Hallucination Detection-Guided Preference Optimization for Clinical Summarization

Anonymous ACL submission

Abstract

Large language models (LLMs) have shown promise on summarization tasks, but they often produce hallucinations, which are unsupported or incorrect statements that limit their reliability in specialized healthcare applications. We introduce HALLUCINATION DETECTION GUIDED SELF-REFINEMENT (HDSR), an inference-time method that leverages hallucination detectors to guide iterative summary revisions toward factual corrections. Building on this, we propose HDSR for Preference Learning (HDSR-PL), which converts detector-guided refinement trajectories into preference pairs for model finetuning. Extensive experiments show that our methods substantially reduce hallucinations for Llama and Gemma models in summarizing real-world clinical notes from *MIMIC-IV-Note v2.2*. For example, HDSR reduces 24% and HDSR-PL reduces 48% hallucinations in Llama-3.1-8B-Instruct. Importantly, both methods preserve summary fluency, coherence, and relevance according to human expert and LLM-Jury evaluations. Together, these results demonstrate that detection-informed refinement and preference learning offer an automated solution for improving factual faithfulness in clinical summarization.

1 Introduction

Large language models achieve strong summarization performance but often generate hallucinations, defined as content not supported by the source or inconsistent with world knowledge (Maynez et al., 2020; Ji et al., 2023a; Tang et al., 2023). Hallucinations remain prevalent even in recent LLMs, with rates of roughly 40-50% on benchmarks such as FAITHBENCH (Bao et al., 2025) indicating persistent factual unreliability. This issue is critical in clinical summarization, where LLMs condense long patient records to support care delivery. Despite comparable performance with clinician-written summaries (Veen et al., 2024), hallucinations manifest as fabricated or distorted clinical statements that are hard to detect (Asgari et al., 2025; Kim et al., 2025). Even domain-adapted models frequently introduce hallucinations expressed with medically plausible terminology, requiring expert review (Hegselmann et al., 2024; Williams et al., 2025; Fang et al., 2024; Das et al., 2025), and underscoring the need for stricter factual reliability in clinical summarization than in general-domain tasks.

Existing hallucination mitigation strategies are broadly divided into training-time and inference-time approaches. Training-time methods include domain pretraining, continual pretraining, supervised fine-tuning, parameter-efficient adapters (Veen et al., 2024; Zaretsky et al., 2024), and specialized loss functions or reinforcement learning objectives (Fabbri et al., 2021; Bao et al., 2025; Asgari et al., 2025). These methods require large volumes of high-quality domain data (Hegselmann et al., 2024) and depend on factuality metrics that correlate weakly with clinical correctness, limiting their practical impact. Reinforcement learning from human or AI feedback offers greater flexibility, but requires carefully engineered preference data that is difficult to scale in medical settings (Lee et al., 2024). Inference-time techniques avoid altering model parameters: retrieval-augmented generation grounds outputs in external documents yet depends on retrieval quality (Koopman and Zuccon, 2023; Wan et al., 2024; Kim et al., 2025), while self-refinement(Madaan et al., 2023) and verification loops (Tang et al., 2023) iteratively critique and revise generations, improving factuality at the cost of higher inference overhead and a risk of over-editing. Recent work combining iterative refinement with alignment shows such approaches can strip salient content or introduce new hallucinations even when fluency is preserved (Ji et al., 2023b). A closely related approach is SYNFAC-EDIT (Mishra et al., 2024), which generates syn-

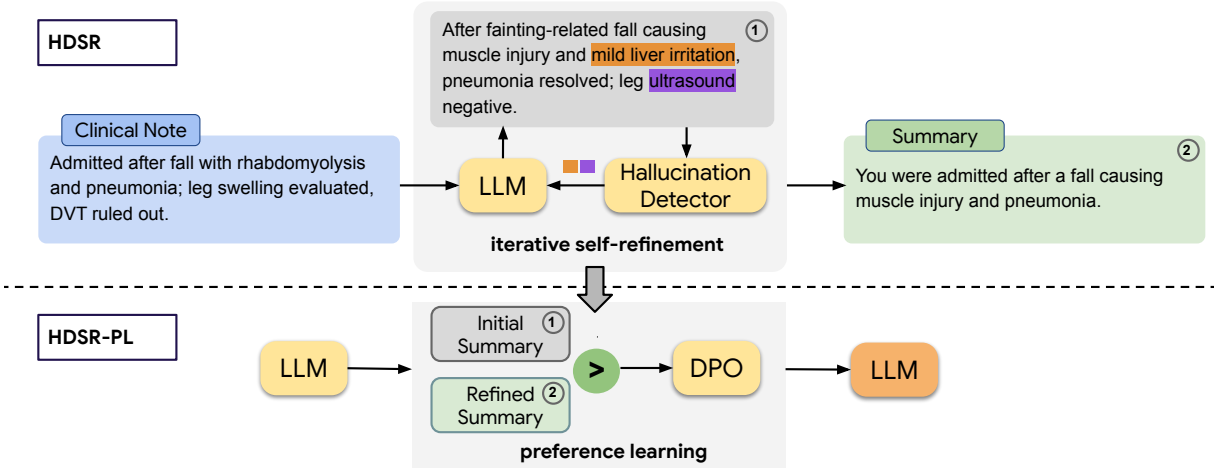

Figure 1: **Overview of hallucination mitigation via detection-informed self-refinement.** Given an input clinical note, a language model generates an initial summary that may contain unsupported or hallucinated medical content. A hallucination detector identifies unsupported content, which is used to guide iterative self-refinement toward removing factual errors rather than stylistic changes (*top;* HDSR). The intermediate outputs from HDSR process are converted into preference pairs and used for preference learning (e.g., DPO), amortizing faithful behavior and yielding hallucination-mitigated summaries at inference time (*bottom;* HDSR-PL).

thetic edit feedback for preference optimization but relies on predefined error types and external edit models.

Our pipeline combines hallucination detection with iterative self refinement, using detector feedback to guide self refinement and to form preference pairs from the original and revised summaries. We then train LLMs using direct preference optimization on these preference pairs, producing models that generate more faithful summaries with fewer hallucinations. On the MIMIC IV clinical summarization hallucination datasets (Hegselmann et al., 2024), our method reduces hallucinations by approximately 24% through self refinement and by up to 48% after fine tuning, while preserving summary quality and fluency according to both human and LLM-as-judge evaluations, and without incurring additional inference time cost.

