, Uttaran Bhattacharya, Sally Fang, Horia Galatanu, Manas Garg, Rachel Hanessian, Nishant Kapoor, Ken Russell, Shivakumar Vaithyanathan, and Yunyao Li

Adobe Inc.

{maharaj, kunq, ubhattac, xinf, horiag, mangarg, hanessia}@adobe.com

{niskapoor, kenrusse, vaithyan, yunyaol}@adobe.com

Abstract

The development of conversational AI assistants is an iterative process with multiple components. As such, the evaluation and continual improvement of these assistants is a complex and multifaceted problem. This paper introduces the challenges in evaluating and improving a generative AI assistant for enterprises, which is under active development, and how we address these challenges. We also share preliminary results and discuss lessons learned.

1 Introduction

Generative AI assistants for enterprises hold the great promise of significantly improved productivity, lowered barrier-to-entry, drastically increased product adoption, transformative amplification of creativity, and delivery of better customer and employee experiences (Kumar et al., 2023). Developing such an AI assistant is typically an iterative process, with its evaluation and continual improvement at the center.

Fig. 1 depicts the high-level architecture of Adobe Experience Platform AI Assistant[1] (Bhambhri, 2024), a generative AI assistant built for an enterprise data platform. As can be seen, it is a complex pipeline with multiple underlying components consisting of one or more machine learning models based on large language models (LLMs) or small language models (SLMs). Users interact with the system via a conversational interface to obtain answers based on heterogeneous data sources. The evaluation and continual improvement of such a system is a complex and multifaceted problem with the following key challenges.

Metrics. The success of `Assistant` is ultimately measured by metrics such as user engagement, user satisfaction, and user retention. However, such metrics are lag measures obtainable only after building and deploying `Assistant` in production. To guide continual improvement of `Assistant`, we also need to define metrics that are *lead* measures for various aspects of `Assistant` that are likely to impact the lag measures.

Data. To produce reliable evaluation metrics for `Assistant`, we need data that are both representative and high-quality. We need a systematic approach to obtain such high-quality data at scale.

Dynamics. As shown in Fig. 1, a real-world AI assistant usually consists of a complex pipeline of components. Each component evolves over time as both the underlying models and the assistant's functionalities change. Further, in enterprise settings, the distribution of questions asked is ever-changing as the customer base shifts and grows and existing customers mature in their adoption of the assistant. We need to consider such customer dynamics.

Human-Centered Design. The success of `Assistant` depends on both the capabilities of its underlying components and the user interface (UI) that surfaces those capabilities to support the overall user experience. As such, the evaluation and continual improvement for `Assistant` need to take all underlying components as well as UI into consideration for such a human-centered system (Liao and Vaughan, 2023).

Privacy and Security. Enterprise AI assistants like `Assistant` often deal with sensitive user data. We need to evaluate its performance while securely handling customer data and prevent unauthorized access or misuse (Wu et al., 2023; Yao et al., 2024).

The rest of this paper presents our proposed solution for addressing these changes. We also share our preliminary results and discuss lessons learned so far. Our main contributions include:

- A comprehensive continual improvement framework to support the evaluation and continual improvement for `Assistant`.
- A taxonomy of error types for error analysis and continual improvement.

[1] Hereafter referred to as `Assistant`

Proceedings of the Fifth Workshop on Data Science with Human-in-the-Loop (Language Advances), pages 17–24

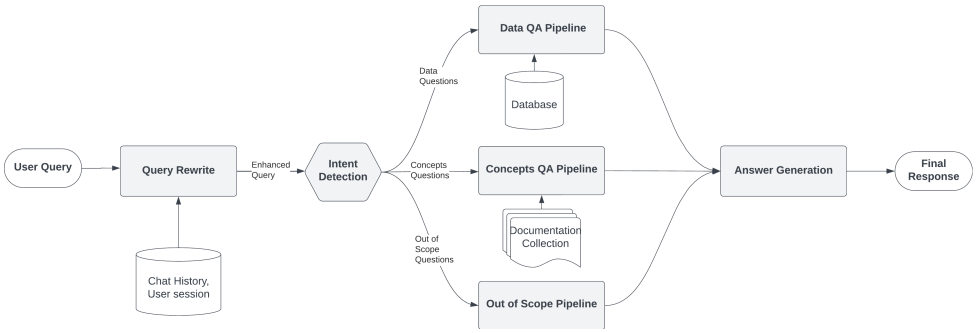

Figure 1: `Assistant` Overall Architecture

- Identifying the limitations of existing approaches on the evaluation of AI assistants.
- Highlighting the influential role of human-centered UI design in the evaluation and continual improvement of `Assistant`.
- Productionizing such a framework, sharing initial results and lessons learned.

2 Limitations of Existing Approaches

Common approaches for evaluating AI assistants include evaluation using explicit feedback, evaluation using implicit feedback, benchmarking (Liang et al., 2022), and human evaluation (Fernandes et al., 2023). Explicit feedback is collected from users through feedback buttons, direct prompts, or questions on their preferences. In contrast, implicit feedback is gathered from user actions within a system, such as clicks, views, or navigation patterns, providing insights into user behavior and preferences without requiring direct input. Evaluating with benchmark datasets is also a common way to evaluate AI assistants. These approaches, while important and effective to a certain degree, suffer from various limitations when it comes to evaluating an enterprise AI assistant such as `Assistant`, which is under active development and improvement.

2.1 Limitations of Explicit Feedback

Collecting explicit feedback from the users seems to be the most straightforward way to gauge user satisfaction and gather input to measure and improve the performance of an AI assistant. Table 1 illustrates the initial set of explicit feedback for `Assistant` from our early customers. We can observe several limitations of this approach.

Sparsity. Explicit user feedback is sparse. From Table 1, we can see that 76% of all customer interactions receive no explicit feedback at all. This sparsity issue makes it challenging to understand user experience and satisfaction comprehensively and hampers efforts to improve `Assistant`.

Representativeness. Since sharing explicit feedback is not mandatory, not every user does so. As shown in Table 1, users from two organizations shared no feedback at all. Further examination showed that most feedback came from a small number of users. In fact, about 30% of all the feedback originated from one user. Such a highly skewed feedback distribution may misrepresent the overall sentiment towards `Assistant`, and fail to reflect the diversity of users' experiences and opinions.

Lack of detailed feedback. partly due to minimizing user effort and partly because users only see the final response, explicit feedback is usually gathered via a simple UI form (*e.g.*, 👍, 👎 buttons). Unfortunately, feedback gathered this way often fails to capture the nuances of user experiences and preferences. For instance, a negative feedback indicating an incorrect final response is insufficient to pinpoint specific components for improvement. New approaches like showing step-by-step explanations and getting user feedback for the explanation are alternative ways to get detailed feedback from users and map them to specific components.

2.2 Limitations of Implicit Feedback

Implicit feedback has been extensively used in evaluating and improving intelligent systems (*e.g.*, Jawaheer et al. (2014); Koren et al. (2021)), and performance measurements of concrete tasks have been recommended as the best metric for evaluating natural language generation systems (Saphra et al., 2023). This approach has several limitations when evaluating AI assistants. First, since implicit feedback is obtained indirectly and passively from user actions, it may not always reflect users' true preferences. Prior work uses denoising techniques to prune the noisy interactions to avoid serious negative impact (Wang et al., 2021). In addition,

Table 1: Feedback type distribution and engagement ratio from different customers

Customer	Positive feedback	Negative feedback	No feedback
Org1	22.8%	16.2%	61%
Org2	12.6%	11.2%	76.2%
Org3	3.2%	24.9%	71.9%
Org4	2.7%	5.0%	92.3%
Org5	11.3%	5.2%	83.5%
Org6	5.6%	9.7%	84.7%
Org7	8.6%	21.4%	70%
Org8	15.4%	7.7%	76.9%
Org9	0%	0%	100%
Org10	0%	0%	100%
Total	**10.72%**	**13.12%**	**76.16%**

deriving implicit feedback from user interactions could be a challenge on its own. For instance, while meaningful implicit feedback is readily available for recommender systems in contexts such as online shopping (clicks, page views, add-to-cart, etc.), implicit signals available in AI assistants are less clearly related to concrete user goals. Specifically, users have a wide variety of goals, and the concrete tasks to achieve those goals are often very delayed.

2.3 Limitations of Off-the-Shelf Benchmarks

Although public benchmark datasets for general tasks are abundant (*e.g.*, Chang et al. (2023) lists 46 public benchmark datasets), they are often not applicable for domain-specific AI assistants. Creating domain-specific benchmark datasets is labor-intensive, time-consuming, and requires domain expertise. Moreover, assistants' workload and tasks may also evolve. Thus, there is no one static benchmark data that suits all (Mizrahi et al., 2024). Therefore, benchmark data creation itself is a continual process.

3 Our Approach

In this section, we introduce our framework to overcome the aforementioned challenges (Section 1) and limitations of existing approaches (Section 2) for evaluating an enterprise-grade AI assistant under active development.

3.1 Design Decisions

We first present a few key design decisions to balance the trade-offs to be made, both in terms of breadth and depth of any given type of evaluation.

Prioritize metrics directly impacted by production changes. The ultimate goal of `Assistant` is to improve the productivity and creativity of our users and lower barriers to entry. Since it takes time to materialize such lag measures, we focus on directly responsive "correctness" metrics, assuming a more correct `Assistant` will ultimately lead to positive downstream outcomes.

Align metrics with user experience. Not all errors are equal. The impact on the user experience of one incorrect citation in an otherwise correct answer is very different from that of a completely hallucinated answer. We aim to capture this nuance in the design of our error metrics.

Human Evaluation over automated evaluation. We believe that, despite challenges (Clark et al., 2021), human judgments are still best aligned with eventual user outcomes. As such, we prioritize human evaluation over automated evaluation. Once high-quality human judgments are collected, they can be used to validate which automatic evaluations are meaningful for specific tasks and components.

Efficient allocation of human evaluators. To conduct human evaluation at scale, we focus on the efficient allocation of human annotators. Specifically, simple annotation tasks are done by non-experts, while complex error analysis and the determination of how to make improvements are left up to engineers with domain expertise.

Collect both end-to-end metrics and component-wise metrics. We collect both individual and collective metrics to understand the overall quality of the system as well as which parts need to improve.

System-wide improvements. All components in `Assistant`, from ML/rule-based models, UI/UX components, to underlying data, may impact system performance. Therefore, instead of focusing solely on ML model improvements, we consider the entire "vertical" system holistically and leave no improvement off the table.

Prioritize human evaluation. Automated evaluation, which utilizes standard metrics and evaluation tools, is popular for its efficiency and objectivity (Chang et al., 2023). However, although more labor-intensive and time-consuming, manual evaluation by domain experts is more reliable in reflecting the final user impact. As such, we prioritize human evaluation over automation.

