sama

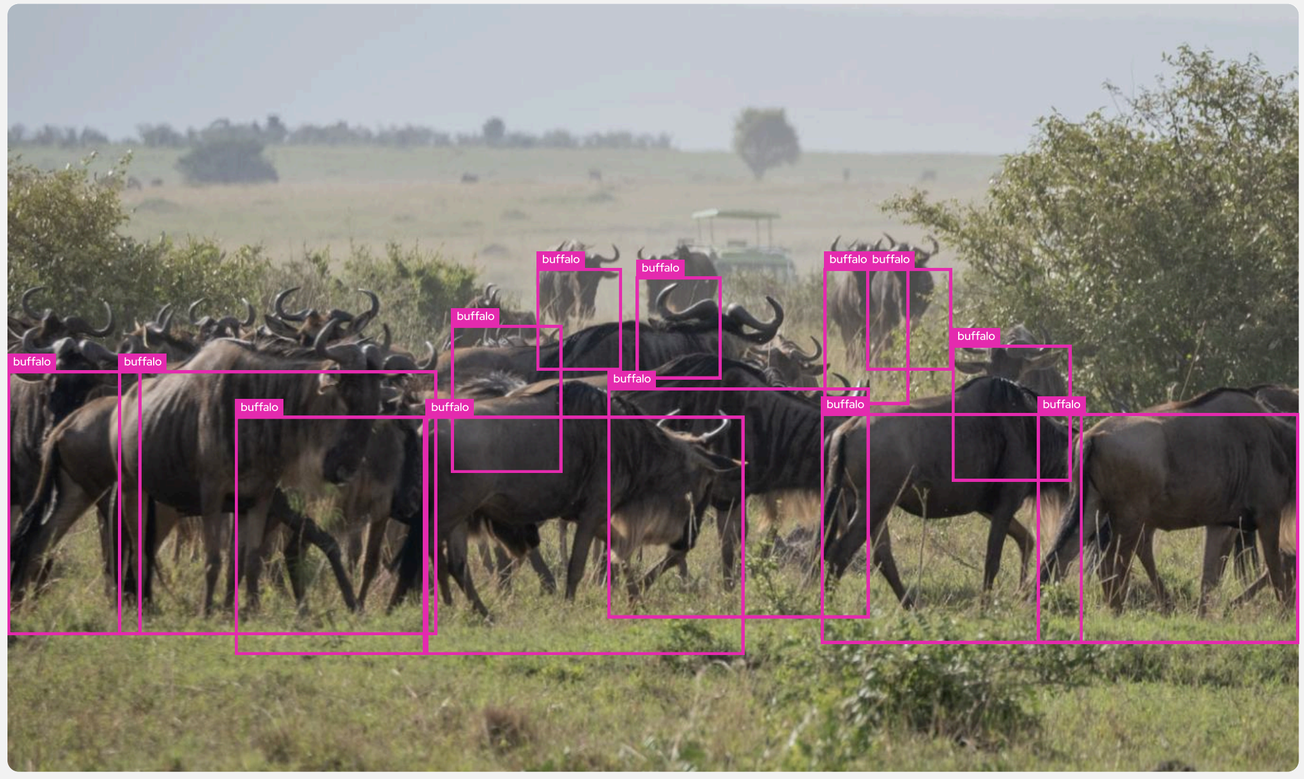
Machines Still Need Us

Human Annotation in ML Development Is **More Crucial Than Ever**

With the recent advancements in LLMs, discussions about automating the annotation process have become increasingly common. It's easy to fall into the trap of assuming that ML models can simply replace the annotation and labeling work traditionally done by humans.

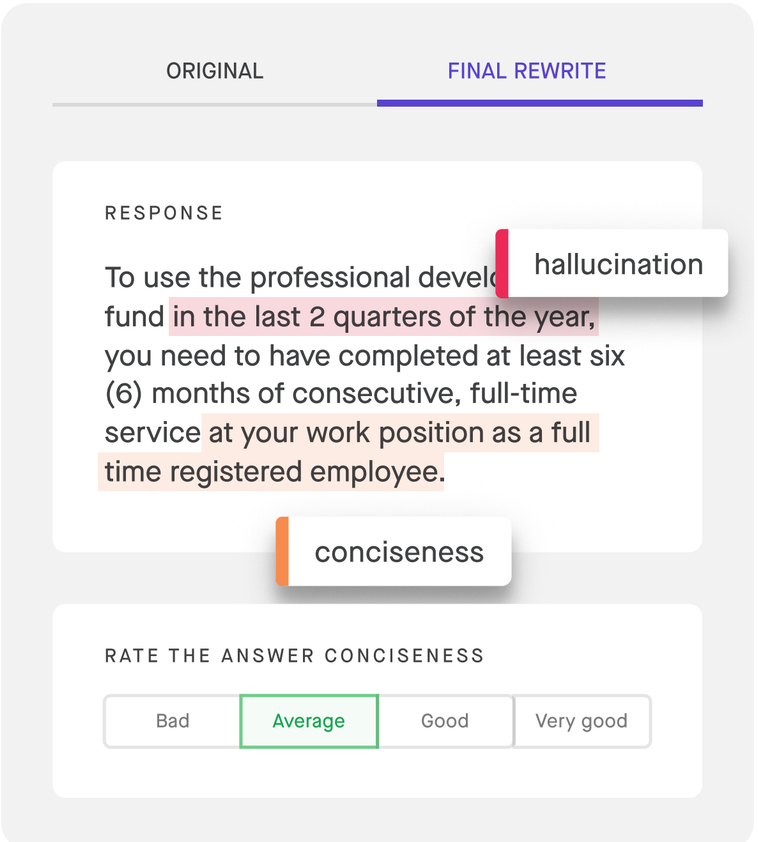
Our Machine Learning (ML) team at Sama considers—and receives—questions about shifting to automated annotation all the time. This paper explains the context, quality, and desired outcomes you should consider for your business when the topic arises.

