Annotation pipelines in practice

Ground truth for AI evals with Prodigy

Agenda

- Introduction
- Annotation vs Evaluation
- Annotation project design
- Prodigy as an annotation tool
- Annotation tasks: classification, NER, IE, summarization, document partitioning, RAG?
- Prodigy demo with code for each task
- Using Prodigy for model output investigation / insights
- Q & A

RTFM

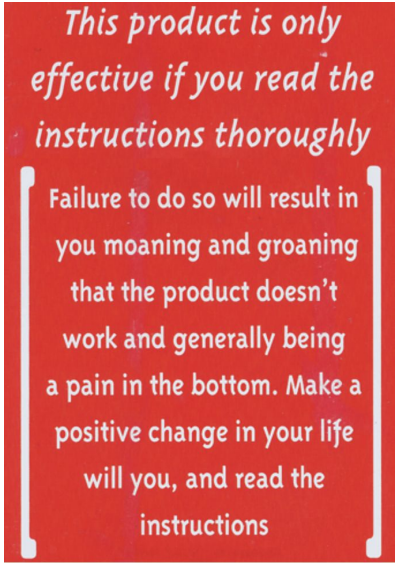
> **RTFM** is an acronym and internet slang for the expression **"Read the f***ing manual"**– typically used to reply to a basic question where the answer is easily found in the documentation, user guide, owner's manual, man page, online help, internet forum, software documentation or FAQ.
>
> – wikipedia.org –

Prodigy has an *excellent* documentation, please take the full advantage of it:

https://prodi.gy/docs

A Unique RTFM Message

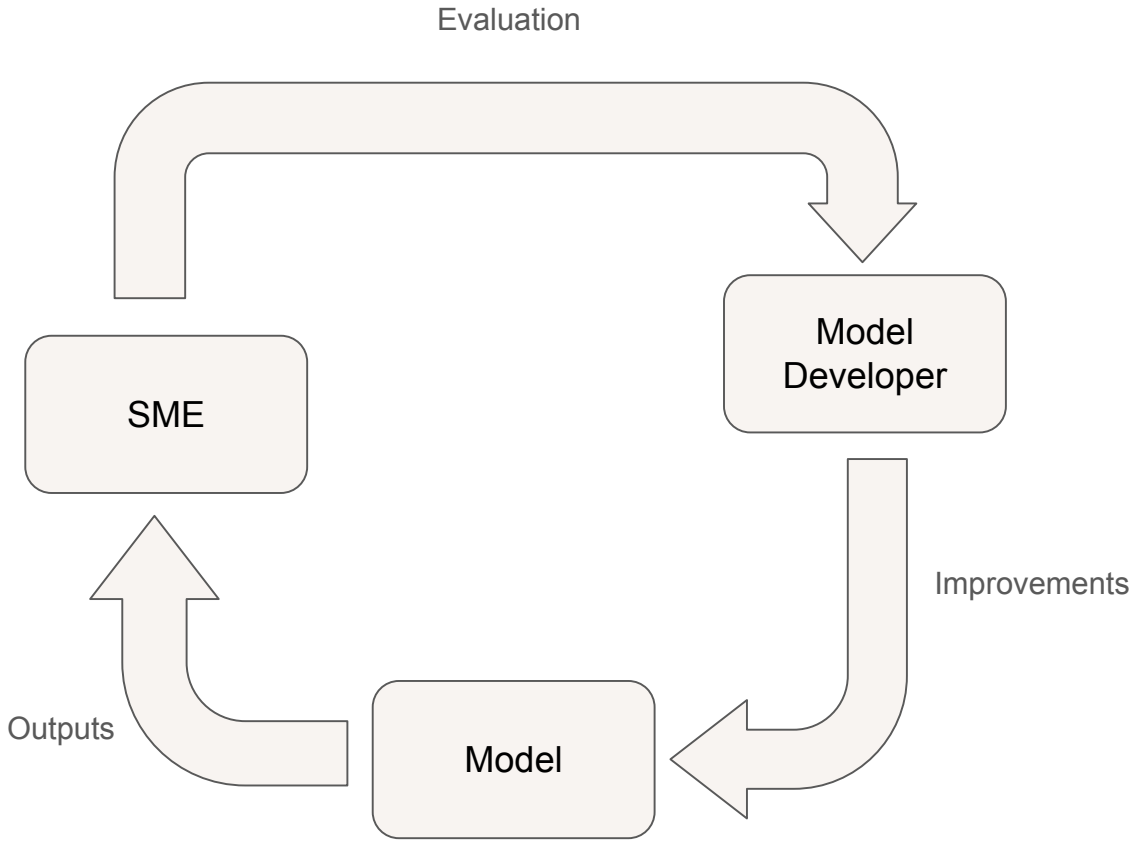
This notice came from the 2012 issue of Maximum PC magazine's guide to building your own PC.