3.2 Severity-based Error Taxonomy

Designing metrics that align with our end users' judgments of the correctness and usefulness of `Assistant` is a complex task. We observed relatively high error rates from an early version of `Assistant` (over 50%), yet our users did not seem

Table 2: Error Severity Framework in Assistant

Category	Definition	Consequence	Examples
Severity 0	**Answer looks right, but is wrong**	***Erodes trust with the users***	- Convincing Concepts QA answers that are pure hallucinations - Incorrect Data QA answers that cannot easily be verified independently
Severity 1	**Answer looks wrong, user can't recover**	***Frustrates users***	- Failure to answer with generic error message - Answers with obvious logical inconsistencies, *e.g.*, mixing UI docs and API docs
Severity 2	**Answer looks wrong, user can recover**	***Annoys users***	- Misunderstood questions that user is able to rephrase and get correct answer - Incorrect out-of-scope question rejection that user is able to override

to perceive error rates to be this high in their self-reported surveys and regular feedback sessions. This discrepancy, consistent with the earlier observation that not all errors are the same (Freitag et al., 2021), led us to develop a *taxonomy* of errors.

To illustrate this point, consider the past two decades, where internet search has become a dominant (semi) natural language interface. In this domain, humans have become accustomed to certain classes of errors. When we do not get the desired results from a search engine, we rephrase and iterate till we find the answer. The initial failure of the search engine is annoying but generally tolerable unless we cannot find our answer even after many re-phrasings. At this point, we are left frustrated. Inspired by DevOps terminology (Kim et al., 2021), we can define two separate classes of errors: **Severity-2** ("Sev-2" for short) errors are annoying but repairable via rephrasing, while **Severity-1** ("Sev-1" for short) errors are not repairable.

Meanwhile, the rise of generative AI has introduced an entirely new class of error: answers that are convincing and look correct but are, in fact, wrong. Depending on the use case, these may be tolerable (or even desirable), but in the realm of enterprise assistants, these errors are troubling. They erode user trust and may lead to complete abandonment of the assistant. We term these **Severity-0** errors, "Sev-0" for short. Table 2 summarizes this severity-based error taxonomy, which has become an organizing principle for the evaluation and improvement of Assistant, as we discuss next.

3.3 Framework for Evaluation and Continual Improvement

Fig. 2 depicts our proposed evaluation and improvement framework. It includes three main components: Assistant, itself, a dedicated Annotation Tool, and a separate environment for Error Analysis. Human evaluation drives the evaluation and improvement of Assistant.

To ensure the efficient allocation of human resources, non-experts provide large-scale annotation of masked production data, while domain experts provide detailed error analysis on a sample of production data. For each annotation task, to ensure annotation quality, we design the UI and annotation guidelines iteratively with pilot study and improvements. We include training modules and exercises to ensure annotators meet a minimum bar of sufficient domain understanding. We assign multiple annotators for each annotation task to further ensure the annotation quality and conform to best practices (van der Lee et al., 2021).

We design different annotation tasks to assess the quality of different Assistant components and improvements needed. By collecting annotations based on prior interactions in the production system, we can generate both error metrics by severity (by comparing human labels to the choices the system made in production), *and* new golden-labeled data for model improvements.

Error analysis is a crucial step in gating improvement. At this step, domain experts — those with deep knowledge of how Assistant is designed — review samples of errors, identify error patterns, and determine specific improvements. These improvements take many potential forms, from prompt engineering to training and improving in-house models, to creating new templates and patterns for synthetic data, to more holistic changes such as improving the user experience or optimizing the specialized data indexes that are queried by Assistant, (for example, fine-tuning embeddings,

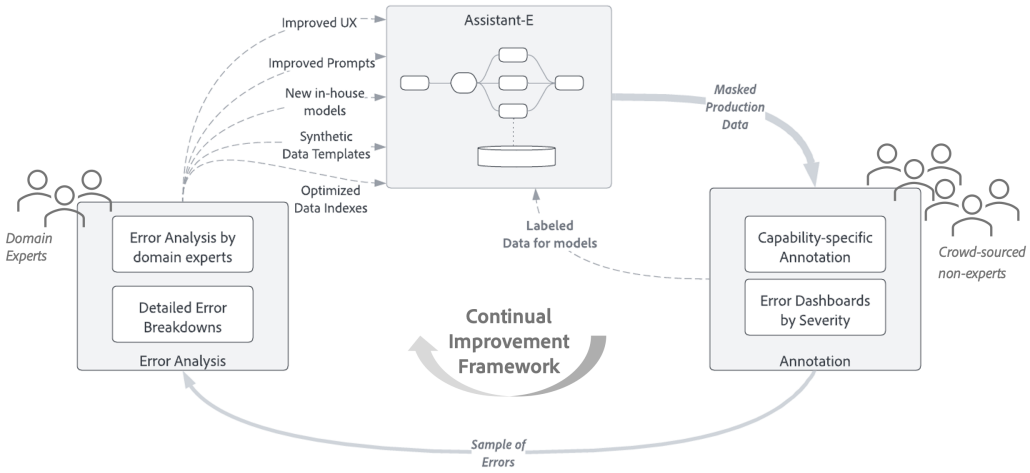

Figure 2: Evaluation and continual improvement framework of `Assistant`

or updating database schema). This last category of improvements is only possible when the application is viewed holistically, and all stakeholders are involved in error analysis.

4 Preliminary Results: Examples

While `Assistant` remains in active development, our evaluation and continual improvement framework already show promising impacts on both the prioritization and the design of improvements. In this section, we share the preliminary results obtained so far by examples.

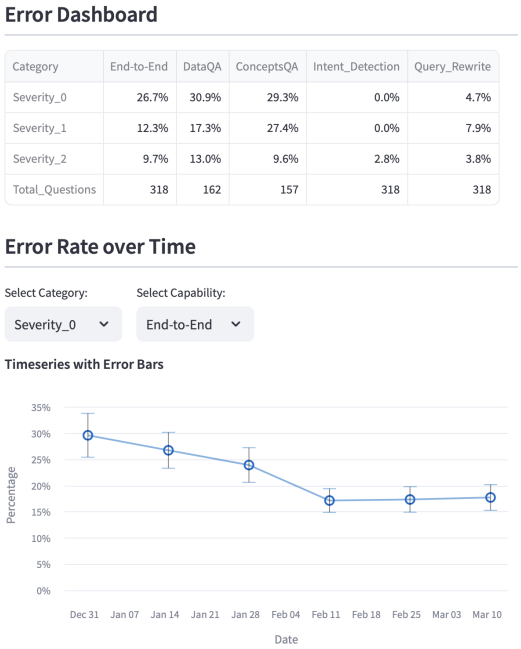

Figure 3: Dashboard showing snapshot of Error Severities and time-evolution for a single component. Illustrative data of similar magnitude to production numbers.

Fig. 3 illustrates an example error dashboard produced by the annotation tool, showing component-wise and end-to-end errors with further breakdown by severity levels as well as how they change over time. This dashboard is monitored by all stakeholders and is used to track the impact of feature releases and improvements. While ideas for improving `Assistant` may be endless, detailed error analysis allows the team to follow a powerful organizing principle: focusing on reducing error rates based on their actual impact on the users.

For instance, the example report in Table 3 shows that Out-of-Scope errors were our largest contributor to Sev-0 errors in Sprint 1. To address this, we introduced an Out-of-Scope text classifier using an in-house model, which achieved 90% precision and successfully reduced most such errors.

However, the new classifier also led to a new, particularly frustrating source of errors: in-scope questions misclassified as out-of-scope would no longer be answered. Without being able to quickly improve the classifier's precision, we used our other available lever of improvement and designed an override mechanism in the UI to allow users to receive an answer. As the Sprint 2 report shows, this UI change converted a potential Sev-1 error (refusal to answer) into a Sev-2 error (the user could now recover), showcasing how human-centered UI design allows holistic improvement of `Assistant`.

Explainability is important for improving user trust and comprehension. By helping users discover wrong answers with better explainability, we can reduce Sev-0 errors and move them to Sev-1/Sev-2 error buckets. We took a data-driven approach to choose from many applicable explainability tech-

Table 3: An example output of error analysis for Concept QA (illustrative data, real labels) from one sprint to the next after Out-Of-Scope detection was deployed.

Error Severity	% Sprint1	% Sprint2
Sev-0	53.4%	36.6%
OutOfScope	21.6%	6.2%
Hallucination	17.0%	16.4%
Doc-Retrieval	13.6%	14.0%
LLM-Error	1.1%	0.0%
Sev-1	46.6%	44.4%
Hallucination	36.4%	33.0%
Citation	5.7%	5.1%
LLM-Error	4.5%	6.3%
Sev-2	-	6.9%
OutOfScope (incorrect rejection)	-	6.9%

niques (Danilevsky et al., 2020). We first went through the Sev-0 queries obtained during a certain window and examined which technique(s) can be used to alleviate the severity of each error based on the potential overall impact of each explainability technique, its implementation difficulty, and human cognitive load. We created a decision matrix (Table 4) based on the analysis, and we focused on only 2 of the 7 options from (Li et al., 2024). As we move forward, we expect many more such informed improvements based on our framework.

5 Discussion

This framework has organically evolved during the development of `Assistant`. While many of the design choices laid out may seem obvious in hindsight, they were not as clear at the beginning of this project, and so it is worth discussing the lessons we have learned along the way.

First, we have found that metric design is of paramount importance. The severity framework came after many iterations in trying to connect enthusiastic early customer feedback with a seemingly large overall error rate. The insight that customers have varying tolerance depending on the class of errors has become a powerful organizing principle for our prioritization and resource allocation to improve `Assistant`.

Next, we have seen firsthand the benefits of building a decomposed system as opposed to depending on a single, monolithic model. The choice to decompose into multiple, orchestrating models was led by constraints such as task specialization and the need to query real-time data. We have also reaped the secondary benefit of having many available "levers of improvement" (prompts, in-house models, specialized indexes, UX improvements, etc.), many more than what is possible in a single language model paradigm.

Table 4: Decision matrix for explainability techniques

Explainability techniques	Potential impact	Engineering difficulty	Congitive load
technique1	0.0%	high	low
technique2	8.6%	high	high
technique3	48.6%	low	low
technique4	88.6%	medium	medium
technique5	20.0%	high	low
technique6	100%	medium	low
technique7	74.3%	high	low

Finally, iterative and agile development are more important than designing everything upfront and building specialized tools. For instance, while it is tempting to build in-house tools, using spreadsheets as a simple alternative initially allows us to learn important lessons on designing the annotation tasks, from annotation guidelines to the actual UI.

6 Future Work

As we continue to develop `Assistant` and onboard more customers, we plan to extend our evaluation and continual improvement framework with more human-in-the-loop/LLM-in-the-loop automation to scale our evaluation and error analysis processes (Zheng et al., 2023). In addition, the current framework heavily focuses on retrospective analysis based on *past* customer interactions. We plan to extend it with more proactive user studies and evaluation of in-development functionalities. Moreover, personalization is also important for enterprise AI assistants since we have customers with different technical levels. To provide the best experience to various personas in potentially different languages, additional evaluation metrics and datasets proposed in (Jadeja and Varia, 2017; Ahuja et al., 2023) may also be considered. As we have emphasized, human-centered design is essential for the success of `Assistant`. We plan to further explore how the deeper interplay between ML and UX components in this new paradigm of HCI can lead to more explainable and accurate assistants. Finally, the impact of generative AI applications in the workplace is an important new area of study (Brynjolfsson et al., 2023). As we enroll new customers, we intend to run A/B tests (Hussey and Hughes, 2007) that assess the causal impact of `Assistant` on the engagement and productivity of customers.

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, et al. 2023. Mega: Multilingual evaluation of generative ai. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267.