2 Method

Figure 1 illustrates our pipeline, which uses hallucination detection feedback to guide summary revision and train models to prefer faithful summaries.

2.1 Problem Statement

Given a source document and a generated summary, the goal of hallucination mitigation is to produce summaries that are informative while remaining faithful to the source. We view a source document and its summary as describing clinical facts documented during a patient's hospital course. A summary is faithful if these facts are supported by the source document. Hallucinations arise when a summary introduces unsupported facts or adds procedures or findings that were not documented, for example when "leg ultrasound negative" appears in the summary without support from the source.

2.2 Hallucination Detection Guided Self-Refinement

During self-refinement, a hallucination detector is applied to the generated summary to identify content that is unsupported or inconsistent with the source document. Detector feedback highlights unsupported spans or statements, and the model is prompted to revise these parts while preserving supported content (see Appendix A.2). This process focuses revisions on factual corrections rather than stylistic changes.

Detection and revision are applied iteratively. After each revision, the updated summary is re-evaluated by the detector, and the resulting feedback guides subsequent revisions until a fixed iteration limit is reached or no further hallucinations are detected. This detector guided refinement improves factual alignment while maintaining overall fluency.

We use existing hallucination detectors from prior work, specifically MedCat (Kraljevic et al., 2021) and prompt based detectors following the

MedAlign annotation guidelines (Fleming et al., 2024). MedCat links clinical concepts in the source document and the summary to biomedical ontologies in order to flag unsupported or missing content. MedAlign defines a clinically motivated taxonomy of hallucination categories, such as unsupported procedures or medications, which we implement as a prompted detector.

2.3 Preference Learning from Detection Guided Self-Refinement

Detection-guided self-refinement produces pairs of summaries for each source document, consisting of an initial summary and a revised summary with fewer hallucinations. We treat the revised summary as the preferred output and the initial summary as the non-preferred output, thereby forming preference pairs without human annotation. We train the summarization model using direct preference optimization on these pairs. This objective encourages the model to internalize detector guided factual corrections, amortizing improvements from self refinement into the model parameters and enabling faithful generation at inference time without additional refinement steps.

3 Experiments

3.1 Experimental Setup

We study clinical summarization from Brief Hospital Course (BHC) sections to Discharge Instructions (DI) using datasets derived from *MIMIC-IV-Note v2.2* (Hegselmann et al., 2024). Experiments are evaluated on the hallucination-annotated subset *Hallucination-Generated-DI*, following the task formulation and annotation guidelines of Hegselmann et al. (2024).

We include GPT-5 (OpenAI, 2025) zero-shot as a closed-source reference and primarily focus on LLaMA-3.1-8B-Instruct (Meta, 2024b) as the open-source base model. In addition, we report results for two smaller open-source models, LLaMA-3.2-3B-Instruct (Meta, 2024a) and Gemma-3-4B-IT (Google, 2025), evaluated under a limited setting. For LLaMA-3.1-8B-Instruct, we evaluate zero-shot generation, supervised fine-tuning (SFT), detection-informed self-refinement at inference time (HDSR), and detection-informed preference learning at training time (HDSR-PL). Detection-informed variants use MedCat (Kraljevic et al., 2021) and MedAlign (Fleming et al., 2024) as hallucination detectors.

Evaluation. We report entity-level hallucination counts and clinician-based human evaluation on summary quality across four main metrics: Consistency, Coherence, Fluency, and Relevance. We follow the protocol of Hegselmann et al. (2024), which includes quantitative fine-grained hallucination annotations identifying unsupported and contradicted content, as well as qualitative analysis of summary quality across the same dimensions. Annotations are performed by a team of clinicians blinded to model identity: for the quantitative task, each summary is annotated independently by two annotators (H1 and H2) with adjudication of disagreements, while for the qualitative analysis, two annotators score summaries independently without adjudication.

For LLaMA-3.1-8B-Instruct, human evaluation is conducted in a blind setting with double annotation and adjudication for hallucination labels, forming the basis of the main results in Table 1. For LLaMA-3.2-3B-Instruct and Gemma-3-4B-IT, due to resource constraints, we report clinician-provided hallucination counts annotated by a single clinician (H1) and replace human qualitative evaluation with automatic LLM-as-judge scores.

3.2 Main Results

Table 1 presents the main results on the *Hallucination-Generated-DI* benchmark, comparing zero-shot generation, supervised fine-tuning (SFT), HDSR, and HDSR-PL using LLaMA-3.1-8B-Instruct. We also report limited results for LLaMA-3.2-3B-Instruct and Gemma-3-4B-IT. The following analysis focuses on LLaMA-3.1.

SFT amplifies hallucinations. Supervised fine-tuning on clinician-written references substantially worsens factual alignment and increases hallucinations (57) to nearly twice the zero-shot baseline. Although supervised fine-tuning improves fluency (4.53) and coherence (4.43), consistency drops sharply (3.03).

HDSR improves summary quality and mitigates hallucinations. Our HDSR yields a substantial improvement over both baselines. Using MedAlign as the detector, HDSR reduces hallucinations to 22 while simultaneously achieving the best overall human-evaluated summary quality across all dimensions.

HDSR-PL further mitigates hallucinations. HDSR-PL, using MedCat-derived preference

Model / Method	Hallucination Count ↓	Summary Quality Metrics				
		Consistency↑	Coherence↑	Fluency↑	Relevance↑	Average↑
GPT-5 Prompting	36	3.55	4.73	4.73	4.08	4.27
LLaMA-3.1-8B-Instruct						
Prompting	29	4.08	3.83	4.05	3.23	3.79
SFT	57	3.03	4.43	4.53	3.03	3.75
HDSR (best; MedAlign)	22	4.13	**4.48**	**4.53**	**3.95**	**4.27**
HDSR-PL (best; MedCat)	**15**	**4.40**	4.28	4.05	3.90	4.16
LLaMA-3.2-3B-Instruct						
Prompting	26	4.13	**4.58**	**4.68**	4.20	4.39
HDSR-PL (best; MedCat)	**13**	**4.23**	**4.58**	**4.68**	**4.30**	**4.44**
Gemma-3-4b-it						
Prompting	15	4.23	4.65	4.68	4.48	4.51
HDSR-PL (best; MedCat)	**13**	**4.33**	**4.65**	**4.70**	**4.63**	**4.58**

Table 1: **Results on *Hallucination-Generated-DI*.** Both variants of our detector-guided approach outperform the zero-shot and SFT baselines. Note that for LLama-3.2-3B-Instruct and Gemma-3-4b-it, we use automatic LLM-as-Judge for summary quality metrics.