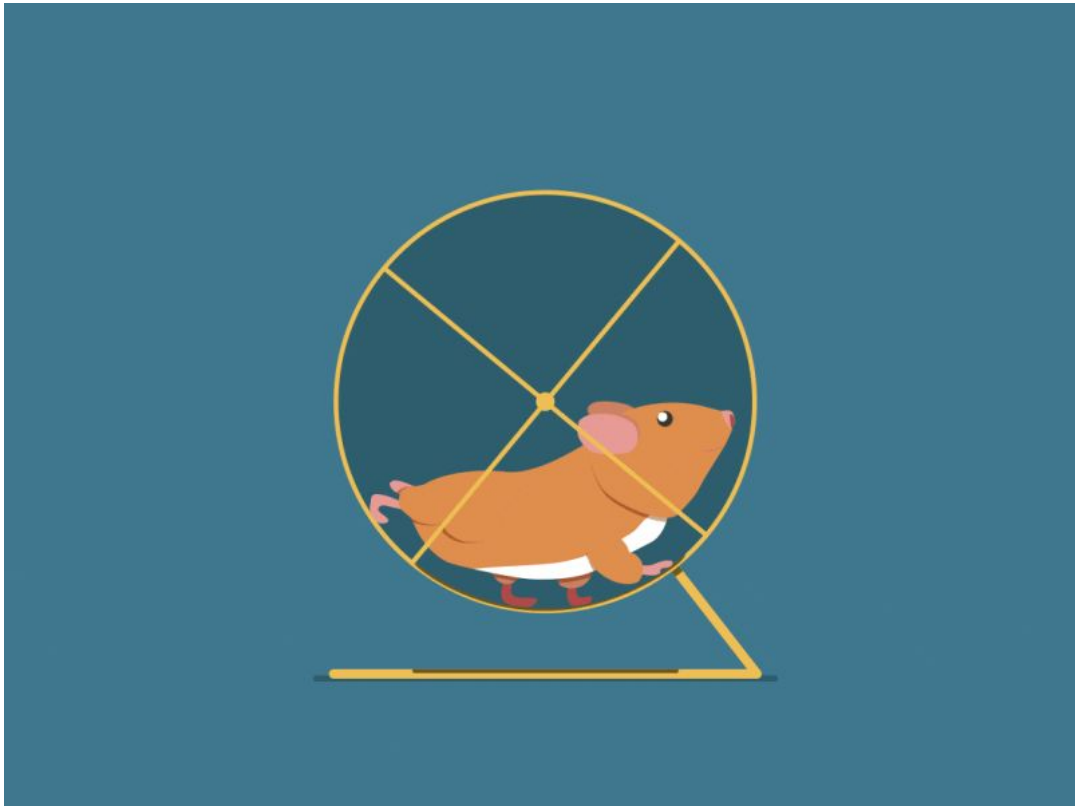
Source: https://www.pcmag.com/encyclopedia/term/rtfm /

