* DISCUSSION VIBE
* YOUR CONTRIBUTIONS
GIVE SPACE TO ASK QUESTIONS
* WHITEBOARD
* OUTLINE + ARCH DIAG.

- What is the project about? 2+ years. : Iterate (breakthrough) Tax Assistant / Tax Co-pilot for tax filing
- Significance (Intuit Assist)
  - earnings call
  - media coverage
  - org-wide, 8+ functions, 60+
  - Ties with TT vision & strategic pillar
- My Role
  - Collaborating: Prod. Str. & Roadmap
  - Driving
    - Tech Leadership : Arch + hands-on (Driver from AI), code-review tech. decisions
    - XFN & peers (Designation) & Keystakeholders
    - Mentoring & coaching (Team size leading: 6 AS + 3 MLEs) (Lead of Leads)

Context
Customer Problem
Prior soln (before GenAI era)
Issues with Prior soln
Business Problem: Helpfulness rating 81 → 64 %. conversion 1% ↓ 40 M
Success Criteria: How did you come up with it?

Tech Strategy in step-wise fashion (Onwards)
LLM selection: Trade-offs : Cost vs Latency vs Accuracy (Hallucination) → explain
Tech Levers : Trade-offs i) Prompt ii) RAG iii) Fine-tuning
Architecture Diagram (whiteboard)
Eval as a challenge, Calc as challenge
Success Criteria
North Star: S2C
Sign-Post : Helpfulness >95%, Accuracy >90%, Relevance >90%, Coherence, hallucinat <1%, latency <5sec, cost <acceptable by LT>

Challenges: Problem: demonstrate values
i) Eval → Auto-Eval ii) Calc → Tools
Learnings: i) switching LLMs ii) Planner (80% questions) → impacts knowledge

SFT Deep-Dive
i) Training data (system prompt, user prompt, quality, quantity) ii) Training (Why not full fine-tuning → PEFT, LoRA, Loss, instability, LoRA learning rate multiplier) iii) Eval & iteration
(Software packages) Catastrophic forgetting & solution (LLM-as-a-judge)
Hyperparams → batch size, learning rate multiplier, epochs — error analysis & iteration

Deployment & Inf. optimization (Paged Attention, KV cache, H100 > A100 GPU)

Learnings:
Business Impact Delivered : 21 M users, 0.25% S2C 10 M
Next steps: Deployment with FT model

SFT Deep Dive

- Data
  - quality over quantity
  - quality ← AWS Mech Turk
    - Consistency across 3 experts
  - quantity : few thousand examples
  - Data cleaning : dedupe
  - Data validation :
  - format Prompt : Response
    - (few shot example) sys. prompt ← Instructions 'assistant'
    - context
    - user prompt ← user question
    - .json)

- Training:
  - Full fine-tuning vs PEFT (LoRA, QLoRA)
  - Hyperparams: batch size, learning rate multiplier, epochs (0.1 to 2)

  Issues: Soln
  - Loss instability : learning rate multiplier
  - Catastrophic forgetting
  - Overfitting : • early stopping
    • Data Augment. (another LLM to create syn. data)
  - Out of memory errors : H100 > A100
    faster training
    faster inference
  - Evaluation : • LLM-as-a-judge with Human-in-the-loop
    • Monitoring in Production

- Inference
  - Inference : VLLM (Dynamic batching + Paged Attention)
  - AWS Bedrock, Hosting in Mfoit VPC
  - Services : K8 + Docker + Autoscale + Load balancer.

- Software Packages:
  - Anthropic's API, Bedrock APIs
  - Hugging Face (PEFT)
  - Axolotl
  - Llama-factory
  - Llama Index
  - Haystack

-※ BREADTH & DEPTH

→ BUSINESS PROBLEM & CONTEXT + BUSINESS GOAL

→ YOUR ROLE & CONTRIBUTIONS

→ EXISTING SOLUTION & PROBLEMS IN EXISTING SOLN

→ PROPOSED SOL & ARCHITECTURE

→ CROSS-FUNCTIONAL TEAM COMPOSITION & KEY STAKEHOLDERS

→ STRATEGY & SUCCESS CRITERIA (Sign-Post metrics or Secondary Metrics)

→ TECHNICAL DETAILS
- High-level Components
- Challenges & their sol
- Trade-offs
- Tools & Technologies used

→ BUSINESS IMPACT

→ LEARNINGS

→ FUTURE PLAN

(

LLM PROJECT

COMPANY/TT VISION & STRATEGIC PILLARS : Taxes done for everyone with
1) ease
2) confidence
3) Putting more money into the pockets of our customers

CONTEXT: While filing taxes, a customer has multiple questions
e.g. What is child tax credit? → FAQs
Am I eligible for EITC? → Personalized FAQs
Why is my refund $x? → Outcome Explanation
What is the effect of withholding $x on my taxes? → Tax Advice