Anjul Bhambhri. 2024. New ai integrations in adobe experience platform. https://business.adobe.com/blog/the-latest/new-ai-assistant-in-adobe-experience-platform. Accessed: 2024-04-23.

Erik Brynjolfsson, Danielle Li, and Lindsey R Raymond. 2023. Generative ai at work. Technical report, National Bureau of Economic Research.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. 2021. All that's 'human' is not gold: Evaluating human evaluation of generated text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7282–7296, Online. Association for Computational Linguistics.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable AI for natural language processing. *CoRR*, abs/2010.00711.

Patrick Fernandes, Aman Madaan, Emmy Liu, António Farinhas, Pedro Henrique Martins, Amanda Bertsch, José G. C. de Souza, Shuyan Zhou, Tongshuang Wu, Graham Neubig, and André F. T. Martins. 2023. Bridging the Gap: A Survey on Integrating (Human) Feedback for Natural Language Generation. *Transactions of the Association for Computational Linguistics*, 11:1643–1668.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Michael A Hussey and James P Hughes. 2007. Design and analysis of stepped wedge cluster randomized trials. *Contemporary clinical trials*, 28(2):182–191.

Mahipal Jadeja and Neelanshi Varia. 2017. Perspectives for evaluating conversational AI. *CoRR*, abs/1709.04734.

Gawesh Jawaheer, Peter Weller, and Patty Kostkova. 2014. Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Trans. Interact. Intell. Syst.*, 4(2).

Gene Kim, Jez Humble, Patrick Debois, John Willis, and Nicole Forsgren. 2021. *The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations*. IT Revolution.

Yehuda Koren, Steffen Rendle, and Robert Bell. 2021. Advances in collaborative filtering. *Recommender systems handbook*, pages 91–142.

Akit Kumar, M.S. Lakshmi Devi, and Jeffrey S. Saltz. 2023. Bridging the gap in ai-driven workflows: The case for domain-specific generative bots. In *2023 IEEE International Conference on Big Data (BigData)*, pages 2421–2430.

Yunyao Li, Dragomir R. Radev, and Davood Rafiei. 2024. *Natural Language Interfaces to Databases*. Springer Nature.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Q Vera Liao and Jennifer Wortman Vaughan. 2023. Ai transparency in the age of llms: A human-centered research roadmap. *arXiv preprint arXiv:2306.01941*.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? a call for multi-prompt llm evaluation.

Naomi Saphra, Eve Fleisig, Kyunghyun Cho, and Adam Lopez. 2023. First tragedy, then parse: History repeats itself in the new era of large language models. *arXiv preprint arXiv:2311.05020*.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Krahmer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language*, 67:101151.

Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising implicit feedback for recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 373–381.

Xiaodong Wu, Ran Duan, and Jianbing Ni. 2023. Unveiling security, privacy, and ethical concerns of chatgpt. *Journal of Information and Intelligence*.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, page 100211.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

Mini-DA: Improving Your Model Performance through Minimal Data Augmentation using LLM

Shuangtao Yang*, **Xiaoyi Liu***, **Xiaozheng Dong***, **Bo Fu**
Lenovo Knowdee (Beijing) Intelligent Technology Co., Ltd., Beijing, China
{yangst, liuxy, dongxz, fubo}@knowdee.com

Abstract

When performing data augmentation using large language models (LLMs), the common approach is to directly generate a large number of new samples based on the original dataset, and then model is trained on the integration of augmented dataset and the original dataset. However, data generation demands extensive computational resources. In this study, we propose Mini-DA, a minimized data augmentation method that leverages the feedback from the target model during the training process to select only the most challenging samples from the validation set for augmentation. Our experimental results show in text classification task, by using as little as 13% of the original augmentation volume, Mini-DA can achieve performance comparable to full data augmentation for intent detection task, significantly improving data and computational resource utilization efficiency.

1 Introduction

Data is the lifeblood of deep learning models, and the availability of high-quality data is crucial for achieving strong model performance. However, acquiring such data can be a challenge, particularly in scenarios where data is limited or unavailable. Moreover, human annotation, a common method for obtaining labeled data, is known to be financially expensive and time-consuming. As such, data augmentation techniques have become increasingly important, especially in scenarios where data is limited.

Data augmentation has been studied for a long time in various domains, with rule-based method, data interpolation techniques, and model based approaches explored (Feng et al., 2021; Hedderich et al., 2021). While these traditional data augmentation methods have shown effectiveness, the rapidly evolving field of large language models (LLMs) has ushered in a new era of augmentation methods for natural language processing tasks. With their remarkable ability to generate human-like text, LLMs have enabled generative data augmentation techniques that can create more diverse and realistic synthetic samples, potentially leading to improved model performance. However, as highlighted in the comprehensive survey by (Ding et al., 2024), the generation of extensive augmented datasets can cause significant expenses due to the demands of considerable computational resources, especially for SOTA models.

To address the limitation of data augmentation with LLMs, we propose Mini-DA, a novel framework that aims to maximize the benefits of LLM-based data augmentation while minimizing the associated costs. The key innovation of Mini-DA lies in its ability to leverage the prediction result of the target model on the validation set during k-fold cross-validation to identify "challenging samples" that the model struggles to predict correctly. Then, for these difficult samples, a instruction-tuned large language model is used to generate synthesized data based on a given query and its label. This process is repeated iteratively, collecting augmented data until the model's performance on the test set stabilizes. Through our experiments on two datasets for the intent detection task, we demonstrate that by focusing augmentation efforts on a limited number of difficult samples, Mini-DA significantly reduces the augmentation volume compared to full data augmentation, leading to substantial savings in computational resources while still producing comparable performance.

2 Related Work

2.1 Pre-LLM Data Augmentation

Data augmentation has been widely studied before the advent of LLMs. Various approaches were in-

*These authors contributed equally to this work

Proceedings of the Fifth Workshop on Data Science with Human-in-the-Loop (Language Advances), pages 25–30

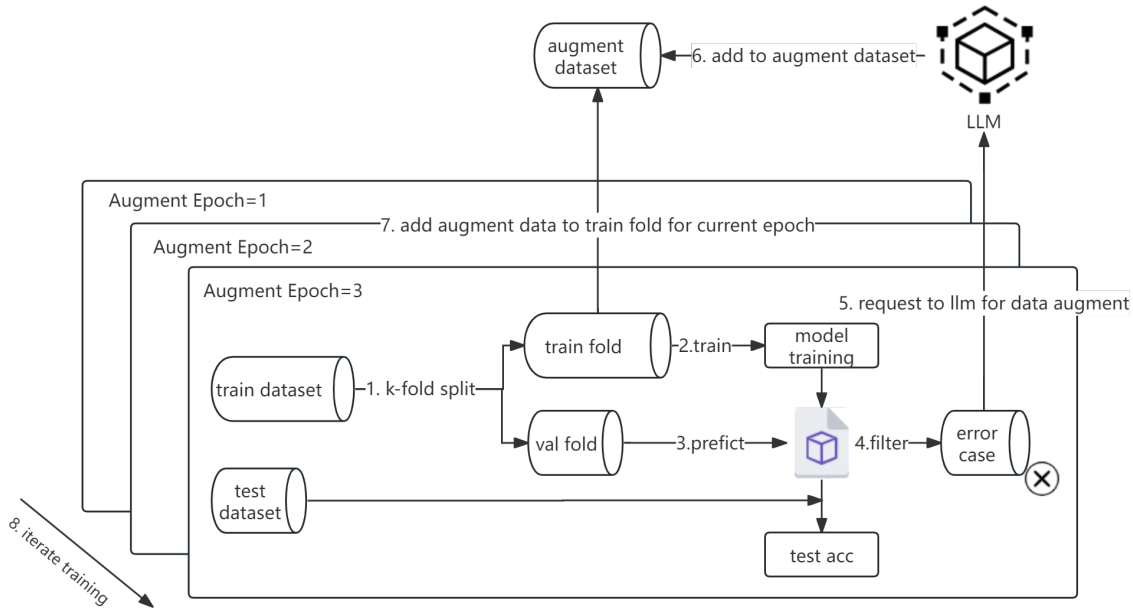

Figure 1: **Mini-DA framework.** The figure shows the iterative augmentation process. (1) One iteration begins by splitting the original dataset into k-folds. (2) Models are then trained on the training folds. (3) The trained models are evaluated on the validation folds, and (4) error cases are selected. (5) A LLM is instructed to perform data augmentation on the selected error samples. (6) The augmented data generated by the LLM is added to the augmented dataset. (7) In the next epoch, the dataset is re-split, and any existing augmented data corresponding to samples in the new training folds is integrated.

vestigated, including rule-based methods like Easy Data Augmentation (EDA) proposed by Wei and Zou (2019). EDA introduced token-level operations such as random insertion, deletion, and swapping. At sentence-level data augmentation, paraphrasing is widely adopted. The most popular one is Backtranslation (Sennrich et al., 2016), which uses Seq2seq and language models to translate a sequence into another language and then back into the original language.

2.2 Data Augmentation with LLMs

With the emergence of Large Language Models (LLMs), data augmentation techniques have undergone significant refinement and innovation. LLMs possess capabilities for generating high-quality, diverse, and contextually relevant text, enabling novel approaches to data augmentation.

One of the most common data augmentation method employing LLMs is to use them as data generators. Chintagunta et al. (2021) utilize powerful models such as GPT-3 to synthesize medical dialogue summaries. By training models on a combination of synthesized and human-annotated data, their approach effectively scales a small set of human-annotated examples to achieve performance comparable to using a significantly larger human-annotated dataset. Møller et al. (2024) employs LLMs to generate examples for specific labels in low-resource classification scenarios, by providing an example and its corresponding label. Lin et al. (2023) uses instruction tuned LLM, GPT-3.5, to generate examples within the context of the training set and subsequently filtered out unhelpful examples. For intent detection task, Sahu et al. (2022) introduces a prompting-based data augmentation using GPT-3, , and demonstrates its effectiveness in improving classifier performance, especially when combined with filtering techniques to address challenges in generating data for closely-related intents.

Another common approach involves using LLMs to reformulate existing data to more diverse variations. These techniques are proved to be particularly valuable in tasks like counterfactual generation, where existing data is transformed into its counterfactual version. For instance, Chen et al. (2023) employs LLMs to generate high-quality counterfactual data on a large scale. CORE (Dixit et al., 2022) also uses GPT-3 for retrieval-augmented generation (RAG), generating counter-

factual edits conditioned on retrieved excerpts from the input. These perturbations serve to reduce model bias and enhance performance.

3 Method

In the following section, we describe our proposed iterative LLM-in-the-loop data augmentation approach Mini-DA, as illustrated in Figure 1. At each iteration, we leverage the feedback from the target model to identify difficult examples from validation set and instruct LLM to only augment these selected samples.

The Mini-DA process can be broken down into the following steps:

1. **Dataset Splitting** If the original dataset does not come with a predefined test set, we first split a portion of the data to create a held-out test set. This test set will be used for monitoring the model's performance and determining convergence during the iterative process. The remaining dataset is then split into k folds. This process employs stratified sampling to ensure that both sets are representative of the underlying data distribution.

2. **Model Training** The target model is trained k times, using one different fold for validation and the remaining k-1 folds are combined for training each time.