Our clients expect us to deliver the highest quality labels for the data they share with us. For computer vision applications, this might involve creating precise, well-identified bounding boxes around objects of interest. In Natural Language Processing (NLP) or Generative AI (Gen AI), it often entails correcting text responses from Large Language Models (LLMs) for factual accuracy or ensuring they meet specific criteria such as tone, correctness, and helpfulness.



Computer Vision (CV) annotation example: Objects of interest are identified by the annotation team with bounding boxes or polygons.

Large Language Model (LLM) annotation example: A model response is scrutinized for factual errors or any properties that don't fit a set of pre-defined requirements (e.g., correctness, helpfulness, format, length, etc.)



Common Questions about **Automation in Annotation**

Inquiries about automation often revolve around techniques aimed at achieving the same high-quality annotations produced by human annotators, but through automated means.

The perception is that human annotation is slow, expensive, and prone to error, while automation is seen as cheaper, faster, and increasingly capable. This understandably raises questions—or at the very least curiosity—among model developers. However, the perceived potential of automation in labeling is often overestimated or, at the very least, misconstrued.

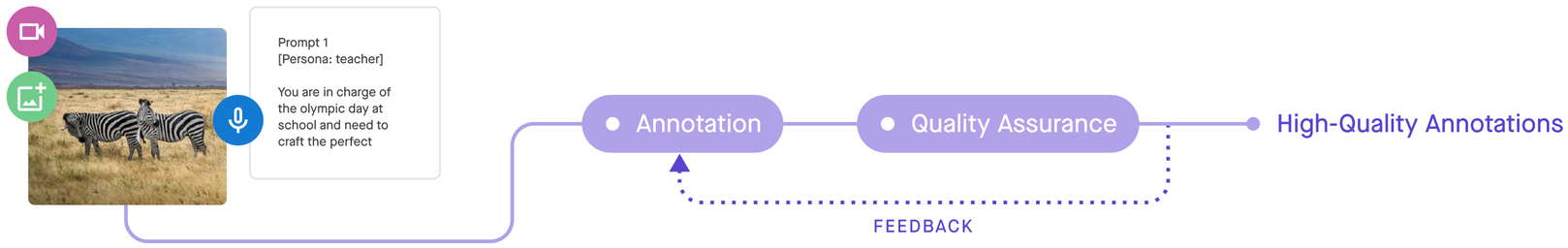
Can models directly annotate the data our clients require?

For instance, in Computer Vision, could we use existing models to generate annotations like bounding boxes around objects in images, leveraging off-the-shelf object detection models like the YOLO series? Or, in NLP, can an LLM perform direct data annotation, such as executing sentiment analysis based on specific instructions?

Can models be used to perform quality assurance on annotations?

A significant portion of professional and large-scale annotation workflows is dedicated to catching and correcting errors made by the first line of annotators, who label the bulk of the data at a rapid pace. While experienced annotators are generally reliable, errors are inevitable. This is why there often exists within these workflows a built-in second line of quality assurance to catch any remaining mistakes and continuously improve the training and performance of the first line of annotators. Could we automate this Quality Assurance phase using state-of-the-art machine learning? For instance, can we use LLMs as fact-checkers directly?

Simplified annotation workflow: In this process, data is initially handled in bulk by the first line of annotators, who produce the initial annotations. A second line of annotators then reviews the work for quality, making necessary corrections or providing feedback to the first line for further refinement. This dual-layered approach ensures that the final output annotations meet the highest standards of quality. The ratio of first-line annotators to second-line reviewers can vary significantly, ranging from 4:1 to 100:1, depending on the complexity of the task.



The Role of **Human Input** in Model Development

While the idea of replacing human annotators with state-of-the-art models might seem logical, it is based on a misconception about how most Machine Learning systems operate. Most modern ML models function as "closed systems" that learn patterns, structures, and relationships from a fixed dataset during their training and fine-tuning phases. Despite the massive scale of these datasets—often encompassing significant portions of the publicly accessible internet as well as proprietary, domain-specific data—these models retain a fixed amount of knowledge and capabilities. This knowledge is often mixed with a certain level of noise due to the inherent variability and inaccuracies in the data they are trained on. It's no secret that the information available on the internet can be noisy, inaccurate, or even deliberately misleading. (The same can apply to proprietary datasets.)