CURRENT SOL^{n}: Static content owned by diff teams
e.g. eligibility
outcome
per tax topic

PROBLEMS IN CURRENT SOL^{n}: 1) Static → Lots of maintenance effort
2) Don't have ability to ask follow-up questions
3) Not personalized (some sections)
(e.g. gen. description of who qualifies for EITC but not whether the user qualifies or not)
4) Cannot customize the response for tone, comprehension per user grp

BUSINESS PROBLEM & BUSINESS GOAL : Helpfulness rating 80% → 64% (Δ16%)

↓5% S2C

CROSS-FUNCTIONAL STAKEHOLDERS: Product, Core Engg, Analytics, Legal, Finance, Compliance, Tax Analysts/ Domain Experts, Security, Content Design

TECHNICAL STRATEGY: Step#1: IRS docs : FAQ
step#2: Personalized FAQs
step#3: Personalized Outcome / Calc.

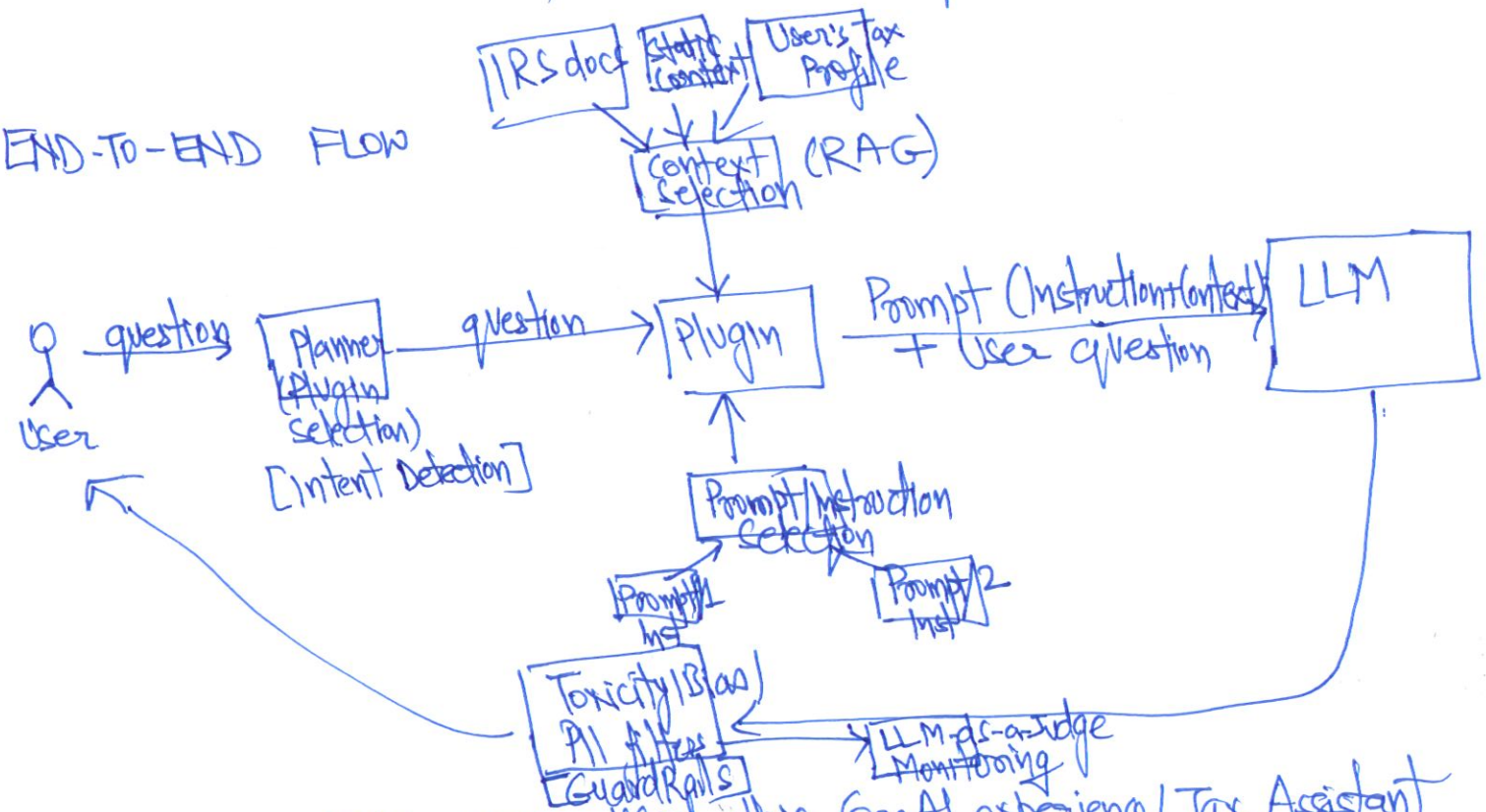
(Issue: LLMs not great at math/calc
Soln: Plugin der