3. **Validation Set Prediction** After training, the best models saved from the training stage are then evaluated on their own validation set.

4. **Challenging Case Collection** Error case from prediction of each fold on validation set are collected and identified as challenging samples

5. **Selective Data Augmentation** The prediction errors are then input to an LLM with predefined data augmentation prompt, and obtain a set of synthetic examples as augmented data.

6. **Augmented Dataset** Data generated from last step is then added to the augmented dataset, and we maintain a augmented dataset mapping between each original sample and its corresponding augmented data for future use.

7. **Augmented Data Integration** For the next augmenting epoch, the dataset is re-split into new k folds. And k new training and validation set pairs are formed. Before training on the new training sets, we check if any samples in each new training set have corresponding augmented data in augmented dataset. If so, we incorporate those augmented samples into each training set. Each training set should only contain augmented samples that are generated from original data it contains.

8. **Iterative Process** Steps 2 through 7 are repeated for a predetermined number of epochs or until a convergence criterion is met, which is typically when the model's performance on a held-out test set stops improving across a predetermined number of epochs. At this point, the augmentation of the original dataset is completed.

a.
You are an experienced data annotator. Please generate five user questions following the requirements below.
1. Focus on the "banking" domain;
2. Should focus on "**{intent_label}**" intent, which represent **{intent_definition}**;
3. The newly generated sentence needs to be semantically similar to sentence: "**{query}**";

b.
You are an experienced data annotator. Please generate five user questions following the requirements below.
1. Focus on the "**{domain_label}**" domain;
2. Should focus on "**{intent_label}**" intent, which represent **{intent_definition}**;
3. The newly generated sentence needs to be semantically similar to sentence: "**{query}**";
4. Newly generated sentences need to be in Chinese;

Figure 2: The prompts used to generate augmented data for a. banking77 dataset and b. ECDT-NLU-2019 dataset

4 Experiments Setup

4.1 Datasets and Task

To verify the effectiveness of our approach, we conduct experiments on two intent detection datasets, including banking77 (Casanueva et al., 2020) and ECDT-NLU-2019[1].

[1] http://conference.cipsc.org.cn/smp2019/evaluation.html

The original banking-77 is an English dataset in the banking domain, which includes 10,003 training and 3,080 test cases labeled with 77 intent. Since our primary focus is on enhancing the model performance in data limited scenario, we sampled a subset from banking77 for our experiments. We will refer the sampled dataset as banking77-filtered in this paper. Banking77-filtered includes 2,047 training and 693 test cases, which still has 77 intent labels.

The original ECDT-NLU-2019 is a Chinese natural language understanding dataset consisted of multiple tasks, including domain classification, intent detection, and slot filling. We only considered the intent detection task in our experiments. This datasets comprises 2,061 training and 516 test cases with 45 intent labels.

4.2 Models

Since the two datasets we used for our experiments are in different languages, we selected bert-base-multilingual-uncased[2] (Devlin et al., 2018) as our base model for training and prediction.

We use GPT-3.5 Turbo as the large language model to generate augmented dataset. The prompts used to augmented each dataset is illustrated in Figure 2.

4.3 Implementation Details

During the data splitting step, we set $k = 5$ for 5-fold cross-validation. In each augmenting epoch, we train bert-base-multilingual-uncased for 30 training epochs with a batch size of 64, learning rate of $2e-5$ and the Adam optimizer (Kingma and Ba, 2017).

The stopping criterion for the iterative augmenting process is set to the average accuracy stop improving on test set for 2 consecutive augmenting epochs. For both datasets, we run the augmenting process for a maximum of 10 epochs.

4.4 Baseline Methods

We compare our proposed method with two baseline methods. It is important to note that our primary focus is on proposing an efficient framework for data augmentation by contrasting full-dataset augmentation with selective augmentation. Therefore, we include a basic prompt-based data augmentation method using a LLM as our baseline. However, the augmentation component (step 5) in our framework is modular and can be modified to other augmentation methods with LLM depending on the specific use case.

1. Baseline 1: We performed 5-fold cross-validation on the same base model, bert-base-multilingual-uncased, using the original, unaugmented training sets and the same hyperparameters in 4.3.

2. Baseline 2: we performed full data augmentation by generating augmented samples for every instance in the training set using GPT-3.5 Turbo with the prompts specified in Figure 2. We then conducted 5-fold cross-validation, where for each fold, the augmented data generated from that fold's training set was integrated into the corresponding fold's training set, ensuring no data augmented from the validation fold was trained on. The same hyperparameters in 4.3 were employed.

4.5 Evaluation Metrics

Considering the imbalanced class distribution present in the two selected datasets, we utilized accuracy as the evaluation metric to assess and compare the model performance across all methods on both datasets.

5 Result and Analysis

In this section, we present the experimental results obtained by evaluating our proposed Mini-DA method and the two baseline approaches on the selected datasets. We report and analyze the performance of each method in terms of the average accuracy of 5-fold cross-validation. Results are shown in Figure 3, Table 1, Figure 4, and Table 2.

For banking77-filter dataset, results shown in Figure 3, the average accuracy on test set of models trained on the original dataset achieved 80.52% (the dotted green line), while average accuracy models trained on the fully augmented dataset reaches 86.41% (the green line), representing a 5.89% improvement from unaugmented baseline. The red line represents the average accuracy on the test set when using the Mini-DA framework for training set augmentation crossing augmentation epochs. At the second augment epoch, Mini-DA achieved an average accuracy of 86.64%, which is even 0.23% higher than the result obtained using the fully augmented dataset, despite only augmenting 24% of the training data. When progressing to the fifth

[2] https://huggingface.co/google-bert/bert-base-multilingual-uncased

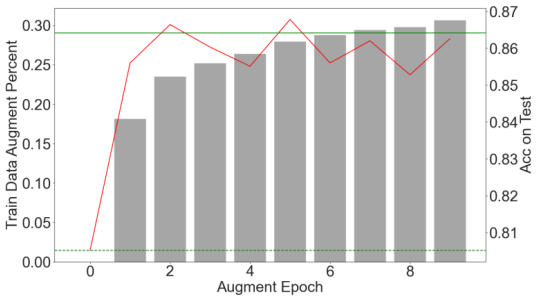

Figure 3: Accuracy on the banking77-filtered test set for the Mini-DA approach (red line) compared to the fully augmented dataset (green line at top) and the original, unaugmented dataset (dotted green line at bottom) across augmentation epochs. The bars indicate the sum of total augmented data added to the training set at each epoch.

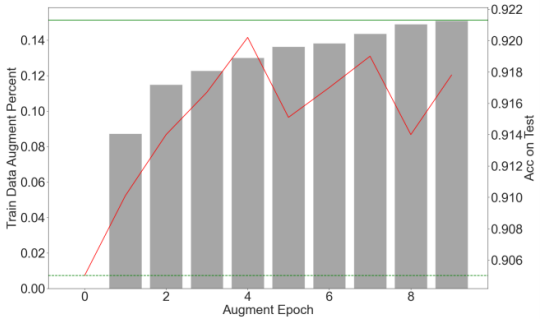

Figure 4: Accuracy on the ECDT-NLU-2019 test set for the Mini-DA approach (red line) compared to the fully augmented dataset (green line at top) and the original, unaugmented dataset (dotted green line at bottom) across augmentation epochs. The bars indicate the sum of total augmented data added to the training set at each epoch.

Method	**Augment Epoch**	**Number of Augmented Data added to Training sets**	**Average ACC on Test set**
No Augmentation		0	0.8052
Full Augmentation		2047	0.8641
Mini-DA	1	372	0.8560
	2	482	0.8664
	3	516	0.8603
	4	541	0.8551
	5	573	**0.8678**
	6	590	0.8560
	7	603	0.8620
	8	610	0.8528
	9	628	0.8626

Table 1: Results of banking77-filter

Method	**Augment Epoch**	**Number of Augmented Data added to Training sets**	**Average ACC on Test set**
No Augmentation		0	0.9050
Full Augmentation		2061	0.9213
Mini-DA	1	180	0.9101
	2	237	0.9140
	3	253	0.9167
	4	268	0.9202
	5	281	0.9151
	6	285	0.9170
	7	296	0.9190
	8	307	0.9140
	9	311	0.9178

Table 2: Results of EDTC-NLU-2019

augment epoch, Mini-DA achieved its optimal performance while a total of 573 training data points were augmented, accounting for 28% of the training set. On the test set, the average ACC reached 86.78%, an improvement of 0.37% compared to the average accuracy using the full augmented data.

On the EDTC-NLU-2019 dataset (shown by Figure 4), we observed similar phenomena. At the fourth augment epoch, the average accuracy on the test set reached 92.02%, which is only 0.11% lower than the result obtained using the fully augmented dataset (92.13%). However, at this point, Mini-DA only augmented 13% of the training data, resulting in a reduction of 1793 GPT-3.5 Turbo requests compared to the full data augmentation approach.

Through these experiments, we can observe that for intent detection scenarios with low resources, Mini-DA effectively combines two stages: fine-tuning on the target model and data augmentation using a LLM. By employing a cross-validation approach to selectively augment difficult samples from the validation set, Mini-DA avoids unnecessary augmentation of correctly predicted samples in the training set, thereby reducing the cost of data augmentation.

6 Conclusion

In this work, we present Mini-DA to efficiently augment intent detection data with LLMs. We design a iterative LLMs-in-the-loop framework that incorporates feedback from fine-tuning stage of target model to generate an augmented dataset. The

results demonstrate that with as little as 13% of the augmented data generated, we can achieve comparable performance to full data augmentation on intent detection task in data-limited scenarios. Overall, Mini-DA presents a promising solution for data augmentation which significantly reducing computational costs and improving data efficiency.

For future work, our plan involves conducting comprehensive experiments across various tasks, including but not limited to question answering, text generation, and text retrieval. We believe this approach can be effective in improving model performance across a wide range of tasks in a data-efficient manner.

References

Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on NLP for ConvAI - ACL 2020*. Data available at https://github.com/PolyAI-LDN/task-specific-datasets.

Zeming Chen, Qiyue Gao, Antoine Bosselut, Ashish Sabharwal, and Kyle Richardson. 2023. DISCO: Distilling counterfactuals with large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5514–5528, Toronto, Canada. Association for Computational Linguistics.

Bharath Chintagunta, Namit Katariya, Xavier Amatriain, and Anitha Kannan. 2021. Medically aware GPT-3 as a data generator for medical dialogue summarization. In *Proceedings of the Second Workshop on Natural Language Processing for Medical Conversations*, pages 66–76, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. 2024. Data augmentation using llms: Data perspectives, learning paradigms and challenges. *Preprint*, arXiv:2403.02990.

Tanay Dixit, Bhargavi Paranjape, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. CORE: A retrieve-then-edit framework for counterfactual data generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2964–2984, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. *Preprint*, arXiv:2105.03075.

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Yen-Ting Lin, Alexandros Papangelis, Seokhwan Kim, Sungjin Lee, Devamanyu Hazarika, Mahdi Namazifar, Di Jin, Yang Liu, and Dilek Hakkani-Tur. 2023. Selective in-context data augmentation for intent detection using pointwise V-information. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1463–1476, Dubrovnik, Croatia. Association for Computational Linguistics.