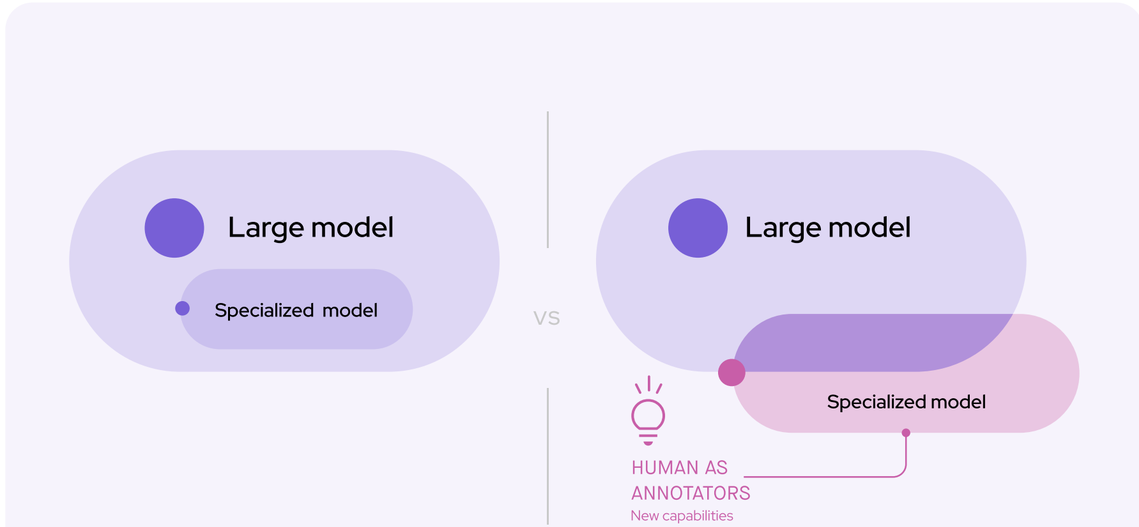
Acknowledging these misconceptions means we must reframe how to improve a model. A model's output won't improve simply by feeding in any amended annotations of the same data. Improvement only comes with new knowledge: adding context, making corrections, adding in false negatives, and other amendments that can—for now—only be injected by human annotators.

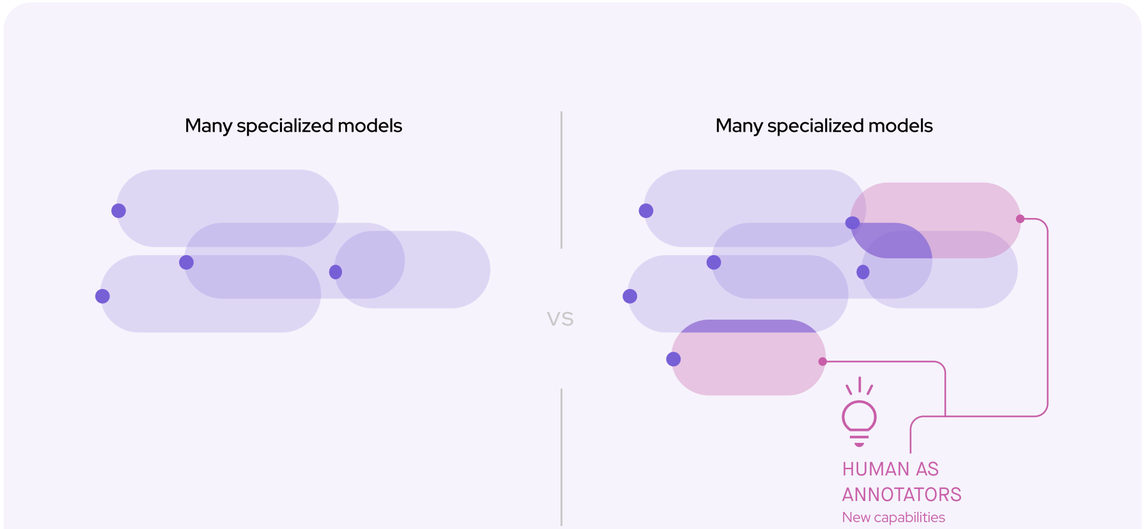
The challenge is to enhance what begins as a hermetic and static model, which the client hopes to improve with each retraining phase by ingesting new, annotated data or corrected annotations—in other words, by injecting new and structured information into the system to address some of its remaining flaws or shortcomings.

Automation techniques like pre-annotations—where data is run through an ML model to generate predictions before being handed to annotators for final adjustments—can streamline the annotation process. However, these techniques often fail to fundamentally enhance a model's capabilities—unless they are paired with subsequent steps that introduce human knowledge in the form of new annotations or corrections to the model's predictions. Techniques like model distillation, where one "super knowledgeable model" is used to train another, can also help, but human annotations often remain essential for injecting new information into the system, and providing the data necessary for the next iteration of model training.