Table 2: Distribution of hallucination error types across models.

Model	Unsup. Condition	Unsup. Procedure	Unsup. Medication	Unsup. Time	Unsup. Location	Unsup. Number	Unsup. Name	Unsup. Word	Unsup. Other	Contradicted Fact	Incorrect Fact
LLaMA-3.1-8B-Instruct											
Prompting	8	2	1	3	4	0	1	2	0	8	0
SFT	15	11	8	3	6	0	3	3	1	7	0
Self-Refine (No Det.)	10	2	6	2	3	3	1	1	1	2	0
HDSR (w/MedCat)	7	1	3	4	3	2	2	2	0	1	0
HDSR (w/MedAlign)	4	2	1	3	3	1	1	0	2	5	0
HDSR-PL (best; MedCat)	3	1	0	1	2	0	0	0	0	8	0

Table 3: **Impact of detection signal for hallucination mitigation.** Comparison between self-refinement with and without detectors on LLaMA-3.1-8B.

Model / Method	Hallucination Count ↓	Summary Quality Metrics			
		Consistency↑	Coherence↑	Fluency↑	Relevance↑
Self-Refine (No Det.)	31	3.85	4.25	**4.58**	**4.20**
HDSR (w/MedCat)	25	4.08	**4.55**	**4.58**	4.13
HDSR (w/MedAlign)	**22**	**4.13**	4.48	4.53	3.95

pairs, achieves the lowest hallucination count overall (15), corresponding to an approximate 48% reduction relative to the zero-shot baseline and a 74% reduction relative to SFT. HDSR-PL slightly underperforms HDSR on fluency and coherence. This trade-off reflects the difference between the two variants of our pipeline: HDSR benefits from direct, entity-specific detector feedback, whereas HDSR-PL learns a more general preference for factual alignment that may smooth over fine-grained stylistic details but yields factually aligned outputs.

Analyzing the types of hallucinations. Table 2 breaks down hallucinations by error type for LLaMA-3.1-8B, revealing that unsupported conditions, procedures, and medications account for the majority of errors in zero-shot and supervised fine-tuned models. SFT amplifies these clinically critical error types, increasing unsupported conditions from 8 to 15 and procedures from 2 to 11. Self-refinement without detection reduces some surface-level errors but leaves substantial unsupported content uncorrected. In contrast, HDSR variants consistently reduce errors across multiple categories, particularly unsupported conditions and procedures, indicating that the detector guides corrections toward factually grounded revisions.

Impact of detector on hallucination mitigation. Table 3 shows that incorporating an explicit detection signal reduces hallucination count for HDSR by 29% when using MedAlign as the hallucination detector and 19% when using MedCat.

4 Conclusion

We presented two complementary approaches for mitigating hallucinations in clinical summarization using automatic hallucination detection as supervision. HDSR is an inference-time method that integrates detector feedback into iterative self-refinement, focusing revisions on unsupported or contradicted medical content while preserving grounded information. HDSR-PL extends this framework to training time by transforming refinement trajectories into preference pairs, enabling direct preference optimization that amortizes factual alignment.

Limitations

Our work highlights several areas for further investigation. First, both HDSR and HDSR-PL rely on the quality of the hallucination detector used to provide supervision. While we treat detectors as black-box assessors, their errors–such as false positives that signal supported content or false negatives that miss subtle inconsistencies–can limit the effectiveness of refinement and downstream preference learning. Improving detector accuracy or combining multiple complementary detectors may further enhance performance.

In addition, our evaluation focuses on clinical summarization tasks derived from *MIMIC-IV-Note v2.2*, specifically BHC to Discharge Instructions generation. Although this is a high-impact and clinically relevant setting, the transferability of our approach to other medical note types or non-clinical summarization domains remains to be explored.

Finally, inference-time self-refinement (HDSR) incurs additional computational cost due to multiple refinement iterations and detector calls, which may limit its applicability in latency-sensitive settings. While HDSR-PL addresses this by amortizing refinement into training, it requires additional training resources and preference optimization, which may not be feasible in all deployment scenarios.

References

Elham Asgari, Nina Montaña-Brown, Magda Dubois, Saleh Khalil, Jasmine Balloch, Joshua Au Yeung, and Dominic Pimenta. 2025. A framework to assess clinical safety and hallucination rates of LLMs for medical text summarisation. *npj Digital Medicine*, 8(1):274.

Forrest Sheng Bao, Miaoran Li, Renyi Qu, Ge Luo, Erana Wan, Yujia Tang, Weisi Fan, Manveer Singh Tamber, Suleman Kazi, Vivek Sourabh, Mike Qi, Ruixuan Tu, Chenyu Xu, Matthew Gonzales, Ofer Mendelevitch, and Amin Ahmad. 2025. FaithBench: A Diverse Hallucination Benchmark for Summarization by Modern LLMs. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 448–461, Albuquerque, New Mexico. Association for Computational Linguistics.

Anindya Bijoy Das, Shibbir Ahmed, and Shahnewaz Karim Sakib. 2025. Hallucinations and Key Information Extraction in Medical Texts: A Comprehensive Assessment of Open-Source Large Language Models. *Preprint*, arXiv:2504.19061.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating Summarization Evaluation. *Preprint*, arXiv:2007.12626.

Biaoyan Fang, Xiang Dai, and Sarvnaz Karimi. 2024. Understanding Faithfulness and Reasoning of Large Language Models on Plain Biomedical Summaries. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9890–9911, Miami, Florida, USA. Association for Computational Linguistics.