Anders Giovanni Møller, Jacob Aarup Dalsgaard, Arianna Pera, and Luca Maria Aiello. 2024. The parrot dilemma: Human-labeled vs. llm-augmented data in classification tasks. *Preprint*, arXiv:2304.13861.

Gaurav Sahu, Pau Rodriguez, Issam H. Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data augmentation for intent classification with off-the-shelf large language models. *Preprint*, arXiv:2204.01959.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

CURATRON: Complete and Robust Preference Data for Rigorous Alignment of Large Language Models

Son The Nguyen
University of Illinois Chicago
Chicago, Illinois, USA
snguye65@uic.edu

Niranjan Uma Naresh
Independent
Kirkland, Washington, USA
un.niranjan@gmail.com

Theja Tulabandhula
University of Illinois Chicago
Chicago, Illinois, USA
theja@uic.edu

1 Introduction

Large Language Models (LLMs) are highly advanced Artificial Intelligence (AI) systems capable of understanding, interpreting, and generating languages. The integration of AI chatbots like ChatGPT into our daily lives and businesses has had a profound impact on both society and industries (Eloundou et al., 2023). However, the success of GPTs/LLMs depends not only on their ability to generate responses and perform tasks well but also on their alignment with human values and expectations.

The prevalent method for aligning AI/LLMs currently involves preference learning (PL) through human feedback. However, gathering human feedback is slow and expensive and often results in incomplete or imperfect data (Bai et al., 2022; Lee et al., 2023). Furthermore, participants may intentionally provide inaccurate or harmful feedback due to malicious intentions, as pointed out by (Casper et al., 2023). These factors can lead to unintended consequences in estimating rankings from preference datasets from models such as BTL. They pose a considerable challenge in ensuring the integrity and reliability of the preference datasets used for aligning LLMs, especially when scaling up the alignment process with large-scale responses and participants.

Approaching the issues, we consider the following learning problem: Suppose there are n responses we wish to order based on a notion of comparison, between every pair of responses, with probabilistic outcomes. Further, we are given a set, $\aleph = \{(i, j, \{y_{ij}^k\})\}$, consisting of K independent pairwise comparison outcomes, denoted by $\{y_{ij}^k\} \in \{0, 1\}, k \in [K]$, between pairs of responses $(i, j) \subseteq [n] \times [n]$, a significant proportion of which might be corrupted by an adversary.

In this passive learning setting, our contributions are as follows. We give a generic definition of (additive) adversarial noise and show that if it is not accounted for, the quality of the estimated ranking can be quite poor. To address this, we develop an efficient and correct ranking method called Robust Preference Data for Rigorous Alignment (RORATRON), which is robust against adversarial noise. Under certain assumptions, we prove that our method guarantees high-probability learnability with a small margin of error. We also devise a method called Complete Robust Preference Data for Rigorous Alignment (CURATRON) to handle the scenario where not all pairs are compared, and the observed pairwise data is adversarially corrupted.

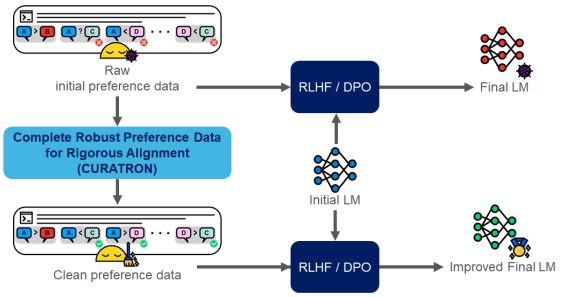

Figure 1: CURATRON corrects incomplete and adversarially corrupted preference data to improve RLHF/DPO alignment results compared to using the raw initial preference data.

2 Related Work

LLM Alignment with PL from human feedback: PL was initially developed to train agents in simulated environments to perform nuanced behaviors that are hard to define but easy to observe and recognize (Christiano et al., 2017). It has recently been found successful in aligning LLMs to human intentions and values such as harmfulness, helpfulness, factuality, and safety. Some of the methods of PL in LLMs are RLHF (Ouyang et al., 2022), RLAIF (Bai et al., 2022; Lee et al., 2023), DPO/ψPO (Rafailov et al., 2023; Tunstall et al., 2023; Zhao et al., 2023),

Proceedings of the Fifth Workshop on Data Science with Human-in-the-Loop (Language Advances), pages 31–39

and SLiC-HF (Zhao et al., 2023).

Ranking Models: In the BTL model, item i has an associated score w_i; then, the probability that item i is preferred over j is given by $P_{ij} = e^{-w_i}/(e^{-w_i} + e^{-w_j})$ where $\mathbf{w} \in \mathbb{R}^n$ is the BTL parameter vector to be estimated from data; here, $\mathbf{P} \in \mathbb{R}^{n \times n}$ is called the 'preference matrix'. A closely related model, in the non-active setting, is the recently proposed LR model (Rajkumar and Agarwal, 2016) wherein a generic class of preference matrices is characterized to be those having low rank under transformations using certain functions; specifically, for BTL-like models, the logit function defined as $\psi(x) = \log(x/(1-x))$ turns out to right choice as shown in their paper. However, while their model accounts for missing information, they do not consider the harder problem of handling adversarial noise.

Robust Subspace Recovery: The Robust PCA (RPCA) problem (Netrapalli et al., 2014) addresses the following question: suppose we are given a data matrix $\mathbf{M}$ which is the sum of an unknown low-rank matrix $\mathbf{L}$ and an unknown sparse matrix $\mathbf{S}$, can we recover each of the component matrices? While several works (Yi et al., 2016; Hsu et al., 2011) analyze this problem, it is shown in (Netrapalli et al., 2014) that, under information-theoretically tight assumptions, a simple iterative algorithm based on non-convex alternating projections of appropriate residuals provably yields an ϵ-accurate solution in $O(\log(1/\epsilon))$ iterations with an overall computational complexity of $O(n^2 r^2 \log(1/\epsilon))$ where r is the rank of $\mathbf{L}$. We will use this result, in particular, to derive guarantees for our ranking problem.

3 Problem Setup

3.1 Notation

We denote the set of all permutations of n LLM responses/items as $\mathcal{S}_n$. If not specifically defined, we use lower-case letters for scalars, upper-case letters for global constants, lower-case bold-face letters for vectors and upper-case bold-face letters for matrices; specifically, $\mathbf{P}$ denotes a preference matrix. Let $\mathcal{P}_n := \{\mathbf{P} \in [0,1]^{n \times n} | P_{ij} + P_{ji} = 1\}$ denote the set of all pairwise preference matrices over n responses. Let the set of stochastic-transitive matrices be $\mathcal{P}_n^{ST} := \{\mathbf{P} \in \mathcal{P}_n | P_{ij} > 1/2, P_{jk} > 1/2 \implies P_{ik} > 1/2\}$. Let the set preference matrices described by the BTL model be $\mathcal{P}_n^{BTL} := \{\mathbf{P} \in \mathcal{P}_n | \exists \mathbf{w} \in \mathbb{R}^n \text{ s.t. } e^{-w_i}/(e^{-w_i} + e^{-w_j})\}$. Let $\psi : [0,1] \mapsto \mathbb{R}$ be a strictly increasing bijective L-Lipschitz function and define the class of low-rank preference matrices with respect to ψ as $\mathcal{P}_n^{LR(\psi,r)} = \{\mathbf{P} \in \mathcal{P}_n | \text{rank}(\psi(\mathbf{P})) \leq r\}$ where $r \in [n]$; when we apply such a transformation to a matrix, it is applied entry-wise. In this paper, we take ψ to be the logit function.

For any matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$, let the infinity norm be denoted by $\|\mathbf{M}\|_\infty = \max_{i,j} |M_{ij}|$, the Frobenius norm be denoted by $\|\mathbf{M}\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^n M_{ij}^2}$, the spectral norm be denoted by $\|\mathbf{M}\|_2 = \max_{\mathbf{x},\mathbf{y} \in \mathbb{R}^n} \mathbf{x}^\top \mathbf{M} \mathbf{y}$. Denoting the indicator function by $\mathbb{1}$, define the zero norm of a matrix to be the maximum number of non-zero elements in any row/column, ie, $\|\mathbf{M}\|_0 = \max(\max_j \sum_{i=1}^n \mathbb{1}(M_{ij} \neq 0), \max_i \sum_{j=1}^n \mathbb{1}(M_{ij} \neq 0))$. Let the Singular Value Decomposition (SVD) of a square matrix be given by $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ where $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{n \times r}$ are orthonormal matrices (whose columns are singular vectors) and $\mathbf{\Sigma} \in \mathbb{R}^{r \times r}$ is the diagonal matrix of singular values. Now, $\mathbf{M}$ is said to be μ-incoherent if $\max\left(\max_i \left\|\mathbf{e}_i^\top \mathbf{U}\right\|_2, \max_i \left\|\mathbf{e}_i^\top \mathbf{V}\right\|_2\right) \leq \mu\sqrt{r/n}$ where $\mathbf{e}_i$ denotes the i^{th} basis vector in $\mathbb{R}^n$. Also, let $\sigma_{\max} := \max_i \Sigma_{ii}$ and $\sigma_{\min} := \min_i \Sigma_{ii}$.

We define the distance between a permutation $\sigma \in \mathcal{S}_n$ and a preference matrix $\mathbf{P} \in \mathcal{P}_n$ as:

$$\text{dist}(\sigma, \mathbf{P}) := \binom{n}{2}^{-1} \sum_{i<j} \mathbb{1}\left((P_{ij} > 1/2) \wedge (\sigma(i) \succ \sigma(j))\right) + \binom{n}{2}^{-1} \sum_{i<j} \mathbb{1}\left((P_{ji} > 1/2) \wedge (\sigma(j) \succ \sigma(i))\right)$$

Note that the above loss function basically is the number of pairs on which the ordering with respect σ and $\mathbf{P}$ differ divided by the number of ways to choose two out of n responses. Finally, let $P_{\min} = \min_{i \neq j} P_{ij}$ and $\Delta = \min_{i \neq j} |\psi(P_{ij}) - \psi(1/2)|$.

3.2 Characterization of the Adversary

The following (weak) assumption characterizes the properties of the adversary.

Assumption 1. *The (additive) adversarial noise which corrupts a μ-incoherent preference matrix $\mathbf{P} \in \mathcal{P}_n^{LR(\psi,r)}$ is modeled by a skew-symmetric sparse matrix $\mathbf{S}$ so that the corrupted preference matrix $\mathbf{P^c} \in \mathcal{P}_n$ is given by $\mathbf{P^c} = \mathbf{P} + \mathbf{S}$. We assume the (deterministic) bounded degree condition that $\|\mathbf{S}\|_0 \leq d < n$ such $d < n/512\mu^2 r$ where $r \leq n$.*

So, why do existing non-robust algorithms not recover the true response ordering in the presence

of an adversarial noise source? This question is answered by the following proposition, which precisely quantifies how bad a ranking could be when an algorithm uses the corrupted pairwise preference matrix. The key idea is to construct an adversary that intentionally flips true comparison results.

Claim 1 (Upper bound on estimation error). *Under Assumption 1 it is possible that* $\mathrm{dist}(\widehat{\sigma}, \mathbf{P^c}) = O(1)$.