When replacing humans with models in the annotation workflow, you might improve efficiency, but you are unlikely to enhance the model's capabilities. The same information is simply reorganized more efficiently; the amount of knowledge or capabilities "within the box" remains largely unchanged.

This is somewhat analogous to asking elementary school students to correct each other's quiz papers before providing them with the correct answers. While their collective knowledge might become better distributed—since some students may understand certain aspects of the material better than others—no new knowledge is created, and any concept that wasn't understood before is unlikely to be grasped suddenly. It's not until the teacher steps in to correct the remaining errors and provide feedback or teach new concepts that genuine learning occurs. In this analogy, human annotators are the teachers, and the models are the students.





Model Distillation: Using the predictions of one large model as training annotations for a more specialized model—a technique known as model distillation—does not introduce new capabilities or knowledge to the specialized model. To expand the model's capabilities, human annotations or corrections are essential.

Multi-Model Distillation: Using the predictions of multiple existing models as training annotations for a more specialized model is also possible, but this process does not inject any new capabilities or knowledge into the resulting model. **To achieve that, there is no substitute for human annotations to generate more quality data.**

What about **models as agents?**

At this stage of the conversation, it's important to recognize that not all state-of-the-art ML models function as entirely "closed systems," limited to learning solely from the data they are provided during training. There is ongoing, active research focused on developing AI agents capable of autonomously seeking external information to enrich their knowledge base or to support additional capabilities through the use of external tools, such as accessing a calculator. Moreover, these models might query databases, interact with external APIs, or engage in other forms of learning.

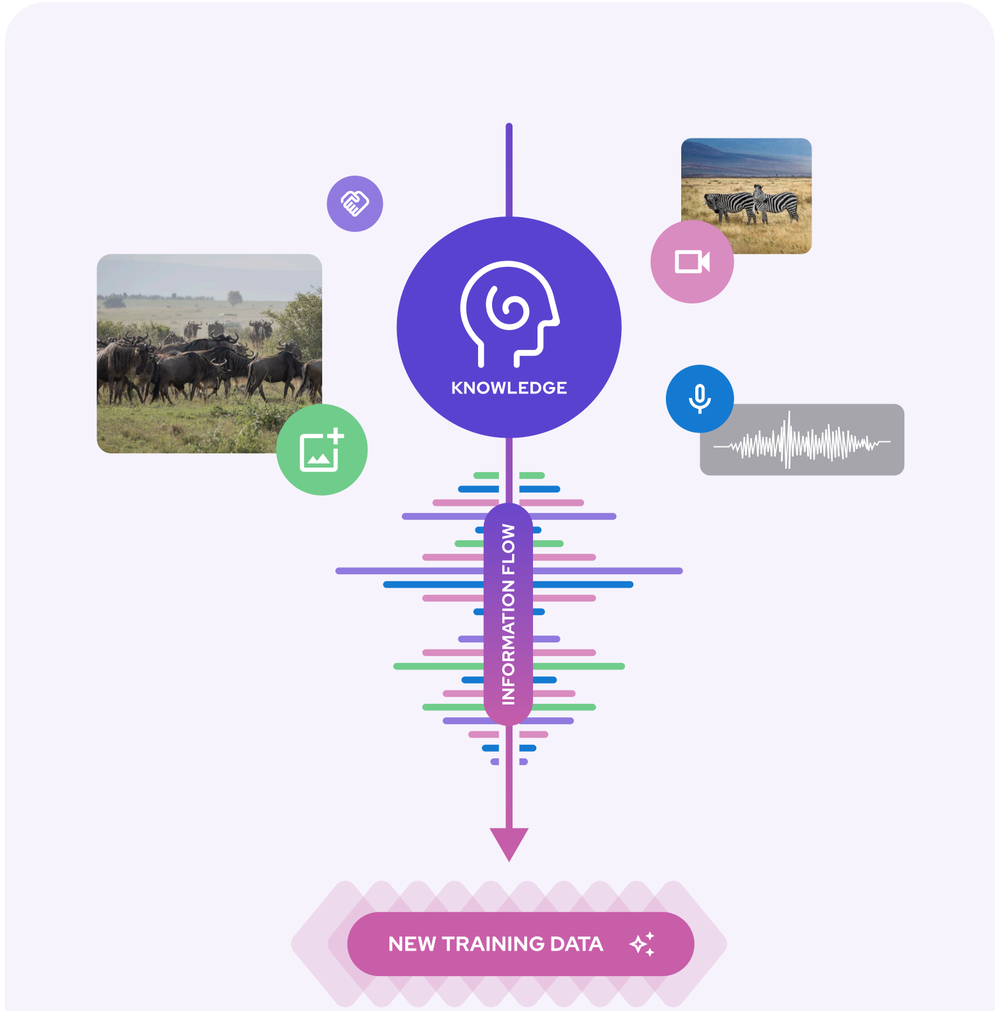
While these advancements offer promising avenues for injecting new knowledge into ML systems, it's crucial to understand that these capabilities are still under exploration. **More importantly, they do not eliminate the need for human knowledge injection.** Instead, they shift where this human input is required—both upstream and downstream. Upstream, because models will need carefully curated examples to learn how to use these tools effectively. Downstream, because humans will still need to validate that the models' use of these tools is achieving the desired outcomes and make the appropriate edits to the data when they aren't.