Scott L. Fleming, Alejandro Lozano, William J. Haberkorn, Jenelle A. Jindal, Eduardo Reis, Rahul Thapa, Louis Blankemeier, Julian Z. Genkins, Ethan Steinberg, Ashwin Nayak, Birju S. Patel, Chia-Chun Chiang, Alison Callahan, Zepeng Huo, Sergios Gatidis, Scott J. Adams, Oluseyi Fayanju, Shreya J. Shah, Thomas Savage, and 11 others. 2024. Medalign: A clinician-generated dataset for instruction following with electronic medical records. In *Thirty-Eighth AAAI Conference on Artificial Intelligence*.

Google. 2025. google/gemma-3-4b-it. Online; accessed 2026-01-05.

Stefan Hegselmann, Shannon Zejiang Shen, Florian Gierse, Monica Agrawal, David Sontag, and Xiaoyi Jiang. 2024. A Data-Centric Approach To Generate Faithful and High Quality Patient Summaries with Large Language Models. *Preprint*, arXiv:2402.15422.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Delong Chen, Wenliang Dai, Ho Shu Chan, Andrea Madotto, and Pascale Fung. 2023a. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12):1–38.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023b. Towards Mitigating Hallucination in Large Language Models via Self-Reflection. *Preprint*, arXiv:2310.06271.

Yubin Kim, Hyewon Jeong, Shan Chen, Shuyue Stella Li, Mingyu Lu, Kumail Alhamoud, Jimin Mun, Cristina Grau, Minseok Jung, Rodrigo Gameiro, Lizhou Fan, Eugene Park, Tristan Lin, Joonsik Yoon, Wonjin Yoon, Maarten Sap, Yulia Tsvetkov, Paul Liang, Xuhai Xu, and 6 others. 2025. Medical Hallucinations in Foundation Models and Their Impact on Healthcare. *Preprint*, arXiv:2503.05777.

Bevan Koopman and Guido Zuccon. 2023. Dr ChatGPT tell me what I want to hear: How different prompts impact health answer correctness. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15012–15022, Singapore. Association for Computational Linguistics.

Zeljko Kraljevic, Thomas Searle, Anthony Shek, Lukasz Roguski, Kawsar Noor, Daniel Bean, Aurelie Mascio, Leilei Zhu, Amos A Folarin, Angus Roberts, Rebecca Bendayan, Mark P Richardson, Robert Stewart, Anoop D Shah, Wai Keong Wong, Zina Ibrahim, James T Teo, and Richard J B Dobson. 2021. Multi-domain clinical natural language processing with MedCAT: The medical concept annotation toolkit. *Artif. Intell. Med.*, 117:102083.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2024. RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback. *Preprint*, arXiv:2309.00267.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-Refine: Iterative Refinement with Self-Feedback. *Preprint*, arXiv:2303.17651.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On Faithfulness and Factuality in Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Meta. 2024a. meta-llama/llama-3.2-3b-instruct. Online; accessed 2026-01-05.

Meta. 2024b. meta-llama/llama-3.1-8b-instruct. Online; accessed 2025-02-21.

Prakamya Mishra, Zonghai Yao, Parth Vashisht, Feiyun Ouyang, Beining Wang, Vidhi Dhaval Mody, and Hong Yu. 2024. SYNFAC-EDIT: Synthetic Imitation Edit Feedback for Factual Alignment in Clinical Summarization. *Preprint*, arXiv:2402.13919.

OpenAI. 2025. Gpt-5 system card. `https://cdn.openai.com/gpt-5-system-card.pdf`. Official technical report describing the GPT-5 architecture and capabilities.

Liyan Tang, Zhaoyi Sun, Betina Idnay, Jordan G. Nestor, Ali Soroush, Pierre A. Elias, Ziyang Xu, Ying Ding, Greg Durrett, Justin F. Rousseau, Chunhua Weng, and Yifan Peng. 2023. Evaluating large language models on medical evidence summarization. *npj Digital Medicine*, 6(1):158.

Dave Van Veen, Cara Van Uden, Louis Blankemeier, Jean-Benoit Delbrouck, Asad Aali, Christian Bluethgen, Anuj Pareek, Malgorzata Polacin, Eduardo Pontes Reis, Anna Seehofnerova, Nidhi Rohatgi, Poonam Hosamani, William Collins, Neera Ahuja, Curtis P. Langlotz, Jason Hom, Sergios Gatidis, John Pauly, and Akshay S. Chaudhari. 2024. Adapted Large Language Models Can Outperform Medical Experts in Clinical Text Summarization. *Nature Medicine*, 30(4):1134–1142.

Alexander Wan, Eric Wallace, and Dan Klein. 2024. What Evidence Do Language Models Find Convincing? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7468–7484, Bangkok, Thailand. Association for Computational Linguistics.

Christopher Y. K. Williams, Jaskaran Bains, Tianyu Tang, Kishan Patel, Alexa N. Lucas, Fiona Chen, Brenda Y. Miao, Atul J. Butte, and Aaron E. Kornblith. 2025. Evaluating large language models for drafting emergency department encounter summaries. *PLOS Digital Health*, 4(6):e0000899.

Jonah Zaretsky, Jeong Min Kim, Samuel Baskharoun, Yunan Zhao, Jonathan Austrian, Yindalon Aphinyanaphongs, Ravi Gupta, Saul B. Blecker, and Jonah Feldman. 2024. Generative Artificial Intelligence to Transform Inpatient Discharge Summaries to Patient-Friendly Language and Format. *JAMA network open*, 7(3):e240357.

