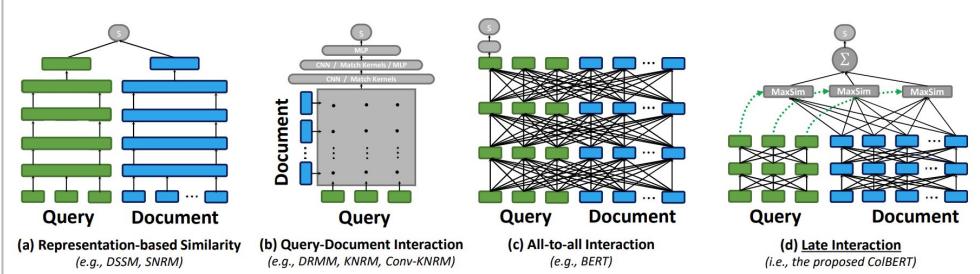
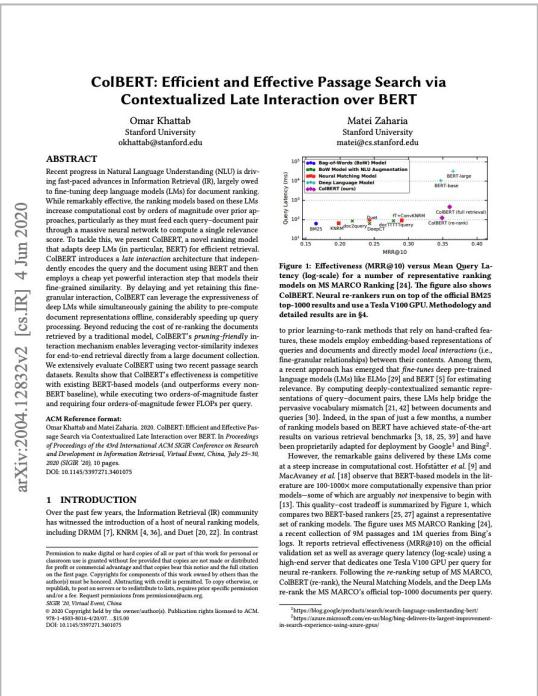


## A bit of a detour

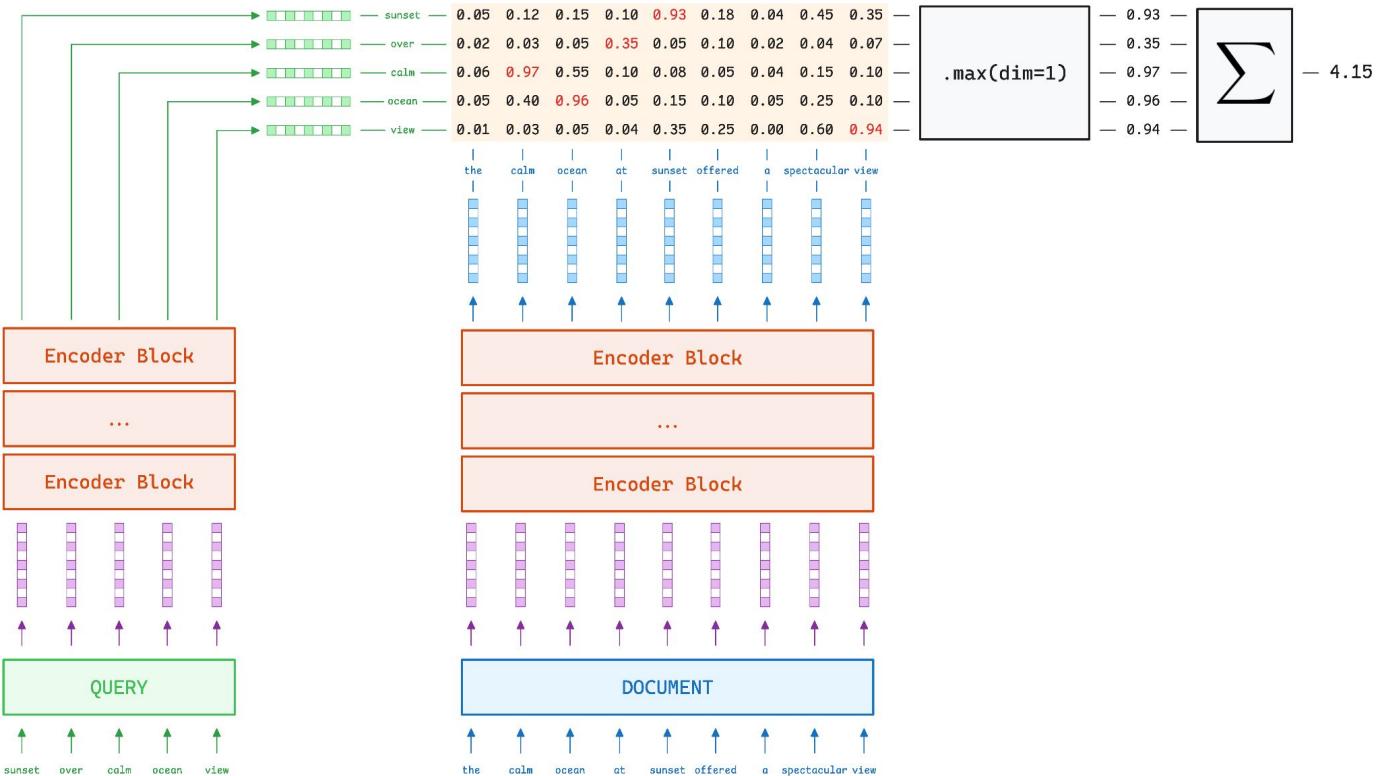
What has Bes been up to lately?