Transforming **Human Knowledge** into Annotation Data

So, how *should* models be used to assist with annotations?

The answer lies in facilitating the process of effectively transferring human knowledge into "the system." Imagine this process as a virtual pipeline where human understanding and expertise, originating in the human brain, are transduced into training data in the form of annotations, corrections, and insights. This information flow is crucial to improving model performance and extending capabilities, but it faces two distinct types of resistance.

First, it's not always clear what new information should be transferred through this pipeline. Human annotators often lack detailed insight into specific aspects of a model's performance that require enhancement. For instance, a computer vision model might already be proficient at detecting bicycles, so adding more bicycle annotations would offer minimal benefits. Conversely, the model might struggle with identifying motorcycles, particularly those with multiple passengers in urban settings. While model designers (on the client side) might have a general sense of these strengths and weaknesses, they may not be able to specify them with enough precision to guide the annotation process effectively. Therefore, the challenge is not just about ensuring a seamless flow of information from human to data, but also making sure that the information being transferred is both rich and relevant - targeting the areas where it can make the most impact.



Information flow from human to training data: Enhancements to the model are directly tied to how effectively human knowledge is transferred into the training data. It's crucial that the information injected by humans is both accurate and relevant, specifically targeting areas that will be most beneficial during the retraining phase. Additionally, the process must be optimized to preserve the quality of this information and maximize its transfer efficiency, ensuring that the model can fully benefit from the enriched data.

Second, and perhaps most importantly, the process of packaging and delivering new information is often more challenging than producing the information itself. For example, in an image captioning task where annotators must describe an image through text, it might be relatively straightforward for the annotator to identify all the important details visually. However, transforming those visual insights into a coherent, well-structured and well-written paragraph that accurately captures the scene is cognitively taxing.

Example of Friction in Human-to-Data Information Flow:

To understand the cognitive friction faced by annotators, consider the difference between mentally processing the important details of this scene in your head and the mental challenge of translating those thoughts into two well-written, cohesive paragraphs that effectively communicate the scene to all readers.

Similarly, it is easy for most **people** to differentiate the skier from other objects and the background in the scene. However, for a **machine** to recognize the same image, one must carefully segment the exact contour of the skier's silhouette using existing editing tools—which can be a very time-consuming and taxing task.

Similarly, in a computer vision task, detecting a partially occluded object might be simple, but precisely segmenting that object - delineating it from obstructing elements and the background using editing tools - can be painstakingly difficult and time-consuming. These examples illustrate that while the production of raw information might not be inherently complex, the cognitive effort required to transduce it into a usable form can be substantial.

CAPTION:
This image captures a dynamic and intense moment of a skier mid-fall on a snowy slope. The skier is dressed in an orange and white outfit with black gloves and a backpack. They are in the middle of a dramatic tumble, with their skis crossed in the air and their body leaning back as they attempt to regain balance. Snow is being kicked up around them, indicating the speed and force of the fall. The backdrop features a mountainous landscape with snow-covered peaks and trees, under a clear blue sky, adding to the dramatic atmosphere of the scene.