Proof. Assume that we are exactly given the entries of the preference matrix as opposed to sampling them. Note that in order to estimate a ranking from a given preference matrix, we still need to use a pairwise ranking procedure. Let $\widehat{\sigma} \in \mathcal{S}_n$ be the output of any Pairwise Ranking (PR) procedure with respect to an underlying preference matrix $\mathbf{Q} \in \mathcal{P}_n$. For a constant $\gamma > 1$, $\widehat{\sigma}$ is said to be γ-approximate if $\mathrm{dist}(\widehat{\sigma}, \mathbf{Q}) \leq \gamma \min_{\sigma \in \mathcal{S}_n} \mathrm{dist}(\sigma, \mathbf{Q})$. Define the following distance which measures the fraction of response pairs over which two preference matrices $\{\mathbf{Q}, \mathbf{R}\} \in \mathcal{P}_n$ disagree.

$$\mathrm{dist}(\mathbf{Q},\mathbf{R}) := \binom{n}{2}^{-1} \sum_{i<j} \mathbb{1}\left((Q_{ij} > 1/2) \wedge (R_{ij} < 1/2)\right) + \binom{n}{2}^{-1} \sum_{i<j} \mathbb{1}\left((Q_{ij} < 1/2) \wedge (R_{ij} > 1/2)\right)$$

By Lemma 20 of (Rajkumar and Agarwal, 2016), for $\mathbf{Q} \in \mathcal{P}_n^{ST}$ and $\mathbf{R} \in \mathcal{P}_n$, we have $\mathrm{dist}(\widehat{\sigma}, \mathbf{Q}) \leq (1+\gamma)\,\mathrm{dist}(\mathbf{Q}, \mathbf{R})$. But note that it is possible that $\mathrm{dist}(\mathbf{Q}, \mathbf{R}) = 1$ as it is easy to construct by $\mathbf{R}$ that disagrees with $\mathbf{Q}$ in every entry by simply setting $\mathbf{R} = \mathbf{Q}^\top$. Now, we may set $\mathbf{Q} = \mathbf{P}$ and $\mathbf{R} = \mathbf{P^c}$ for any algorithm that uses $\mathbf{P^c}$ for ranking; specifically, for the adversary satisfying Assumption 1, we can see by a direct counting argument that $\mathrm{dist}(\mathbf{Q}, \mathbf{R}) \leq \frac{d(2n-1-d)}{n(n-1)}$ which proves the claim. □

4 Fully Observed Adversarial Setting

4.1 Algorithm

We present our main algorithm for robust passive ranking from pairwise comparisons in the presence of adversarial noise in Algorithm 1. The input data consist of the set of pairwise comparison results $\aleph = \{(i, j, \{y_{ij}^k\})\}$, $(i, j) \in [n] \times [n]$, $k \in [K]$, $y_{ij}^k \in \{0, 1\}$. The algorithm assumes the true rank of $\psi(\mathbf{P})$ as an input parameter; specifically, for the BTL model, we set $r = 2$. Algorithm 1 calls the Robust PCA and γ-approximate pairwise ranking procedures.

Algorithm 1 RORATRON: Robust Preference Data for Rigorous Alignment

Input: Comparison dataset $\aleph = \{(i, j, \{y_{ij}^k\})\}$, true rank r.
Output: Ranking of n responses, $\hat{\sigma} \in \mathcal{S}_n$.
1: Estimate entries of $\widehat{\mathbf{P}}$ for $i \leq j$ as:

$$\widehat{P}_{ij} = \begin{cases} \frac{1}{K}\sum_{k=1}^{K} y_{ij}^k & \text{if } i < j \\ 1/2 & \text{if } i = j \end{cases}$$

2: Set $\widehat{P}_{ij} = 1 - \widehat{P}_{ji}$ for all $i > j$.
3: Perform robust PCA: $\{\psi(\overline{\mathbf{P}}), \widehat{\mathbf{S}}\} \leftarrow \mathrm{RPCA}(\psi(\widehat{\mathbf{P}}), r)$.
4: Using a pairwise ranking procedure after taking the inverse transform: $\widehat{\sigma} \leftarrow \mathrm{PR}(\overline{\mathbf{P}})$.
5: **return** $\widehat{\sigma}$.

4.2 Analysis

We begin with a useful short result followed by the statement and the proof of our main result that, with high probability, we achieve ϵ–accurate ranking in polynomial time using polynomial number of samples, despite the presence of adversarial noise. In this context, it is noteworthy that we present the result for LR models which strictly contain the BTL model while being much more general (Rajkumar and Agarwal, 2016); upon proving this result, we specialize it to the classic BTL model as well (Corollary 1).

Lemma 1 (Some properties of the logit function). *Let* $a, b, c \in (0, 1)$ *such that* $c = a + b$. *Then, we have,*

1. $\psi(c) = \psi(a) + \psi(a + b) + \psi(1 - a)$
2. $\psi(a) + \psi(1 - a) = 0$.

Proof. Both follow by using the definition of the logit function that $\psi(a) = \log(a/(1-a))$ and using the property that $\log(ab) = \log(a) + \log(b)$. □

Theorem 1 (Provably good estimation of ranking in LR models in the presence of adversarial noise). *Let* $\mathbf{P} \in \mathcal{P}_n^{LR(\psi,r)}$ *be the true preference matrix according to which the pairwise comparison dataset* $\aleph = \{(i, j, \{y_{ij}^k\})\}$ *is generated for all responses pairs* (i, j) *such that* $k \in [K]$. *Let* $\widehat{\mathbf{P}}$ *be the empirical preference matrix computed using* $\aleph$. *Let* $\mathbf{S} \in [0, 1]^{n \times n}$ *be the adversarial matrix that additively corrupts* $\widehat{\mathbf{P}}$. *Let* ψ *be L-Lipschitz in* $[\frac{P_{\min}}{2}, 1 - \frac{P_{\min}}{2}]$ *and* $\psi(\mathbf{P})$ *be* μ*-incoherent. Let each pair be compared independently* $K \geq 16384\mu^2(1+\gamma)L^2n^2\log^2(n)/\epsilon\Delta^2$ *times where* $\Delta = \min_{i \neq j} \left|\psi(P_{ij}) - \psi(1/2)\right|$. *Then, with probability atleast* $1 - 1/n^3$, *Algorithm 1 returns an estimated permutation* $\widehat{\sigma}$ *such that* $\mathrm{dist}(\widehat{\sigma}, \mathbf{P}) \leq \epsilon$.

Remark 1 (Computational complexity). *In Algorithm 1, Step 1 takes* $O(n^2K) = O(n^4 \log^2 n/\epsilon)$

time, Step 3 takes $O(n^2r^2\log(1/\epsilon))$, and Step 4 takes $O(n^2 + n\log n)$ time. Thus, putting together the cost of these main steps, the overall computational complexity of our robust ranking algorithm for $\mathbf{P} \in \mathcal{P}_n^{LR(\psi,r)}$ is $O(n^4\log^2 n/\epsilon)$.

Remark 2 (Identifying adversarially corrupted pairwise comparisons). *From Step 3 of Algorithm 1, using Theorem 2 of (Netrapalli et al., 2014), we also have* $\mathrm{Supp}(\widehat{\mathbf{S}}) \subseteq \mathrm{Supp}(\mathbf{S})$ *and thus we can identify the corrupted pairwise comparison results.*

Proof. Let $\widetilde{P}_{ij}$ be the empirical probability estimate of P_{ij}. Note that we compute $\widetilde{P}_{ij} = \frac{1}{K}\sum_{k=1}^{K} y_{ij}^k$ from the given pairwise comparison dataset, $\aleph = \{(i,j,\{y_{ij}^k\})\}$. Now, $\widehat{\mathbf{P}} = \widetilde{\mathbf{P}} + \mathbf{S}$. By Lemma 1, we may write the adversarially corrupted empirical probability estimate as $\psi(\widehat{\mathbf{P}}) = \psi(\widetilde{\mathbf{P}}) + \widetilde{\mathbf{S}}$ where $\widetilde{\mathbf{S}} = \psi(\widetilde{\mathbf{P}}+\mathbf{S}) + \psi(1-\widetilde{\mathbf{P}})$. We have $\psi(\widetilde{\mathbf{P}}) = \psi(\mathbf{P}) + \widetilde{\mathbf{N}}$ where $\widetilde{\mathbf{N}} = \psi(\widetilde{\mathbf{P}}) - \psi(\mathbf{P})$. Now, this noise, $\widetilde{\mathbf{N}}$, is purely due to finite-sample effects which can be controlled (using concentration arguments given in the inequality ξ_3 below) by driving it down to as small a value as we want by ensuring large enough number of comparisons for each pair. Note that we input $\psi(\widehat{\mathbf{P}}) = \psi(\mathbf{P}) + \widetilde{\mathbf{S}} + \widetilde{\mathbf{N}}$ to Subroutine **??** and obtain $\psi(\overline{\mathbf{P}})$ as the output in Step 3 of Algorithm 1. Hence, using Theorem 2 from (Netrapalli et al., 2014), if $\left\|\widetilde{\mathbf{N}}\right\|_\infty \le \sigma_{\min}(\psi(\mathbf{P}))/100n$, we have,

$$\left\|\psi(\overline{\mathbf{P}}) - \psi(\mathbf{P})\right\|_F \le \epsilon' + 2\mu^2 r\left(7\left\|\widetilde{\mathbf{N}}\right\|_2 + \frac{8n}{r}\left\|\widetilde{\mathbf{N}}\right\|_\infty\right)$$

after $T \ge 10\log(3\mu^2 r\sigma_1/\epsilon')$ iterations associated with Subroutine RPCA. Next, we have, with probability at least $1 - 1/n^3$,

$$\begin{aligned}\left\|\psi(\overline{\mathbf{P}}) - \psi(\mathbf{P})\right\|_F &\le \epsilon' + 2\mu^2 r\left(7\left\|\widetilde{\mathbf{N}}\right\|_2 + \frac{8n}{r}\left\|\widetilde{\mathbf{N}}\right\|_\infty\right)\\ &\overset{\xi_1}{\le} \epsilon' + 32\mu^2 n\left\|\widetilde{\mathbf{N}}\right\|_2 \overset{\xi_2}{\le} \epsilon' + 32\mu^2 n\tau\\ &\overset{\xi_3}{\le} n\sqrt{\frac{\epsilon}{1+\gamma}}\frac{\Delta}{2}\end{aligned}$$

where ξ_1 follows by using $r \le n$ and $\left\|\widetilde{\mathbf{N}}\right\|_\infty \le \left\|\widetilde{\mathbf{N}}\right\|_2$, ξ_2 follows by substituting for $\widetilde{\mathbf{N}}$ from Lemma 2 with $K \ge \frac{L^2n^2\log^2 n}{\tau^2}$, and ξ_3 is obtained using $\epsilon' = n\sqrt{\frac{\epsilon}{1+\gamma}}\frac{\Delta}{4}$, $\tau = \min\left(\sigma_{\min}(\psi(\mathbf{P}))/100, \sqrt{\frac{\epsilon}{1+\gamma}}\frac{\Delta}{128\mu^2}\right)$. Then using similar arguments as proof of Theorem 13 in (Rajkumar and Agarwal, 2016), we obtain our result. □

Lemma 2 (Concentration of sampling noise). *Under the conditions of Theorem 1, let each response pair be compared such that the number of comparisons per response pair is $K \ge \frac{L^2n^2\log(n)}{\tau^2}$; with probability at least $1 - 1/n^3$, $\left\|\widetilde{\mathbf{N}}\right\|_2 \le \tau$.*

Proof. Let L be the Lipschitz constant of ψ and set $K \ge \frac{L^2n^2\log(n)}{\tau^2}$. Using the inequality that $\left\|\widetilde{\mathbf{N}}\right\|_2 \le n\left\|\widetilde{\mathbf{N}}\right\|_\infty$,

$$\begin{aligned}\Pr(\|\widetilde{\mathbf{N}}\|_2 \ge \tau) &\le \Pr\left(\left\|\widetilde{\mathbf{N}}\right\|_\infty \ge \frac{\tau}{n}\right)\\ &= \Pr\left(\exists(i,j) : \left|\psi(\widehat{P}_{ij}) - \psi(P_{ij})\right| \ge \frac{\tau}{n}\right)\\ &\le \sum_{i,j}\Pr\left(\left|\psi(\widehat{P}_{ij}) - \psi(P_{ij})\right| \ge \frac{\tau}{n}\right)\\ &\le \sum_{i,j}\Pr\left(\left|\widehat{P}_{ij} - P_{ij}\right| \ge \frac{\tau}{nL}\right) \le \frac{1}{n^3}\end{aligned}$$

□

Next, for completeness, we recall the following lemma (proved in Theorem 8 and Lemma 14 of (Rajkumar and Agarwal, 2016)) which characterizes the incoherence constant μ of $\mathbf{P} \in (\mathcal{P}_n^{LR(\psi,2)} \cap \mathcal{P}_n^{ST})$ in Assumption 1.