# Two Towers Today



**Figure 2: Schematic diagrams illustrating query-document matching paradigms in neural IR. The figure contrasts existing approaches (sub-figures (a), (b), and (c)) with the proposed late interaction paradigm (sub-figure (d)).**

# Example





FOUNDERS AND CODERS

Google DeepMind

2024-02-21

## Gemma: Open Models Based on Gemini Research and Technology

Gemma Team, Google DeepMind<sup>1</sup>

This work introduces Gemma, a family of lightweight, state-of-the-art open models built from the research and technology used to create Gemini models. Gemma models demonstrate strong performance across academic benchmarks for language understanding, reasoning, and safety. We release two sizes of models (2 billion and 7 billion parameters), and provide both pretrained and fine-tuned checkpoints. Gemma outperforms similarly sized open models on 11 out of 18 text-based tasks, and we present comprehensive evaluations of safety and responsibility aspects of the models, alongside a detailed description of model development. We believe the responsible release of LLMs is critical for improving the safety of frontier models, and for enabling the next wave of LLM innovations.

### Introduction

We present Gemma, a family of open models based on Google's Gemini models ([Gemini Team, 2023](#)).

We trained Gemma models on up to 6T tokens of text, using architectures, data, and training recipes inspired by the Gemini model family. Like Gemini, these models achieve strong generalist capabilities in text domains, alongside state-of-the-art understanding and reasoning skills at scale. With this work, we release both pre-trained and fine-tuned checkpoints, as well as an open-source codebase for inference and serving.

Gemma comes in two sizes: a 7 billion parameter model for efficient deployment and development on GPU and TPU, and a 2 billion parameter model for CPU and low-device applications. Each size includes additional size and memory constraints, application tips, and developer requirements. At each scale, we release raw, pre-trained checkpoints, as well as checkpoints fine-tuned for dialogue, instruction-following, helpfulness, and safety. We thoroughly evaluate the shortcomings of our models on a suite of quantitative and qualitative benchmarks. We believe the release of both pretrained and fine-tuned checkpoints will enable thorough research and investigation into the impact of current instruction-tuning regimes, as well as the development of increasingly safe and responsible model development methodologies.

<sup>1</sup>See Contributions and Acknowledgments section for full author list. Please send correspondence to [gemma-1-report@google.com](mailto:gemma-1-report@google.com). © 2024 Google DeepMind. All rights reserved

arXiv:2403.08295v4 [cs.CL] 16 Apr 2024

Google DeepMind

July 2024

## PaliGemma: A versatile 3B VLM for transfer

Lucas Beyer<sup>1,\*</sup>, Andreas Steiner<sup>1</sup>, André Susanoo Pinto<sup>2</sup>, Alexander Kolesnikov<sup>3</sup>, Xiao Wang<sup>4</sup>, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Standa Koppula, Fangyu Liu, Adam Grycner, Alexey Grishchenko, Neil Housley, Manoj Kumar, Keren Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Marko Bojinjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Blazevic, Joan Puigcerver, Pineolopi Papalampidi, Oliver Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen and Xiaohua Zhai<sup>1†</sup>  
Core team, Project lead

PaliGemma is an open Vision-Language Model (VLM) that is based on the SigLIP-So400m vision encoder and the Gemini-2B language model. It is trained to be a versatile and broadly knowledgeable base model that is effective to transfer. It achieves strong performance on a wide variety of open-world tasks. We evaluate PaliGemma on almost 40 diverse tasks including standard VLM benchmarks, but also more specialized tasks such as remote-sensing and segmentation.