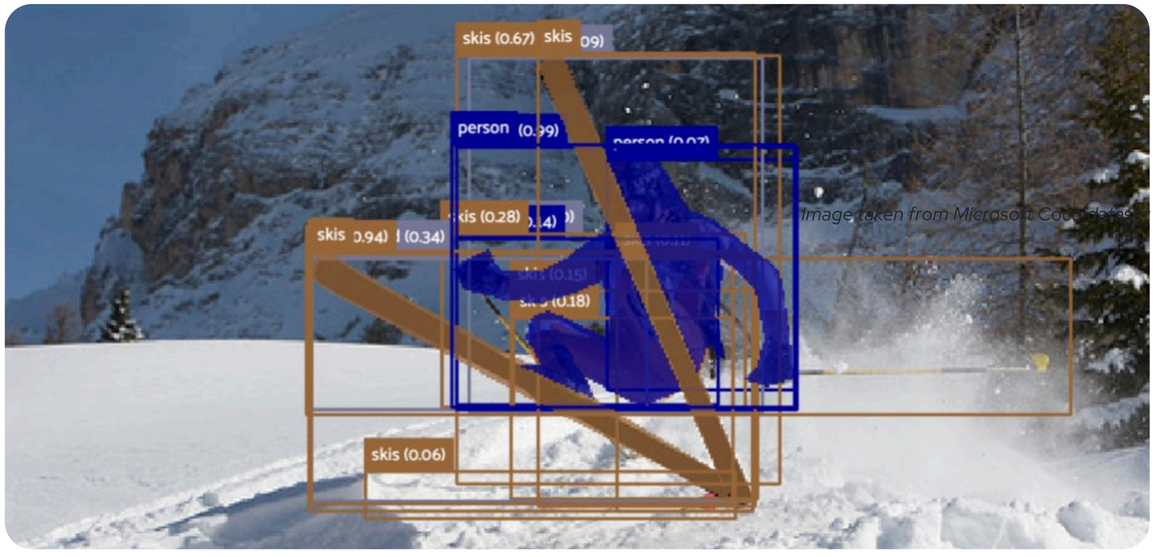
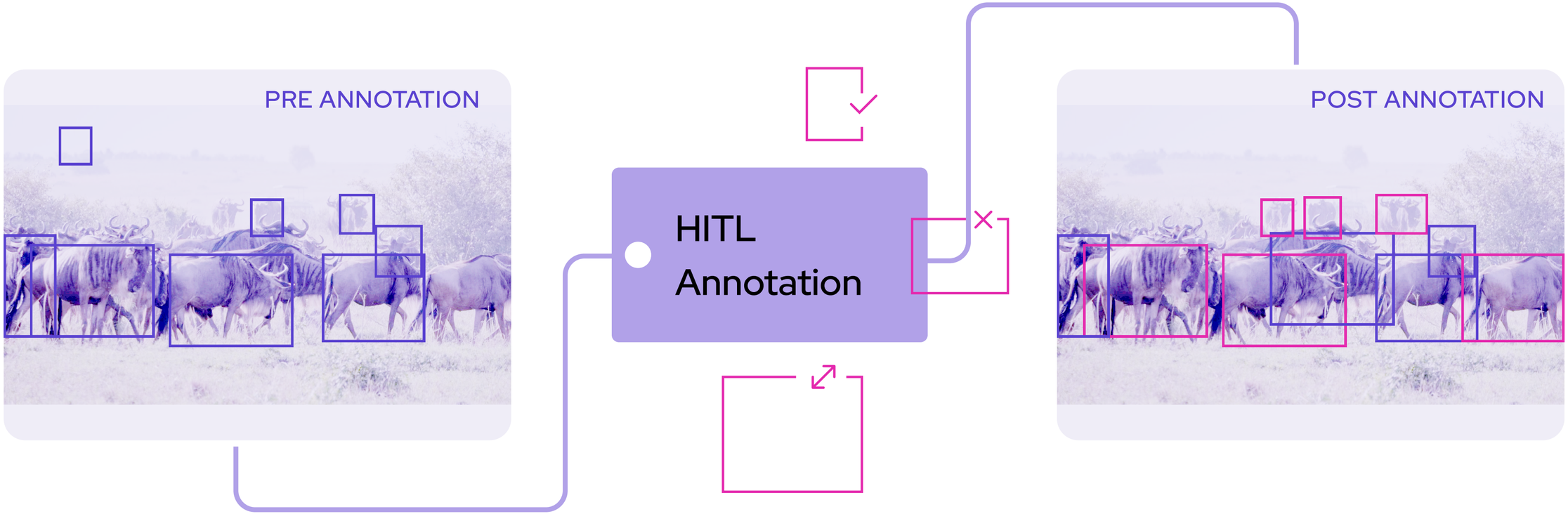


Image taken from Microsoft COCO dataset

Enhancing **Annotation Efficiency**



- **Pre-annotation workflow**: In this workflow, client images are first processed through a Machine Learning model (e.g., an object detector) to generate pre-annotations. However, these pre-annotations alone are typically insufficient for producing rich training data that can significantly enhance the client model's capabilities. It is only when human annotators validate correct predictions, adjust or remove incorrect annotations (e.g., resizing or deleting bounding boxes), and add missing elements (in cases of false negatives) that new, valuable information is truly injected into the system, enhancing the model's overall capabilities.