A Additional Details

A.1 Summarization Prompt

Summarization Prompt

```
<|system|>
You are a helpful assistant that helps patients understand their medical records.
<|end|>

<|user|>
You will be given some doctor's notes and you will need to summarize the patient's brief
    hospital course in one paragraph. Please only include key events and findings and avoid
    using medical jargons, and you MUST start the summary with You were admitted.
{% if demonstrations %}
Here are some examples:

{% for example in demonstrations %}
DOCUMENT:
{{ example.text }}

SUMMARY:
{{ example.summary }}

{% endfor %}
{% endif %}
DOCUMENT: {{ context }}
<|end|>
```

A.2 Self-refinement Prompts

Self-refinement revision without detectors Prompt

```
<|system|>
You are a clinical summarization assistant.
<|end|>

<|user|>
You will be given:
- A brief hospital course (SOURCE FACTS)
- A DRAFT SUMMARY that contains potential issues.

Your task is to REVISE the DRAFT SUMMARY so that it is fully consistent with the SOURCE FACTS
    and suitable as an accurate after-visit summary.

Instructions:
1. Treat the SOURCE FACTS as authoritative.
2. Check for missing or incomplete information in the DRAFT SUMMARY compared to the SOURCE
    FACTS, especially:
   - Chief complaint / reason for visit
   - Presenting symptoms
   - Procedures performed
   - Medications (new, changed, or discontinued)
   - Vital signs
   - Key laboratory or imaging findings
4. Do **not** invent or infer new diagnoses, medications, procedures, or dates.
5. Keep professional tone and structure suitable for a discharge or after-visit summary.
6. Prefer terms and values exactly as stated in the SOURCE FACTS.
7. If the DRAFT SUMMARY is missing clinically important info that IS present in the SOURCE
    FACTS (e.g. a key procedure or discharge medication), ADD it.
8. Always start the summary with You were admitted and refer to the patient as you / your.
9. Return only the revised summary text (no explanations or markup).

---
[SOURCE FACTS / BRIEF HOSPITAL COURSE]
{{ context }}
```

```
[DRAFT SUMMARY WITH FEEDBACK ANNOTATIONS]
{{ summary }}
<|end|>
```

Self-refinement revision with detectors Prompt

```
<|system|>
You are a clinical summarization assistant.
<|end|>

<|user|>
You will be given:
- A brief hospital course (SOURCE FACTS)
- A DRAFT SUMMARY that contains feedback annotations marking potential issues.
  Potentially incorrect or unsupported text is enclosed in <error> ... </error> tags.

Your task is to REVISE the DRAFT SUMMARY so that it is fully consistent with the SOURCE FACTS
     and suitable as an accurate after-visit summary.

Instructions:
1. Treat the SOURCE FACTS as authoritative.
2. For each segment wrapped in <error> ... </error>:
   - If the content is unsupported or contradicted by the SOURCE FACTS  remove or correct it.
   - If the content is partially correct  rewrite it using information from the SOURCE FACTS.
   - If the content is accurate  keep it, but remove the <error> tags.
3. Check for missing or incomplete information in the DRAFT SUMMARY compared to the SOURCE
     FACTS, especially:
   - Chief complaint / reason for visit
   - Presenting symptoms
   - Procedures performed
   - Medications (new, changed, or discontinued)
   - Vital signs
   - Key laboratory or imaging findings
4. Do **not** invent or infer new diagnoses, medications, procedures, or dates.
5. Keep professional tone and structure suitable for a discharge or after-visit summary.
6. Prefer terms and values exactly as stated in the SOURCE FACTS.
7. If the DRAFT SUMMARY is missing clinically important info that IS present in the SOURCE
     FACTS (e.g. a key procedure or discharge medication), ADD it.
8. Always start the summary with You were admitted and refer to the patient as you / your.
9. Return only the revised summary text (no explanations or markup).

---
[SOURCE FACTS / BRIEF HOSPITAL COURSE]
{{ context }}

[DRAFT SUMMARY WITH FEEDBACK ANNOTATIONS]
{{ summary_with_errors }}
<|end|>
```

A.3 Hallucination Detection

Medalign zero-shot Prompt

```
<|system|>
You are a helpful assistant that helps patients understand their medical records.
<|end|>

<|user|>
You will be given a doctor's notes and a summary with potentially incorrectness. Your task is
     to identify spans with erroneous, contradictory, or unsupported facts in the summary,
     and label them using the <error> tag (e.g. <error>incorrect fact</error>). There could
     be more than one error in the summary.
```

```
{% if demonstrations %}
Here are some examples:

{% for example in demonstrations %}
DOCUMENT:
{{ example.context }}

ORIGINAL SUMMARY:
{{ example.summary }}

SUMMARY WITH LABELED ERRORS:
{{ example.summary_with_errors }}

{% endfor %}
{% endif %}
Can you identify the errors for the following document and summary?
DOCUMENT:
{{ context }}

ORIGINAL SUMMARY:
{{ summary }}

SUMMARY WITH LABELED ERRORS:
<|end|>
```