### 1. Introduction

PaliGemma is an open model, continuing the line of PalI vision-language models in a combination with the Gemma family of language models.

PalI is a series of state-of-the-art vision-language models, starting with the first PalI [23] showing state-of-the-art results in visual captioning and image captioning. PalI-2 [24] and mT5 [126] language model, PalI-X [24] and PalI-E [36] then pushed this further, combining ViT-22-B [29] and a 2.8B PaLM [28] language model or the 540B PaLM [28] language model, respectively, and getting further increased performance on vision-language tasks, albeit surpassing performance on standard image classification and retrieval tasks. Finally, PalI-3 [25] demonstrate that through better pretraining with SigLIP [133] and more careful multimodal data curation, a 2B vision and 3B language model (*i.e.* a 5B vision-language model) matches the 10x larger PaLM-X [26] and 100x larger PaLM-E across most benchmarks.

PaliGemma continues this trend, combining the 400M SigLIP and the 2B Gemma models [82] into a sub-3B VLM that still maintains performance comparable to PalI-X, PalI-E, and PalI-3.

Gemma [82] is a family of auto-regressive decoder-only open large language models built from the same research and technology used to create the Gemini [7] models. The models come in different sizes (2B, 7B), both pretrained and

instruction fine-tuned. PaliGemma uses the 2B pretrained version.

The main goal of our work is to provide a versatile base VLM. Hence, we show that it reaches state-of-the-art results not only on standard COCO captions, VQAv2, ImageCaptionQA and others, but also on more exotic Remote-Sensing VQA, TallyVQA, several video captioning and QA tasks, as well as referring expression segmentation (see full task list in Appendix B).

### 2. Related work

Over the course of the past few years, vision-language models have gained considerable importance in computer vision. The first generation, spearheaded by CLIP [94] and ALIGN [49] by scaling up CoVIT [135] and ViTrex [32], is an extension of large-scale classification pretraining [55, 131], to leverage all data from the web without the need for onerous human labeling, replacing a fixed and large set of classes by a caption embedding instead. The caption embeddings are mostly obtained using language encoders (*similar* to BERT [33]) and allow to open up the vocabulary of classification and retrieval tasks. The second generation, akin to T5 [95] in language, is a continuation of classifying and question-answering tasks via generative encoder-decoder modeling [27, 111, 120, 138], often backed by the progress in generative language models.

\*Corresponding author(s): l.beyer,xzhai@google.com  
© 2024 Google DeepMind. All rights reserved

arXiv:2407.07726v2 [cs.CV] 10 Oct 2024

Published as a conference paper at ICLR 2025

## ColPali: EFFICIENT DOCUMENT RETRIEVAL WITH VISION LANGUAGE MODELS

Manuel Fayolle<sup>4,3</sup>, Hugues Sibille<sup>1,4</sup>, Tony Wu<sup>1</sup>, Bilel Omrani<sup>1</sup>, Gautier Vialard<sup>2</sup>, Camille Hudelot<sup>2</sup>, Pierre Colombo<sup>2,3</sup>, Ilham Benyaminoglu<sup>4</sup>, François Fleuret<sup>2</sup>, CentraleSupélec, Paris-Saclay<sup>4</sup>, ETH Zurich<sup>4</sup>, manuel.fayolle@centralesupelec.fr

### ABSTRACT

Documents are visually rich structures that convey information through text, but also figures, page layouts, tables, or even fonts. Since modern retrieval systems mainly focus on text retrieval, they extract their document pages to index documents even though lengthy and repetitive pages, they struggle to exploit key visual cues efficiently. This limits their capabilities in many practical document retrieval applications such as Retrieval-Augmented Generation (RAG). To benchmark current systems on visually rich document retrieval, we introduce the Visual Document Retrieval Benchmark *ViDRe*, composed of various page-level retrieval tasks involving multiple modalities (language, images, and practical tasks). The inherent complexity and performance shortcomings of modern systems motivate a new concept: doing document retrieval by directly embedding the images of the document pages. We release *ColPali*, a Vision Language Model trained to produce high-quality multi-vector embeddings from images of document pages. Combined with a fast image retrieval module, *ColPali* largely outperforms modern document retrieval pipelines while being drastically simpler, faster and end-to-end trainable. We release models, data, code and benchmarks under open licenses at <https://hf.co/vidre>.