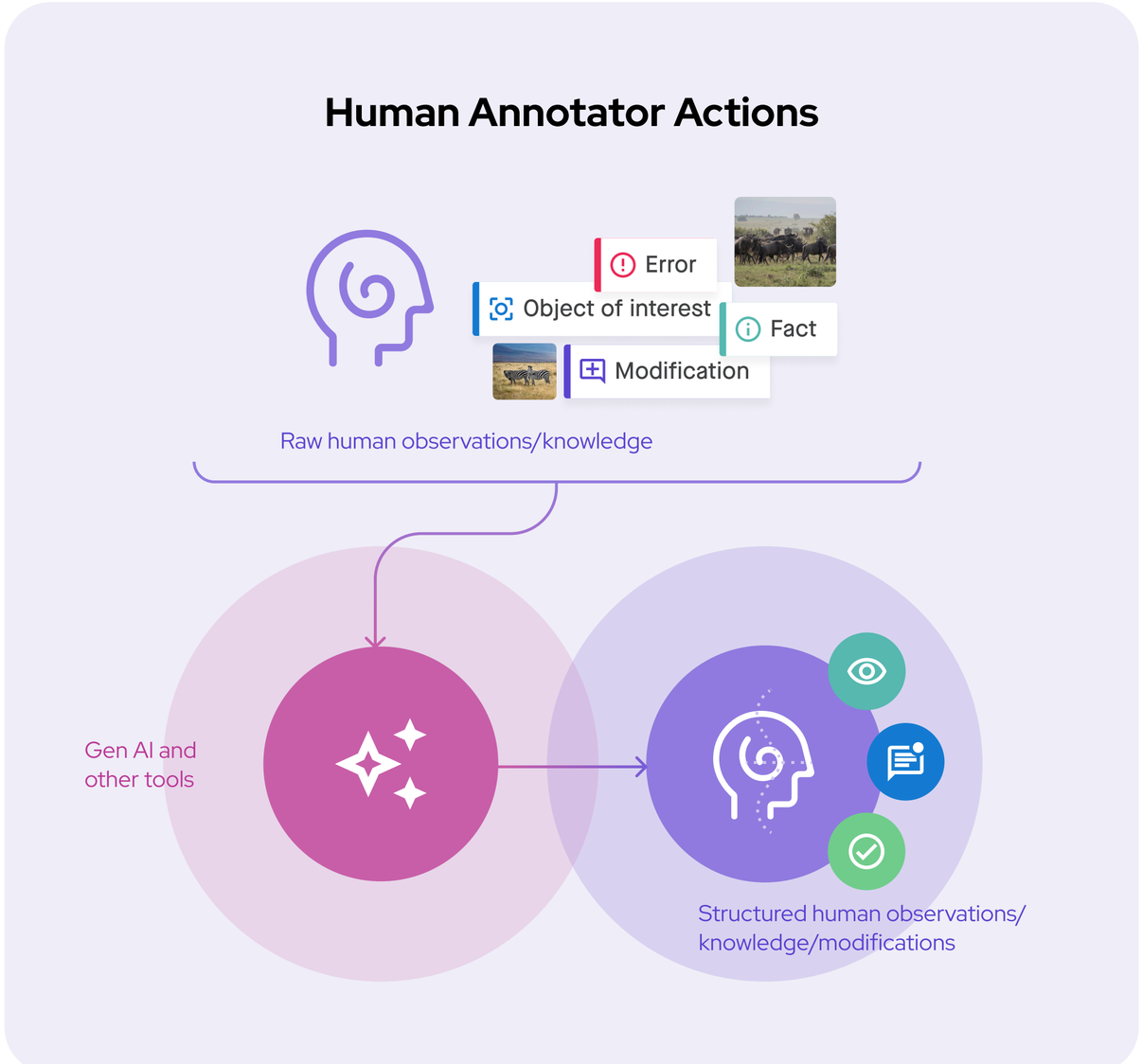
For the first challenge of identifying what new information is needed, numerous techniques can help pinpoint which data from a large pool is most valuable to annotate for a particular model. These techniques typically fall under the umbrella of data curation.

An even simpler method involves pre-annotating tasks with existing models and asking annotators to correct the errors made by the model. This approach ensures low-value annotations are avoided (since the model is already getting them right) and focus is instead placed on areas where the model performs poorly. This pre-annotation strategy is most effective when the client model is used to generate the pre-annotations, as it directly highlights areas needing improvement.