Medalign Chain-of-thought k-shot Prompt

```
<|system|>
You are a helpful assistant that helps patients understand their medical records.
<|end|>

<|user|>
We will present you with a pair of a brief hospital course (BHC) and a patient after visit
     summary (AVS). The AVS is also referred to as discharge summary. The BHC contains a
     detailed summary of the hospital stay written by medical service. It usually contains
     medical jargon, and it can follow different structures based on the hospital course and
     responsible medical specialty. The AVS summarizes the hospital stay for the patient in
     plain language. In practice, the BHC is not the only source of information to write the
     AVS. However, in our setting we treat the BHC as the only context for the summary.

For this labeling task, we are interested in errors in the AVS that are either unsupported by
     the BHC, contradict content in the BHC, or are wrong medical facts. We allow statements
     that contain general medical knowledge or advice that are often used in patient
     summaries. Most errors are due to unsupported facts, so we further distinguish those
     based on their specific content. This leads to the following error types or labels:
1. Unsupported facts, including
     condition/procedure/medication/time/location/number/name/word/other
2. Contradicted fact
3. Incorrect fact
And below is the detailed guideline, and we label error spans with the <error> tag (e.g.
     <error>incorrect fact</error>).

### Allowed General Medical Knowledge and Medical Advice
We allow general medical knowledge and advice that is often part of the AVS. Usually, these
     are information that are not specific for the hospital course given in the BHC. For
     example
- Please take your medications as prescribed contains no error even though the BHC does not
     contain this instruction because this is general medical advice.
- If the symptoms get worse, please contact your doctor contains no error even when the BHC
     does not contain this fact, since it is general medical knowledge that a doctor should
     be seen for worsening symptoms.

### Determining Span of Errors
We label the smallest possible consecutive span that specifies the error given the BHC as a
     context. Removing further parts from the span would remove important information. A
     useful heuristic is to identify the minimal span that must be replaced to obtain a
```

```
    correct statement that is grammatically correct. For example
- We performed an <error>esophageal-gastro-duodenoscopy (EGD).<error> when no such procedure
    is reported in the BHC. The article an is not labeled as an error. When no procedure at
    all was performed performed an esophageal-gastro-duodenoscopy (EGD) should be labeled as
    error because there is no suitable substitute for esophageal-gastro-duodenoscopy (EGD).
- After the surgery, we <error>transitioned you to oral oxycodone</error>. when the BHC
    contains no information for such a transition. If another medication transition is
    mentioned in the BHC and makes sense in this sentence only oral oxycodone should be
    labeled. If another oral medication transition is mentioned in the BHC only oxycodone
    should be labeled.
- <error>Your symptoms responded well</error>. when no part of the sentence makes sense in
    the given context of the AVS.

### Dealing with Deidentified Information
The data contains deidentified information shown with ___ in the text. We always treat this
    as non-existent information. So, the annotators should not infer what the deidentified
    information could be. In general, deidentified fields in the AVS should not be labeled
    as errors. However, sometimes they belong to a wrong statement or clearly contain
    unsupported information (e.g., a doctor's name or phone numbers) that are not given in
    the BHC. In these cases, deidentified fields should be included in the error span. For
    example
- Take ___ <error>200mg daily</error> and try to rest when no such dosage information is
    provided in the BHC, but the statement to rest. The deidentified medication name is
    excluded from the error span.
- Please avoid going up <error>more than ___ stairs</error> at a time when restrictions for
    the number of stairs taken at a time are note mentioned in the BHC.
- <error>Dr. ___ will follow up with you</error> when no follow-up is mentioned in the BHC.
- Please stop taking Aspirin <error>on ___</error> when no stopping date is given in the BHC.
- Your RBC peaked <error>at ___ million</error> if there is no hint of a specific red blood
    cell count given in the BHC.

### One Error per Span
To get reliable error counts a span should only contain a single error.
- You received <error>Tylenol</error> and <error>Ciprofloxacin</error> when there is no
    evidence in the BHC that the two medications were administered to the patient.
- You have a <error>follow-up appointment with your PCP</error> and <error>your
    cardiologist</error> when no such follow up is mentioned in the BHC. Both errors are
    labeled separately.

{% for example in demonstrations %}
### Example {{ loop.index }}

BHC:
{{ example.context }}

AVS:
{{ example.summary }}
{% if example.error_descriptions %}

ERRORS:
{{ example.error_descriptions }}

{% endif %}
AVS WITH ERRORS LABELED:
{{ example.summary_with_errors }}

{% endfor %}
{% if demonstrations %}

### Example {{ len(demonstrations) + 1 }}
{% endif %}

BHC:
{{ context }}

AVS:
{{ summary }}
```

```
ERROR:
<|end|>
```