Expert-in-the-loop Treadmill

- **Non-scalable:** Each model update demands renewed SME review time
- **No knowledge accumulation:** Expert feedback is not captured as structured data, so the process never compounds.
- **High cost:** SMEs are expensive
- **Delayed iteration:** Models can’t be retrained or deployed quickly because review cycles block releases.
- **Opaque traceability:** Without a persistent dataset, it’s impossible to reproduce prior evaluation outcomes.
- **Burnout risk:** Experts repeat the same reviews indefinitely

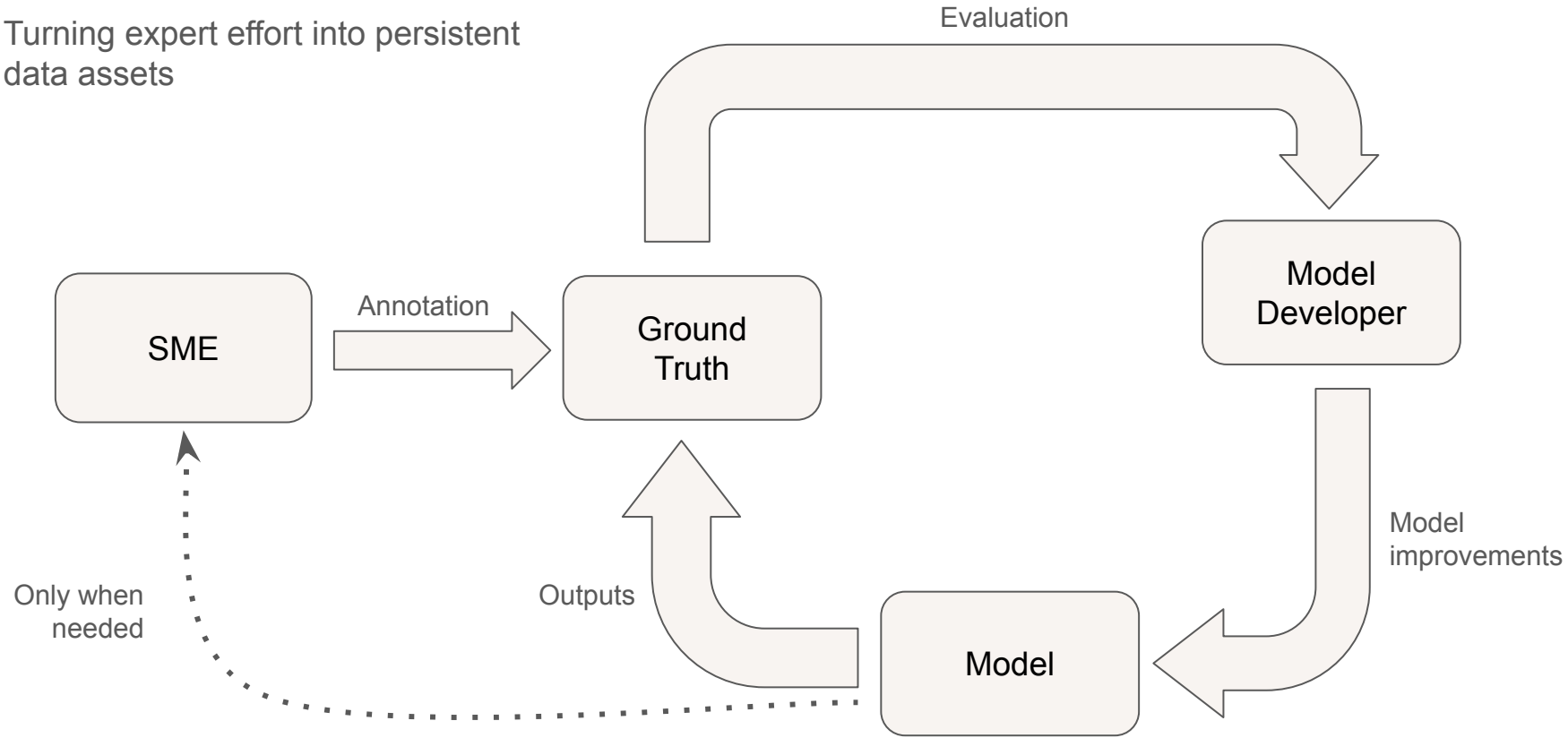


Expert-in-the-loop Treadmill



Continuous Evaluation Loop

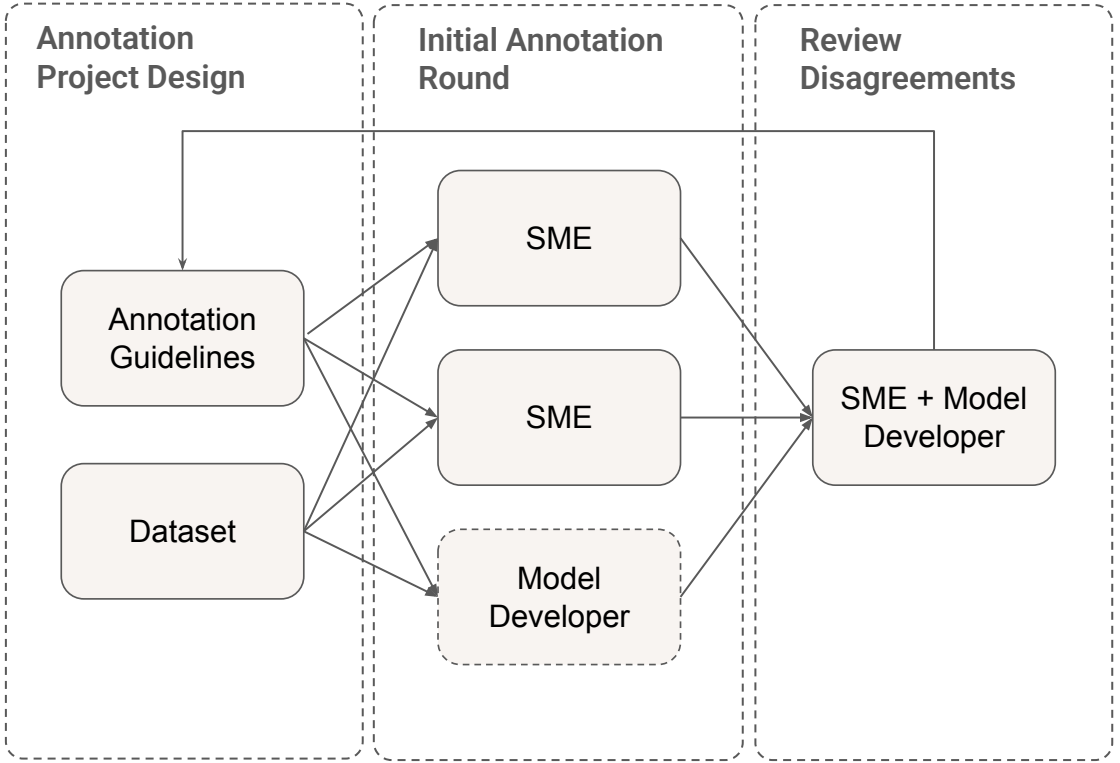
Turning expert effort into persistent data assets



Annotation Project Design

Some design considerations:

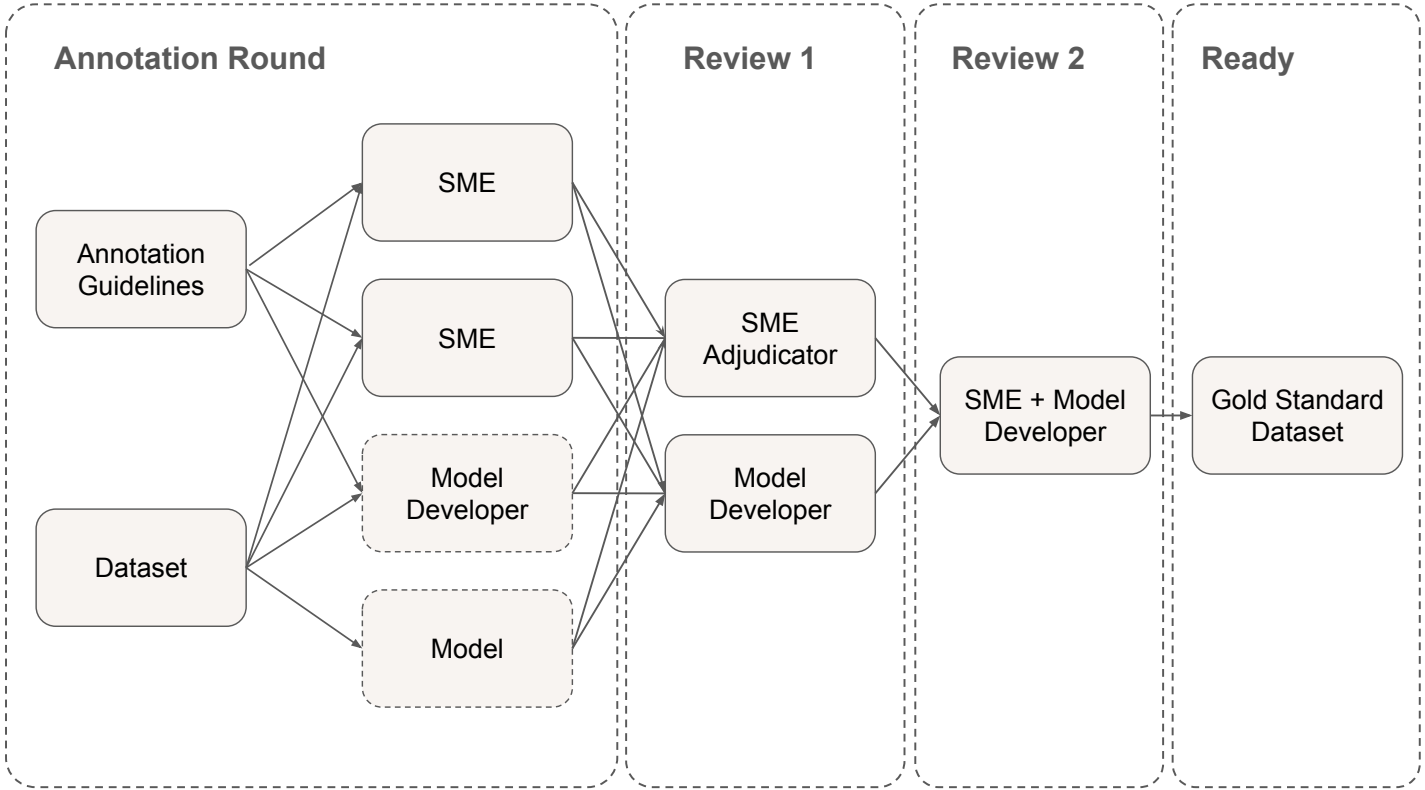
- **Labeling schema:** entities, spans, text classification, binary choice, etc
- **Granularity**: full document vs fragment vs sentence
- **Single-step vs multi-step workflows:** some tasks need two phases (identify spans -> categorize), while others can be single-pass.
- **Pre-labeling strategy:** using model-assisted labeling saves time but risks anchoring bias.
- **Adjudication:** must define how conflicts are resolved — consensus, majority vote, or expert adjudicator.