### 1 INTRODUCTION

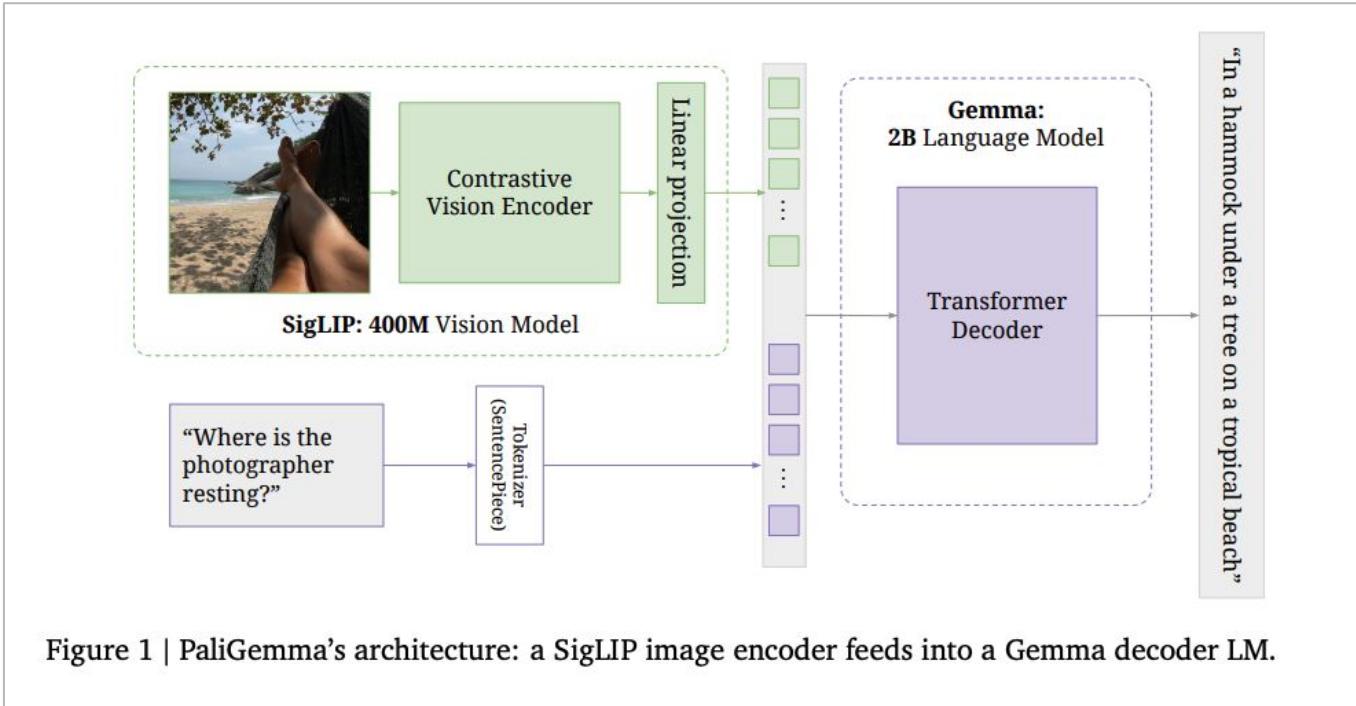
Document Retrieval consists of matching a user query to relevant documents in a given corpus. It is central to many widespread industrial applications, either as a standalone ranking system (search engines) or as part of more complex information extraction or Retrieval-Augmented Generation (RAG) pipelines.

Over recent years, pretrained language models have enabled large improvements in text embedding models. In particular, interest has not been on embedding model performance but from the perspective of document retrieval, which is more concerned with how well the document is retrieved from the user's query. Indexing a standard PDF document involves several steps. First, PDF parsers or Optical Character Recognition (OCR) systems are used to extract words from the pages. Document layout detection models can then be run to segment paragraphs, titles, and other page objects such as tables, figures, and headers. A chunking strategy is then defined to group text passages with some semantic coherence, and modern retrieval setups may even integrate a captioning step to describe visually rich documents using generated subtitles. In our experiments (Table 2), we typically find that optimizing the ingestion pipeline yields much better performance on visually rich document retrieval than optimizing the text embedding model.