Lemma 3 (Incoherence of BTL and LR models). *We have $\mathbf{P} \in (\mathcal{P}_n^{LR(\psi,2)} \cap \mathcal{P}_n^{ST})$ if and only if $\psi(\mathbf{P}) = \mathbf{u}\mathbf{v}^\top - \mathbf{v}\mathbf{u}^\top$ for $\mathbf{u} \in \mathbb{R}_+^n$ and $\mathbf{v} \in \mathbb{R}^n$ where $\mathbf{u}^\top\mathbf{v} = 0$. Moreover, $\psi(\mathbf{P})$ is μ-incoherent where $\mu = \sqrt{\frac{n}{2}}\left(\frac{u_{\max}^2}{u_{\min}^2} + \frac{v_{\max}^2}{v_{\max}^2}\right)^{1/2}$ where $u_{\min} = \min_i|u_i|$, $u_{\max} = \max_i|u_i|$, $v_{\min} = \min_i|v_i|$ and $v_{\max} = \min_i|v_i|$. We also have $\mathcal{P}_n^{BTL} \subset (\mathcal{P}_n^{LR(\psi,2)} \cap \mathcal{P}_n^{ST})$ since we may set $\mathbf{u} = \mathbf{1}$ where $\mathbf{1}$ is the all-ones vector and $\mathbf{v} = \mathbf{w}$ where $\mathbf{w}$ is the BTL parameter vector. In this case, we may rewrite $\mu = \sqrt{\frac{n}{2}}\left(1 + \frac{(w_{\max}-\overline{w})^2}{(w_{\min}-\overline{w})^2}\right)$ where $\overline{w} = \frac{1}{n}\sum_{i=1}^n w_i$.*

The following corollary makes precise our claim that up to $O(n^2)$ response pairs may be subject to adversarial corruption, but our RORATRON algorithm still recovers a good ranking.

Corollary 1 (Recovery result for BTL model). *Consider $\mathbf{P} \in \mathcal{P}_n^{BTL}$. Using Assumption 1, let the adversarial matrix be $\mathbf{S} \in [0,1]^{n\times n}$ satisfying $\|\mathbf{S}\|_0 \le n/1024\mu^2$ where μ is characterized as in Lemma 3. Then, with probability $1 - 1/n^3$, the output of Algorithm 1 with input $\widehat{\mathbf{P}}$ computed using $\aleph = \{(i,j,\{y_{ij}^k\})\}$ satisfies and $r = 2$, $\mathrm{dist}(\widehat{\sigma}, \mathbf{P}) \le \epsilon$.*

Algorithm 2 CURATRON: Complete Robust Preference Data for Rigorous Alignment

Input: Comparison dataset $\aleph = \{(i, j, \{y_{ij}^k\})\}$, true rank r.
Output: Ranking of n responses, $\widehat{\sigma} \in \mathcal{S}_n$.
1: Estimate entries of $\widehat{\mathbf{P}}$ for $i \le j$ as:

$$\widehat{P}_{ij} = \begin{cases} \frac{1}{K}\sum_{k=1}^{K} y_{ij}^k & \text{if } i < j \text{ and } (i,j) \in \Omega \\ 1/2 & \text{if } i = j \text{ and } (i,j) \in \Omega \\ 1/2 & \text{if } (i,j) \notin \Omega \end{cases}$$

2: Set $\widehat{P}_{ij} = 1 - \widehat{P}_{ji}$ for all $i > j$.
3: Set $\mathbf{R} \leftarrow \text{OptSpace}(\psi(\widehat{\mathbf{P}})_\Omega)$.
4: Use a robust PCA procedure: $\psi(\overline{\mathbf{P}}) \leftarrow \text{RPCA}(\mathbf{R})$.
5: Using a pairwise ranking procedure after taking the inverse transform: $\widehat{\sigma} \leftarrow \text{PR}(\overline{\mathbf{P}})$.
6: **return** $\widehat{\sigma}$.

5 Partially Observered Adversarial Setting

In this section, we consider the partially observed and adversarially corrupted comparison results setting. Both factors can be modeled in a unified manner by setting the corresponding missing entries of the preference matrix to zero (or a specific constant to account for numerical stability). We present our robust ranking algorithm for this setting in Algorithm 2 – this essentially involves using the 'OptSpace' matrix completion algorithm of (Keshavan et al., 2010) followed by using the robust PCA algorithm of (Netrapalli et al., 2014) as sub-routines. We now derive the recovery guarantees as follows.

Theorem 2 (Provably good estimation of ranking in BTL model in the presence of adversarial noise as well as missing data). *Consider a similar notation as in Theorem 1 but let* $\mathbf{P} \in \mathcal{P}_n^{BTL}$. *Let* $\Omega \subseteq [n] \times [n]$ *be a set of compared response pairs. Assume* Ω *is drawn uniformly from all subsets of* $[n] \times [n]$ *of size* $|\Omega|$ *such that* $|\Omega| \ge C'' n \log(n)$ *and let the sparse noise satisfy* $\|\mathbf{S}\|_\infty \le \Delta_w \frac{\log(n)}{C_\Delta n}$ *where* $\Delta_w := \min_{i,j} |w_i - w_j|$. *Let the number of comparisons per pair be* $K \ge cn^4/\Delta_w$. *Then with probability at least* $1 - 2/n^3$, *Algorithm 2 returns a ranking that satisfies* $\text{dist}(\widehat{\sigma}, \mathbf{P}) \le \epsilon$.

Remark 3 (Robust Estimation of BTL Model in the Partially Observed Case). *For the BTL model, Theorem 2 says* $O(n \log n)$ *pairs suffice to estimate the BTL model, which matches bounds from (Rajkumar and Agarwal, 2016). Further, even in this incomplete comparison data case, we are able to tolerate uniformly random additive sparse noise with its maximum absolute entry scaling as the order of the BTL 'score-gap' divided by the number of responses up to logarithmic factors, ie,* $\widetilde{O}(\Delta_w/n)$.

Proof. From Lemma 3, we have $\psi(\mathbf{P}) = \mathbf{1}\mathbf{w}^\top - \mathbf{w}\mathbf{1}^\top$ for the BTL model where ψ is the logit function. Clearly, in this case, $\psi(\mathbf{P})$ is a real skew-symmetric matrix of rank $r = 2$. Since it is skew-symmetric, its eigenvalues, which are the roots of its charateristic polynomial, are of the form $\pm\lambda i$ for some $\lambda \in \mathbb{R}$ and $i = \sqrt{-1}$, and hence, $\sigma_{\min}(\psi(\mathbf{P})) = \sigma_{\max}(\psi(\mathbf{P}))$, ie, the condition number of $\psi(\mathbf{P})$, $\kappa = 1$. Now, we recall the spectral-lower bound from Corollary 2 of (Horne, 1997),

$$\sigma_{\min}(\psi(\mathbf{P})) \ge \frac{\|\psi(\mathbf{P})\|_F}{\sqrt{r(r-1)}} \ge \sqrt{\frac{n(n-1)}{2}}\Delta_w \tag{1}$$

where $\Delta_w = \min_{i,j} |w_i - w_j|$.

Let $\Omega \subseteq [n] \times [n]$ be a subset of all the response pairs with comparison results among which some might be corrupted by sparse noise, ie, $\psi(\widehat{\mathbf{P}}_\Omega) = \psi(\mathbf{P}_\Omega) + \widetilde{\mathbf{S}}_\Omega + \widetilde{\mathbf{N}}_\Omega$. Let $\mathbf{T} := \widetilde{\mathbf{S}}_\Omega + \widetilde{\mathbf{N}}_\Omega$. From Theorem 1.2 of (Keshavan et al., 2010), we have $\frac{1}{n}\left\|\psi(\widehat{\mathbf{P}}) - \psi(\mathbf{P})\right\|_F = \frac{1}{n}\|\mathbf{T} + \mathbf{M}\|_F \le C\kappa^2 \frac{n\sqrt{r}}{|\Omega|}\|\mathbf{T}\|_2$ where $\mathbf{M}$ is the noise matrix after obtaining the completed matrix $\psi(\widehat{\mathbf{P}})$ from $\psi(\widehat{\mathbf{P}}_\Omega)$ using OptSpace. Using triangle inequality and noting that $|\Omega| \ge C'' n \log(n)$, the noise may be bounded as

$$\begin{aligned} \|_{\widetilde{\mathbf{N}}_\Omega + \mathbf{M}}\|_\infty \le \left\|\widetilde{\mathbf{N}}_\Omega + \mathbf{M}\right\|_F &\le \|\mathbf{T}\|_2 \frac{\sqrt{2}Cn^2}{|\Omega|} + \left\|\widetilde{\mathbf{S}}_\Omega\right\|_F \\ &\overset{\zeta_1}{\le} C' \frac{n}{\log(n)} \left\|\widetilde{\mathbf{S}}_\Omega\right\|_2 \end{aligned} \tag{2}$$

where C, C' and C'' are constants and ζ_1 is obtained by using the triangle inequality that $\|\mathbf{T}\|_2 \le \left\|\widetilde{\mathbf{S}}_\Omega\right\|_2 + \left\|\widetilde{\mathbf{N}}_\Omega\right\|_2$, followed by setting $K \ge cn^4/\Delta_w$ for constant c and finally using $\left\|\widetilde{\mathbf{S}}_\Omega\right\|_F \le \sqrt{n}\left\|\widetilde{\mathbf{S}}_\Omega\right\|_2$. Then, combining Equations 2 and 1, we have if

$$\begin{aligned} \tfrac{\log(n)}{C_\Delta n}\Delta_w \ge \left\|\widetilde{\mathbf{S}}_\Omega\right\|_2 &= \left\|\psi(\widehat{\mathbf{P}}) - \psi(\widetilde{\mathbf{P}})\right\|_2 \\ &\ge \left\|\psi(\widehat{\mathbf{P}}) - \psi(\widetilde{\mathbf{P}})\right\|_\infty \ge L\left\|\widehat{\mathbf{P}} - \widetilde{\mathbf{P}}\right\|_\infty \ge \|\mathbf{S}\|_\infty \end{aligned}$$

where C_Δ is a global constant and using Lemma 2, then we have the guarantee (along similar lines as that of Theorem 1 that Algorithm 2 returns an estimated permutation which satisfies $\text{dist}(\widehat{\sigma}, \mathbf{P}) \le \epsilon$. □

6 Experiments

We now perform simulations in order to understand the performance of our robust ranking approach in practice in both general and LLM preference dataset settings.