Annotation Workflow

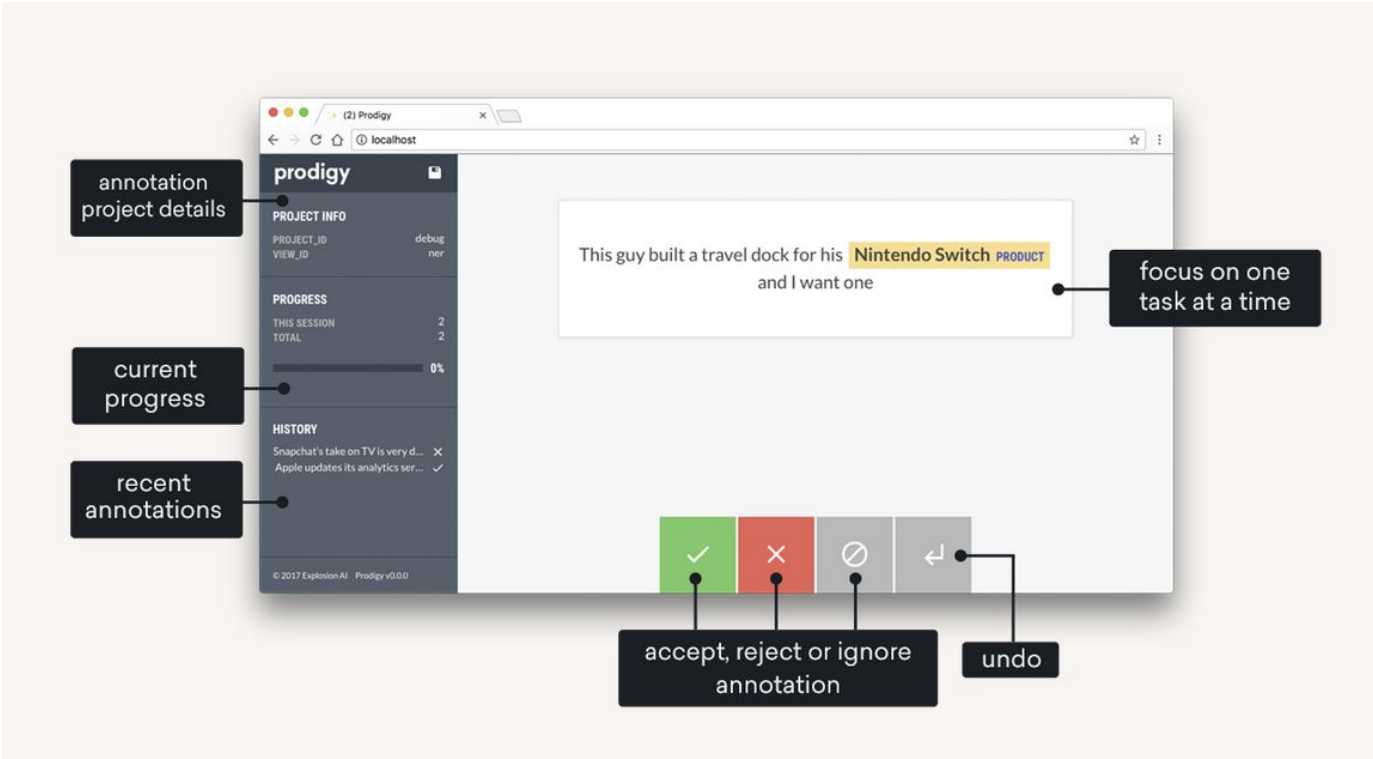
Some design considerations:

- **Model developer as annotator:** quite unconventional, but highly effective practice
- **Model outputs**: if model is mature, can help with discovering annotation errors. Also, may be used for pre-labeling.
- **Double-review**: we observed ~7-8% disagreement rate between annotators. First, SME and Model Developer review disagreements separately, and then review their adjudication differences together.



Prodigy: a modern annotation tool from creators of spaCy

- **Runs locally:** no privacy restrictions
- **Built for customization**: allows to define fully custom data feeds and interfaces
- **Out-of-the-box interfaces**: multiple annotation interfaces are available (e.g. NER, Entity Relations, Classification)
- **Clean, distraction-free interface**: allows to focus on one task at a time
- **Stores data in DB**: supports Sqlite, PostgresQL and MySQL databases



Resources

Prodigy website: https://prodi.gy/

Documentation: https://prodi.gy/docs

Live Demo: https://demo.prodi.gy

Blog: https://explosion.ai/_/topic/llms