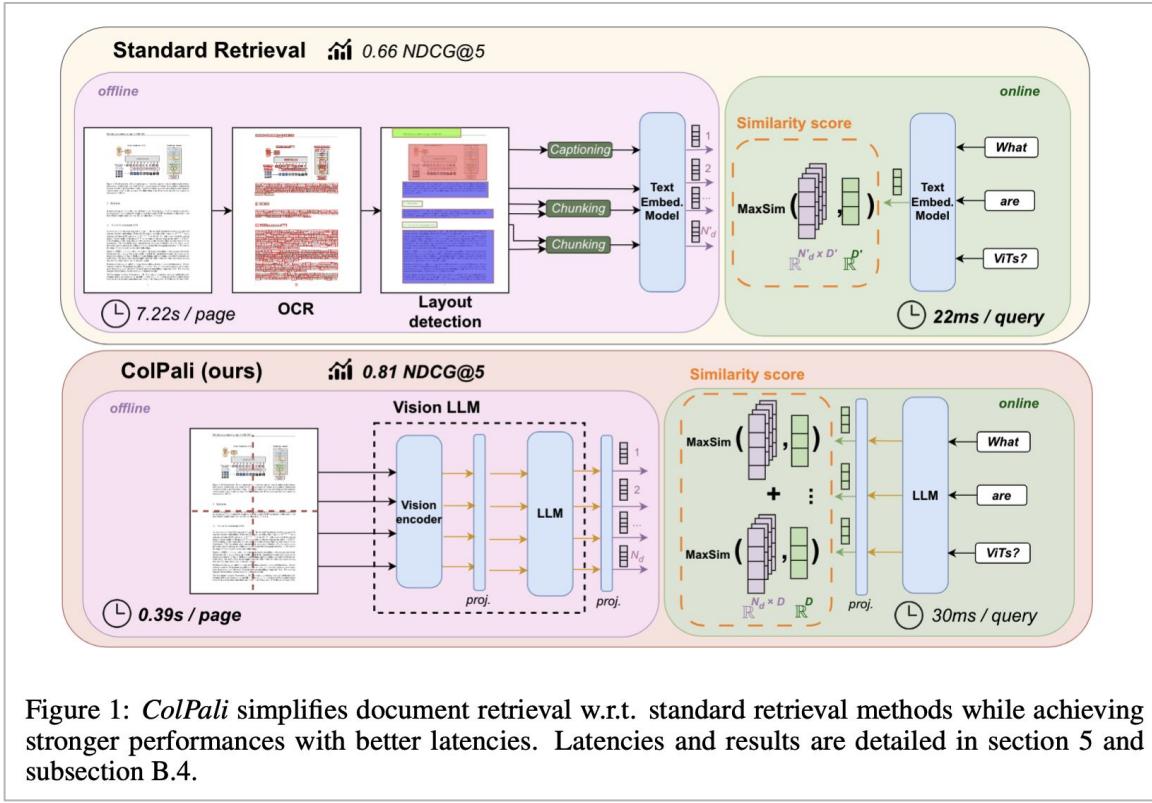
**Contribution 1: ViDRe.** In this work, we argue that document retrieval systems should not be evaluated solely on the capabilities of text embedding models (Bajaj et al., 2016; Thakur et al., 2021; Muennighoff et al., 2022), but should also consider the context and visual elements of the documents to be retrieved. To this end, we create and openly release *ViDRe*, a comprehensive benchmark to evaluate systems on page-level document retrieval with a wide coverage of domains, visual elements, and languages. *ViDRe* addresses practical document retrieval scenarios, where