6.1 Performance of Robust Ranking in LLM Preference Dataset

In this illustrative experiment, from the MT-Bench dataset (Zheng et al., 2023), we collect the data of the first prompt "Compose an engaging travel blog post about a recent trip to Hawaii, highlighting cultural experiences and must-see attractions" and its six responses from `GPT-3.5`, `GPT-4` (OpenAI et al., 2023), `Claude-v1` (Anthropic, 2023), `Vicuna-13B` (Chiang et al., 2023), `Alpaca-13B` (Taori et al., 2023), and `LLaMA-13B` (Touvron et al., 2023a). Additionally, we generated nine responses to the same prompt using `Llama-2-70B-chat-hf` (Touvron et al., 2023b), `Falcon-180B-chat` (Almazrouei et al., 2023), `Openchat-3.5` (Wang et al., 2023), `Mixtral-8x7B-Instruct-v0.1` (Jiang et al., 2024), `Mistral-7B-Instruct-v0.2` (Jiang et al., 2023), `Gemini-pro` (Gemini et al., 2023), `Dolphin-2.2.1-mistral-7B` (Hartford, 2023), `Solar-10.7B-instruct-v1.0` (Kim et al., 2023), `Yi-34B-chat` (01.ai, 2023) from Hugging Face's HuggingChat (Hugging Face, 2023) and LMSYS's Chatbot Arena (Zheng et al., 2023). So we have $n = 15$ responses.

Next, we rank the responses using OpenAI's GPT-4 Turbo `GPT-4-1106-preview` (OpenAI et al., 2023). This ranking helps us create the BTL parameter vector $\mathbf{w}$. We then sort this vector descendingly for visually accessible when building the corresponding preference matrix $\mathbf{P} \in \mathbb{R}^{n \times n}$. With $\binom{n}{2}$ comparisons in $\mathbf{P}$, we randomly remove entries based on a specified deletion probability parameter, dp, to simulate unobserved comparisons. We then create an adversarial skew-symmetric sparse matrix, $\mathbf{S}$, using the given matrix $\mathbf{P}$ and an adversarial corruption probability parameter ap. When corruption is applied, it involves randomly selecting a value from $U(-5, 5)$ and then adding to the $\mathbf{P}$ to give $\mathbf{P}^c$, which then becomes the input of our algorithm. It's important to note that $\mathbf{P}$ is a skew-symmetric matrix, any corruption must be applied to both ij and ji values.

Our experiment results visualized in Figure 2 show that $dp = 10\%$ and $ap = 10\%$ can significantly affect the ranking of different models and the rank of the matrix when performing logit link transformation. The ranking can get altered quite badly when compared to the original matrix. Also, the logit link transformation of the corrupted matrix is high-rank, which indicates that there are noises in the matrix. By using `CURATRON` to impute the missing comparisons and filter out the noisy sparse matrix, we successfully reconstruct the original matrix, which is low-rank when in logit link transformed form. As a result, we obtain the correct ranking. We also obtain noisy comparisons that can be used to identify responders with malicious intent and prevent them from continuing to alter results.

We now examine how our algorithm performs across different levels of unobserved and adversarially corrupted comparisons. In the plots shown in Figure 3, we compare the performance of our approach by varying two parameters, dp and ap. We use normalized Frobenius error, correlation, and ranking distance as evaluation metrics. Our results are averaged over 5 runs. When there is no adversarial noise, we can recover the original $\mathbf{P}$ with no normalized Frobenius error and perfect correlation and ranking, even if 50% of the comparison data was missing. This suggests that we may not need to collect all comparisons from humans to obtain the entire data. We observe that, with $n = 15$, we only need to obtain about $50 - 55\%$ of the 105 comparisons and fill in the rest with our algorithm to achieve a strict 0% NFE, perfect correlation, and ranking. On the other hand, when missing data is absent, our algorithm performs well with NFE of approximately 6%, even when 35% of the comparison data is adversarially corrupted. When both adversarial noise and missing data are present, we can achieve a low NFE of around 4% when both 15% of the comparison data is missing and 15% of adversarially corrupted comparisons (30% in total) affect $\mathbf{P}$.

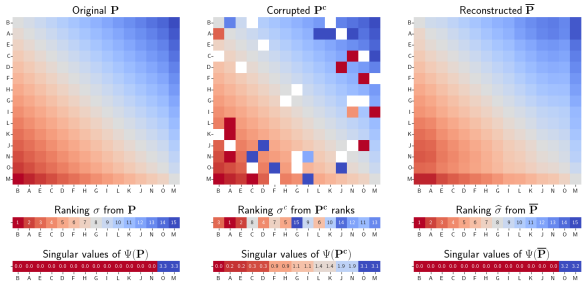

Figure 2: Left: Original matrix. Middle: corrupted matrix. Right: reconstructed matrix. The corrupted matrix has 10% adversarial corruptions and 10% of unobserved comparisons. We use our `CURATRON` algorithm to successfully recover the original matrix.

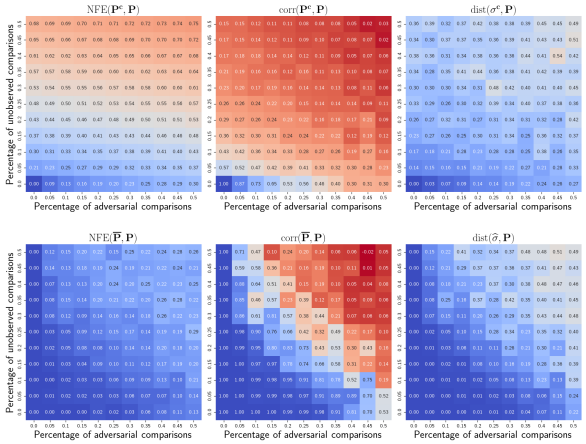

Figure 3: Average over 5 runs of reconstruction error, correlation, and distance between reconstructed ranking and original ranking for different percentages of unobserved and adversarial comparisons.

7 Conclusion

Our study examines how missing information and distorted feedback can impact LLMs, potentially compromising their performance in terms of alignment with human values. We have proposed a robust algorithm for provably correct and efficient ranking responses in the BTL, LR, and general binary choice models. This robust ranking data is then input in the PL step. Further, we also handled the partially observed setting, wherein only some response pairs are compared, by integrating matrix completion techniques into our robust learning algorithm. In all cases, we provided statistical and computational guarantees using novel techniques. Through our comprehensive analysis, we hope to contribute to the ongoing discussion on AI safety by helping to create and scale LLMs/AGI models that align with human values and expectations. Some future research directions include tightening the recovery results for partially observed settings under weaker conditions (possibly using noisy-case extensions of (Yi et al., 2016)), exploring other notions of adversarial noise, and understanding the minimax optimal rates for ranking estimators under various noise models. We also plan to study the parametric non-active pairwise ranking setting, studying lower bounds and practical algorithms in the active setting similar to (Heckel et al., 2016). Furthermore, it would be interesting to investigate whether we can extend this approach to solve the entity corruption problem in retrieval models, as shown in (Naresh et al., 2022). Another research direction could be defining an alignment framework that expands DPO to various objective functions based on Rank Centrality (Negahban et al., 2017). Finally, we aim to examine the relationship between robust PL and model capacity, as this can shed light on the trade-offs between model complexity and generalization performance.

References

01.ai. 2023. Yi-34b. https://www.01.ai. Accessed 03-03-2024.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models. *Preprint*, arXiv:2311.16867.

Anthropic. 2023. Introducing Claude. https://www.anthropic.com/news/introducing-claude. Accessed 03-03-2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. https://lmsys.org/blog/2023-03-30-vicuna/. Accessed 03-03-2024.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. 2023. GPTs are GPTs: An early look at the labor market impact potential of Large Language Models. *Preprint*, arXiv:2303.10130.

Gemini, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Eric Hartford. 2023. Dolphin-2.2.1-mistral-7b. https://huggingface.co/cognitivecomputations/dolphin-2.2.1-mistral-7b. Accessed 03-03-2024.

Reinhard Heckel, Nihar B Shah, Kannan Ramchandran, and Martin J Wainwright. 2016. Active ranking from pairwise comparisons and when parametric assumptions don't help. *arXiv preprint arXiv:1606.08842*.

Bill G Horne. 1997. Lower bounds for the spectral radius of a matrix. *Linear algebra and its applications*, 263:261–273.

Daniel Hsu, Sham M Kakade, and Tong Zhang. 2011. Robust matrix decomposition with sparse corruptions. *IEEE Transactions on Information Theory*, 57(11):7221–7234.

Hugging Face. 2023. Huggingchat. https://huggingface.co/chat. Accessed 03-03-2024.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. 2010. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11(Jul):2057–2078.

Dahyun Kim, Chanjun Park, Sanghoon Kim, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, Changbae Ahn, Seonghoon Yang, Sukyung Lee, Hyunbyung Park, Gyoungjin Gim, Mikyoung Cha, Hwalsuk Lee, and Sunghun Kim. 2023. Solar 10.7b: Scaling large language models with simple yet effective depth up-scaling. *Preprint*, arXiv:2312.15166.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Niranjan Uma Naresh, Ziyan Jiang, Ankit, Sungjin Lee, Jie Hao, Xing Fan, and Chenlei Guo. 2022. PENTATRON: Personalized context-aware transformer for retrieval-based conversational understanding. *Preprint*, arXiv:2210.12308.

Sahand Negahban, Sewoong Oh, and Devavrat Shah. 2017. Rank centrality: Ranking from pairwise comparisons. *Operations Research*, 65(1):266–287.

Praneeth Netrapalli, UN Niranjan, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain. 2014. Non-convex robust PCA. In *Advances in Neural Information Processing Systems*, pages 1107–1115.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Arun Rajkumar and Shivani Agarwal. 2016. When can we rank well from comparisons of $o(n \log(n))$ non-actively chosen pairs? In *29th Annual Conference on Learning Theory*, pages 1376–1401.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca. Accessed 03-03-2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of LM alignment. *Preprint*, arXiv:2310.16944.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023. Openchat: Advancing open-source language models with mixed-quality data. *Preprint*, arXiv:2309.11235.

Xinyang Yi, Dohyung Park, Yudong Chen, and Constantine Caramanis. 2016. Fast algorithms for Robust PCA via Gradient Descent. *arXiv preprint arXiv:1605.07784*.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J. Liu. 2023. SLiC-HF: Sequence likelihood calibration with human feedback. *Preprint*, arXiv:2305.10425.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *Preprint*, arXiv:2306.05685.

Author Index