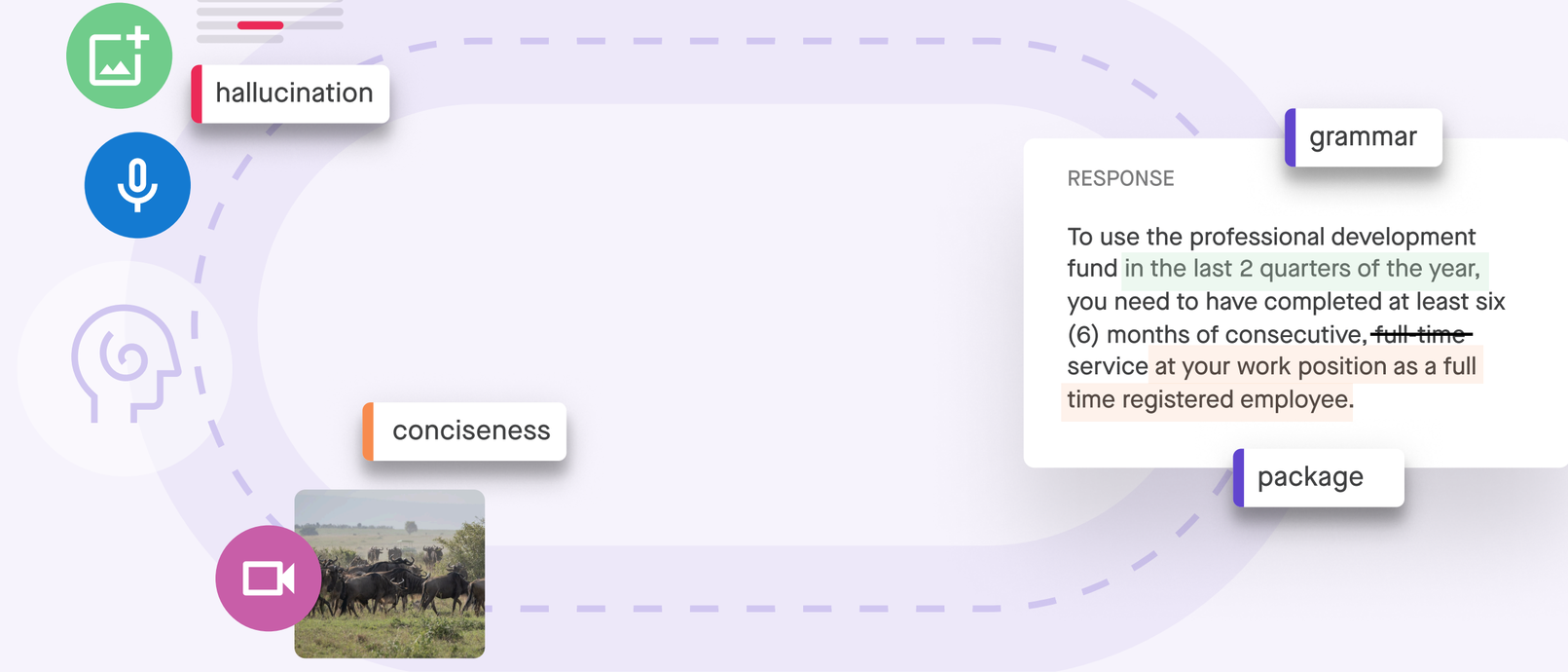
However, it can also work with a proxy model, as there is often significant overlap between the areas where both the client and proxy models struggle.

Much like in that class of students, concepts not fully understood by one student are likely misunderstood by others. Pre-labeled tasks are common in GenAI, for instance, when annotators are asked to validate or edit LLM responses according to a prompt rather than writing them from scratch. In computer vision use cases, this approach is becoming more common, though it is often driven more by the client's desire to reduce annotation costs than by a focus on generating high-value information.

Another essential aspect of ensuring that the right data is being transferred involves effectively managing annotation instructions. As models evolve, the instructions given to the annotating workforce may need to be adjusted, often requiring a higher level of specificity. This is particularly important when dealing with a large group of annotators, where consistency and clarity are critical. Being nimble in updating and refining these instructions helps to ensure that the annotations remain aligned with the model's current needs, maximizing the value of the data being captured.



ML tools can also greatly assist with the second challenge: relieving human annotators from tasks that strain their cognitive resources without adding significant new value. This is where LLMs can truly shine. They can take the raw output from humans and refine it into something more succinct, digestible, and structured. In this information flow system, humans-in-the-loop should be seen as the entities that extract or inject new raw data (left side of image below), while LLMs should be leveraged as tools that package, organize, and structure this raw data, perhaps even identifying areas where more details or clarification are needed (right side of image). Annotators shouldn't have to worry about aspects like composition, sentence structure, grammar, punctuation, avoiding redundancy, maintaining a consistent voice, or ensuring seamless sentence flow. Instead, they should focus on clearly identifying the errors or omissions made by the models.



Optimizing the **Human-Data Pipeline**

At Sama, we have dedicated teams that develop prototypes and tools aimed at achieving these two approaches. We are constantly exploring, prototyping, and productizing tools that combine the best of what state-of-the-art models can do with a deep understanding of where humans still provide the most value during the annotation process.

While we track and strive to improve industry-standard metrics such as data quality and time per task (TPT), we also place particular emphasis on the flow of new and impactful information from humans.

Our goal is to increase this throughput by using every means possible—including ML techniques—to remove anything that stands in the way of capturing that information in the form of best-in-class annotation data.

THE AUTHORS

Claudel Rheault

Human-AI Interaction Lead

With a background in communication studies and interaction design, Claudel Rheault is a user experience research lead focused on human-AI interactions at Sama. Her research areas include human in the loop systems, trust building with AI tools and human-centered product development. Over the past 8 years, Claudel has worked in design agencies, helping clients leverage AI in their digital initiatives, and also in startups helping build ML powered products for finance, cybersecurity and document processing. She is now dedicated to helping on the data side of AI, working in a startup that provides ML engineers with data centric tools to help understand their data better.

Jerome Pasquero

Director, Machine Learning

Jerome Pasquero is the Machine Learning Director at Sama, where he leads the development and implementation of ML models. He earned his Ph.D. in Electrical Engineering from McGill University and has over 15 years of experience in technology, contributing to products that have reached millions of users. Over the past 7 years, Jerome has focused on AI initiatives, helping to expand Sama's product offerings since he joined in March 2021. He is also an inventor with over 120 US patents and has published more than a dozen peer-reviewed papers.