Medalign Chain-of-thought K-Shot with Class Explanation Prompt

```
<|system|>
You are a helpful assistant that helps patients understand their medical records.
<|end|>

<|user|>
We will present you with a pair of a brief hospital course (BHC) and a patient after visit
     summary (AVS). The AVS is also referred to as discharge summary. The BHC contains a
     detailed summary of the hospital stay written by medical service. It usually contains
     medical jargon, and it can follow different structures based on the hospital course and
     responsible medical specialty. The AVS summarizes the hospital stay for the patient in
     plain language. In practice, the BHC is not the only source of information to write the
     AVS. However, in our setting we treat the BHC as the only context for the summary.

## Instructions

For this labeling task, we are interested in errors in the AVS that are either unsupported by
     the BHC, contradict content in the BHC, or are wrong medical facts. We allow statements
     that contain general medical knowledge or advice that are often used in patient
     summaries. Most errors are due to unsupported facts, so we further distinguish those
     based on their specific content. This leads to the following error types or labels:
1. Unsupported facts, including
     condition/procedure/medication/time/location/number/name/word/other
2. Contradicted fact
3. Incorrect fact
And below is the detailed guideline, and we label error spans with the <error> tag (e.g.
     <error>incorrect fact</error>).

### Determining Span of Errors
We label the smallest possible consecutive span that specifies the error given the BHC as a
     context. Removing further parts from the span would remove important information. A
     useful heuristic is to identify the minimal span that must be replaced to obtain a
     correct statement that is grammatically correct. For example
- We performed an <error>esophageal-gastro-duodenoscopy (EGD).<error> when no such procedure
     is reported in the BHC. The article an is not labeled as an error. When no procedure at
     all was performed performed an esophageal-gastro-duodenoscopy (EGD) should be labeled as
     error because there is no suitable substitute for esophageal-gastro-duodenoscopy (EGD).
- After the surgery, we <error>transitioned you to oral oxycodone</error>. when the BHC
     contains no information for such a transition. If another medication transition is
     mentioned in the BHC and makes sense in this sentence only oral oxycodone should be
     labeled. If another oral medication transition is mentioned in the BHC only oxycodone
     should be labeled.
- <error>Your symptoms responded well</error>. when no part of the sentence makes sense in
     the given context of the AVS.

We allow general medical knowledge and advice that is often part of the AVS. Usually, these
     are information that are not specific for the hospital course given in the BHC. For
     example
- Please take your medications as prescribed contains no error even though the BHC does not
     contain this instruction because this is general medical advice.
- If the symptoms get worse, please contact your doctor contains no error even when the BHC
     does not contain this fact, since it is general medical knowledge that a doctor should
     be seen for worsening symptoms.

We try to ignore grammatical errors in the BHC and AVS. If the original meaning can still be
     inferred (e.g. medictaions instead of medications), the most likely corrected form can
     be used. If the meaning cannot be inferred, they can be ignored in the BHC or labeled as
     Unsupported Other in the AVS.

If a sentence or phrase is repeated, then please treat it as you would any other sentence and
     highlight all errors (even if you did so in a previous sentence). For example
- Please take Tylenol. Please take Tylenol when Tylenol was prescribed in the BHC.
```

```
- Limit your <error>use of stairs</error>. Please limit <error>use of stairs</error> when
    movement was encouraged.

To get reliable error counts a span should only contain a single error.
- You received <error>Tylenol</error> and <error>Ciprofloxacin</error> when there is no
    evidence in the BHC that the two medications were administered to the patient.
- You have a <error>follow-up appointment with your PCP</error> and <error>your
    cardiologist</error> when no such follow up is mentioned in the BHC. Both errors are
    labeled separately.

### Dealing with Deidentified Information
The data contains deidentified information shown with ___ in the text. We always treat this
    as non-existent information. So, the annotators should not infer what the deidentified
    information could be. In general, deidentified fields in the AVS should not be labeled
    as errors. However, sometimes they belong to a wrong statement or clearly contain
    unsupported information (e.g., a doctor's name or phone numbers) that are not given in
    the BHC. In these cases, deidentified fields should be included in the error span. For
    example
- Take ___ <error>200mg daily</error> and try to rest when no such dosage information is
    provided in the BHC, but the statement to rest. The deidentified medication name is
    excluded from the error span.
- Please avoid going up <error>more than ___ stairs</error> at a time when restrictions for
    the number of stairs taken at a time are note mentioned in the BHC.
- <error>Dr. ___ will follow up with you</error> when no follow-up is mentioned in the BHC.
- Please stop taking Aspirin <error>on ___</error> when no stopping date is given in the BHC.
- Your RBC peaked <error>at ___ million</error> if there is no hint of a specific red blood
    cell count given in the BHC.

### Error Types
In general, we ask for the most specific error that is applicable. If there is uncertainty
    which type applies, prefer the one mentioned first in the enumeration of all error types
    shown earlier. For instance, if the error contains an unsupported medication name, the
    Unsupported medication type should be used instead of the Unsupported name type. Here is
    a detailed description of the error types:
- `Unsupported Condition`: includes unsupported symptoms, diseases, or findings of the
    patient. For example
    - You were found to have a <error>left clavicle fracture</error> when no information was
    given for this condition in the BHC.
- `Unsupported Procedure`: includes any unsupported medical procedures. For example
    - You had a <error>filter placed in your vein</error> when no intervention with a filter
    was mentioned.
- `Unsupported Medication`: contains all errors related to unsupported medications. This
    includes medication classes, substances, routes, frequencies, and dosages. For example
    - You were placed on <error>antibiotics</error> when only blood thinners were prescribed.
- `Unsupported Time`: includes all errors for unsupported time or interval statements. For
    example
    - Keep your arm in a sling for the <error>next 6 weeks</error> when no specific duration
    is given.
- `Unsupported Location`: Locations include both unsupported physical places as well as
    regions of the patient. For example
    - The patient was admitted to the <error>Acute Surgery Service</error> when no admission
    location was provided in the BHC.
- `Unsupported Number`: any number either as digits or written that are unsupported. This
    also includes words such as a and an. For example
    - Your pacemaker rate was increased to <error>50</error> when the rate of 50 is not given
    in the BHC.
- `Unsupported Name`: named entities that are not supported by the BHC. For example
    - You were seen by the <error>interventional pulmonary service</error> when no consult
    with this service was mentioned in the BHC.
- `Unsupported Word`: incorrect or inappropriate words or phrases which do not fit in any of
    the above types. For example
    - We will send you home with a <error>drain</error> in place when drain not mentioned in
    the BHC.
- `Unsupported Other`: If there is a mistake which clearly does not belong to any of the
    above categories, you may use this category as a last resort. We cannot give precise
    instructions because the other category is very broad.
- `Contradicted Fact`: This error type is independent of the content and contains all facts
    that clearly contradict information provided in the BHC. For example
```

```
    - Your pacemaker rate was increased to <error>50</error> when the context state a
      pacemaker rate of 40.
- `Incorrect Fact`: This error type is independent of the content and contains all facts that
      clearly contradict general medical knowledge or advice. For example
    - We diagnosed a seizure, and you <error>can continue driving your car</error> when no
      reason for allowing driving after a seizure is provided this contradict common medical
      knowledge.

{% for example in demonstrations %}
## Examples
### Example {{ loop.index }}

BHC:
{{ example.context }}

AVS:
{{ example.summary }}
{% if example.error_descriptions %}

ERRORS:
{{ example.error_descriptions }}

{% endif %}
AVS WITH ERRORS LABELED:
{{ example.summary_with_errors }}

{% endfor %}
{% if demonstrations %}

### Example {{ len(demonstrations) + 1 }}
{% endif %}
BHC:
{{ context }}

AVS:
{{ summary }}
<|end|>
```

Medalign Chain-of-thought K-Shot with Class Explanation and Class Aware Prediction Prompt

```
<|system|>
You are a helpful assistant that helps patients understand their medical records.
<|end|>

<|user|>
We will present you with a pair of a brief hospital course (BHC) and a patient after visit
     summary (AVS). The AVS is also referred to as discharge summary. The BHC contains a
     detailed summary of the hospital stay written by medical service. It usually contains
     medical jargon, and it can follow different structures based on the hospital course and
     responsible medical specialty. The AVS summarizes the hospital stay for the patient in
     plain language. In practice, the BHC is not the only source of information to write the
     AVS. However, in our setting we treat the BHC as the only context for the summary.

## Instructions

For this labeling task, we are interested in errors in the AVS that are either unsupported by
     the BHC, contradict content in the BHC, or are wrong medical facts. We allow statements
     that contain general medical knowledge or advice that are often used in patient
     summaries. Most errors are due to unsupported facts, so we further distinguish those
     based on their specific content. This leads to the following error types or labels:
1. Unsupported facts, including
     condition/procedure/medication/time/location/number/name/word/other
2. Contradicted fact
3. Incorrect fact
And below is the detailed guideline, and we label error spans with the <error> tag (e.g.
     <error class=error_type>incorrect fact</error>).
```

Determining Span of Errors
We label the smallest possible consecutive span that specifies the error given the BHC as a context. Removing further parts from the span would remove important information. A useful heuristic is to identify the minimal span that must be replaced to obtain a correct statement that is grammatically correct. For example
- We performed an <error>esophageal-gastro-duodenoscopy (EGD).<error> when no such procedure is reported in the BHC. The article an is not labeled as an error. When no procedure at all was performed performed an esophageal-gastro-duodenoscopy (EGD) should be labeled as error because there is no suitable substitute for esophageal-gastro-duodenoscopy (EGD).
- After the surgery, we <error>transitioned you to oral oxycodone</error>. when the BHC contains no information for such a transition. If another medication transition is mentioned in the BHC and makes sense in this sentence only oral oxycodone should be labeled. If another oral medication transition is mentioned in the BHC only oxycodone should be labeled.
- <error>Your symptoms responded well</error>. when no part of the sentence makes sense in the given context of the AVS.