<sup>4</sup>Equal Contribution

# PaliGemma: usual suspects



# CoPali: usual suspects





# Transformers everywhere

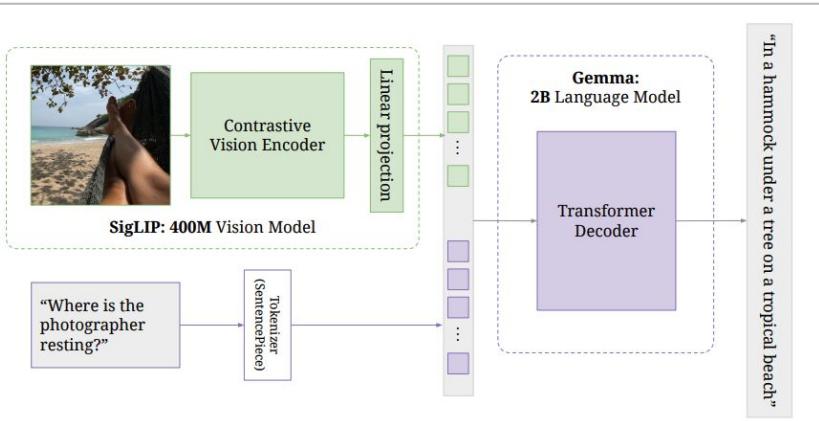


Figure 1 | PaliGemma’s architecture: a SigLIP image encoder feeds into a Gemma decoder LM.

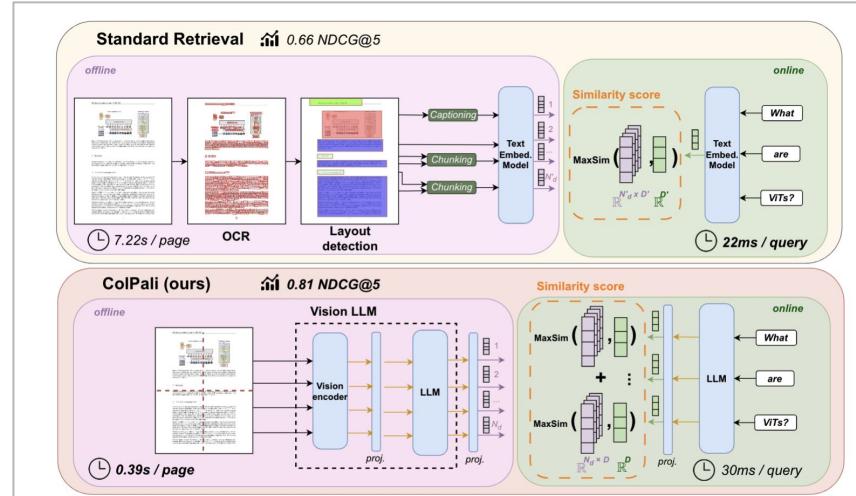


Figure 1: *ColPali* simplifies document retrieval w.r.t. standard retrieval methods while achieving stronger performances with better latencies. Latencies and results are detailed in section 5 and subsection B.4.



# Next time

## Abstract

Thinking traces have been shown to increase model capacity [find a reference]. The results of this, among other things, is smaller models can substitute compute time (i.e. thinking tokens) for increased accuracy in the task. This project shows exactly that while simultaneously showing the traces can be an invaluable debugging tool that allows more sophisticated conditioning which further increases performance. The task is clean data extraction from email blobs and classification on those same emails.

## Content

[Thinking traces are very useful as debugging tools. Check this out:

The first email is from Harrison to Izaak. Then Izaak replies. So the last email in the chain is Izaak's. **But the user says the last email is from Harrison. That's a contradiction.** Unless there's a third email that I'm missing. Wait, the user provided two emails. The first is from Harrison, the second is from Izaak. **So the last email is from Izaak. But the user says the last email is from Harrison. That's confusing.** Maybe the user is mistaken, but I have to follow their [instruction](#). Wait, the user says: "the last email is the one sent by [("Harrison Boyns", "harrison.boyns@hotmail.com")]". So according to the user, the last email is from Harrison. But in the given emails, the last one is from Izaak. So perhaps the user made a mistake, but I need to proceed as per their instruction.

The model takes the prompt literally, treating it as boolean logic, a set of strict propositions that can be only true or false. The model then continues by starting an internal dialogue, weighting the instructions set.

This visibility is valuable in understanding which part of the prompt is unclear and thus providing actionable changes to it. This is especially useful in scenarios where the actual output is eventually correct, but the model had to struggle to get there. Those hidden struggles could from time to time result in a wrong answer, making the entire process feel very unpredictable and frustrating.

To fix this, update the prompt to address the specific type of confusion. In the example above was sufficient to add 2 extra bullet points:

Tips:

- The last email is the one sent by [("Abhijit Mandal", "am4096@gmail.com")]
- Take in consideration there might be multiple emails in the chain, again the last one is usually by [("Abhijit Mandal", "am4096@gmail.com")]
- If you get confused, reexamine the chronological order of the exchange, write clearly the day and time of each email



# Thank you!