We allow general medical knowledge and advice that is often part of the AVS. Usually, these are information that are not specific for the hospital course given in the BHC. For example
- Please take your medications as prescribed contains no error even though the BHC does not contain this instruction because this is general medical advice.
- If the symptoms get worse, please contact your doctor contains no error even when the BHC does not contain this fact, since it is general medical knowledge that a doctor should be seen for worsening symptoms.

We try to ignore grammatical errors in the BHC and AVS. If the original meaning can still be inferred (e.g. medictaions instead of medications), the most likely corrected form can be used. If the meaning cannot be inferred, they can be ignored in the BHC or labeled as Unsupported Other in the AVS.

If a sentence or phrase is repeated, then please treat it as you would any other sentence and highlight all errors (even if you did so in a previous sentence). For example
- Please take Tylenol. Please take Tylenol when Tylenol was prescribed in the BHC.
- Limit your <error>use of stairs</error>. Please limit <error>use of stairs</error> when movement was encouraged.

To get reliable error counts a span should only contain a single error.
- You received <error>Tylenol</error> and <error>Ciprofloxacin</error> when there is no evidence in the BHC that the two medications were administered to the patient.
- You have a <error>follow-up appointment with your PCP</error> and <error>your cardiologist</error> when no such follow up is mentioned in the BHC. Both errors are labeled separately.

Dealing with Deidentified Information
The data contains deidentified information shown with ___ in the text. We always treat this as non-existent information. So, the annotators should not infer what the deidentified information could be. In general, deidentified fields in the AVS should not be labeled as errors. However, sometimes they belong to a wrong statement or clearly contain unsupported information (e.g., a doctor's name or phone numbers) that are not given in the BHC. In these cases, deidentified fields should be included in the error span. For example
- Take ___ <error>200mg daily</error> and try to rest when no such dosage information is provided in the BHC, but the statement to rest. The deidentified medication name is excluded from the error span.
- Please avoid going up <error>more than ___ stairs</error> at a time when restrictions for the number of stairs taken at a time are note mentioned in the BHC.
- <error>Dr. ___ will follow up with you</error> when no follow-up is mentioned in the BHC.
- Please stop taking Aspirin <error>on ___</error> when no stopping date is given in the BHC.
- Your RBC peaked <error>at ___ million</error> if there is no hint of a specific red blood cell count given in the BHC.

Error Types
In general, we ask for the most specific error that is applicable. If there is uncertainty which type applies, prefer the one mentioned first in the enumeration of all error types shown earlier. For instance, if the error contains an unsupported medication name, the Unsupported medication type should be used instead of the Unsupported name type. Here is

```
    a detailed description of the error types:
- `Unsupported Condition`: includes unsupported symptoms, diseases, or findings of the
    patient. For example
    - You were found to have a <error class=unsupported_condition>left clavicle
    fracture</error> when no information was given for this condition in the BHC.
- `Unsupported Procedure`: includes any unsupported medical procedures. For example
    - You had a <error class=unsupported_procedure>filter placed in your vein</error> when no
    intervention with a filter was mentioned.
- `Unsupported Medication`: contains all errors related to unsupported medications. This
    includes medication classes, substances, routes, frequencies, and dosages. For example
    - You were placed on <error class=unsupported_medication>antibiotics</error> when only
    blood thinners were prescribed.
- `Unsupported Time`: includes all errors for unsupported time or interval statements. For
    example
    - Keep your arm in a sling for the <error class=unsupported_time>next 6 weeks</error>
    when no specific duration is given.
- `Unsupported Location`: Locations include both unsupported physical places as well as
    regions of the patient. For example
    - The patient was admitted to the <error class=unsupported_location>Acute Surgery
    Service</error> when no admission location was provided in the BHC.
- `Unsupported Number`: any number either as digits or written that are unsupported. This
    also includes words such as a and an. For example
    - Your pacemaker rate was increased to <error class=unsupported_number>50</error> when
    the rate of 50 is not given in the BHC.
- `Unsupported Name`: named entities that are not supported by the BHC. For example
    - You were seen by the <error class=unsupported_name>interventional pulmonary
    service</error> when no consult with this service was mentioned in the BHC.
- `Unsupported Word`: incorrect or inappropriate words or phrases which do not fit in any of
    the above types. For example
    - We will send you home with a <error class=unsupported_word>drain</error> in place when
    drain not mentioned in the BHC.
- `Unsupported Other`: If there is a mistake which clearly does not belong to any of the
    above categories, you may use this category as a last resort. We cannot give precise
    instructions because the other category is very broad.
- `Contradicted Fact`: This error type is independent of the content and contains all facts
    that clearly contradict information provided in the BHC. For example
    - Your pacemaker rate was increased to <error class=contradicted_fact>50</error> when the
    context state a pacemaker rate of 40.
- `Incorrect Fact`: This error type is independent of the content and contains all facts that
    clearly contradict general medical knowledge or advice. For example
    - We diagnosed a seizure, and you <error class=incorrect_fact>can continue driving your
    car</error> when no reason for allowing driving after a seizure is provided this
    contradict common medical knowledge.

{% for example in demonstrations %}
## Examples
### Example {{ loop.index }}

BHC:
{{ example.context }}

AVS:
{{ example.summary }}
{% if example.error_descriptions %}

ERRORS:
{{ example.error_descriptions }}

{% endif %}
AVS WITH ERRORS LABELED:
{{ example.summary_with_errors }}

{% endfor %}
{% if demonstrations %}
### Example {{ len(demonstrations) + 1 }}

{% endif %}
BHC:
```

487

```
{{ context }}

AVS:
{{ summary }}
<|end|>
```