Step#4: Co-ordinating above plugins together in chat via Planner (to switch seamlessly b/w topics)

SUCCESS CRITERIA/ REQUIREMENTS:

1)	Accuracy	>98%. (Compliance business)
2)	Relevance	>90%.
3)	Coherence	>90%.
4)	Hallucination	<1%.
5)	Latency	<1 sec
6)	Cost	<acceptable by SLT>

END-TO-END FLOW



IMPACT: 21M users used this GenAI experience/ Tax Assistant
$4.5M incremental revenue

COMPONENTS DETAILS

1. LLM Selection : Accuracy Vs Cost Vs Latency

Evaluated GPT3.5, GPT4, GCP Palm2, Claude Instant, Claude 2.

Winner: Claude Instant
(Latency < 1 sec
Accuracy ✓
cost ✓)

2. INSTRUCTION/PROMPT :
- Diff. prompt engg. techniques for diff. LLMs e.g. GPT Vs Anthropic
- Zero-shot prompting, few shot prompting Chain of Thought etc.
- Iterative & labor intensive
- Security Testing for prompt injection/ prompt leakage
- Prompt/Inst selection based on user question intent / topic

3. CONTEXT (RAG) AUGMENTATION:
- Augmenting tax specific info
- Latest tax info (LLMs are stuck in time due to their training data cutoff)
- Overcomes context length limits
- Cost effective (due to lower cost of less no. of input tokens)
- Biggest lever before fine-tuning & after instruction/ prompt

Docs / Artifact → Chunking → Generate embeddings for chunks (using embedding model)

↓

Store in Vector DB

↓

Create indexes (optional) (Llama Index)

User Query → Generate embeddings using embedding model

↓

Semantic search over chunks in Vector DB (ANN or using indexes)

↓

Fetch relevant chunks, attach it to inst/prompt + user question (Instruction + "Relevant context" + Query)

↓

LLM

4. EVALUATION:

0) Human Evaluators

1) Syntax related metrics e.g. BLEU, ROUGE → ✗ Human evaluation

2) Semantic metrics b/w user question+context & response (cosine similarity b/w embeddings) → ✗ Human eval.

3) Using syntax & semantic metrics as features & evaluation verdict (by humans), build model → ✗

Automation

NLP is ~~TASK-SPECIFIC~~ ANOTHER LLM as a JUDGE

4) i) Regression: "Expected Response" known

Using (i) Checklist (ii) steps followed by top experts (iii) expected response, code a regression prompt for another powerful LLM as a judge (iv) few-shot examples

ii) Online: "Expected Response" not known

Check: generated response bounded by "context" using another LLM as a judge

5. FINE TUNING: SFT / RLHF

SFT:
① – Cannot fine-tune on user data → compliance reasons
② – Fine-tuned open source & models available on AWS Bedrock e.g. Falcon using synthetic data but it did not meet accuracy bar (>98% tax accuracy)

RLHF :- Partnering with Anthropic (specialist) (providing data & assisting in crowdsourcing human evaluation)
– Can describe about Reward Modeling Issue (ordering responses → Many human labellers needed)

CHALLENGES: 1) Context length limit (initially) : RAG

2) Switch ~~from~~ GPT to Claude Instant
- prompt engg. challenges

3) Automated Eval.

4) Planner (OOS questions)

5) Calc. → plugin

TRADEOFFS: 1) Latency+Cost Vs. Accuracy : (LLM selection)
Claude Instant accuracy lower than GPT4

2) ~~Prompt~~ RAG Vs fine-tuning : Initial phase RAG ✓ (tax law can chang
Later: Combo of RAG + Fine-tun

WHAT COULD HAVE BEEN DONE DIFFERENT? / LEARNINGS

1) LLM to be fixed first (prompt, context, eval. tightly coupled with LLM)

2) ~~Tax~~ Accuracy definition adapted to LLM world
3) Very hard to get tax accuracy > 90% (since we are in compliance business)
4) Automated Evals almost a necessity for faster iterations

YOUR CONTRIBUTIONS: 1) Strategy & Roadmaps

2) E2E system Design / Architecture (Genos as well

3) ~~Implementation~~ : e.g. RAG using llama-index, Prompt engg, Regression ~~tuning~~ online eval

4) Leading E2E development + deployment + integration with other subsystems (cross-collab

PROMPT Vs FINE-TUNING

	PROMPT	FINE-TUNING
PROS:	→ No data to get started → Smaller upfront cost	→ Learn new info → Reduce hallucination → Lower cost per request (after fine-tuning)
CONS:	→ Hallucinations → Less consistent	→ More high quality data requirement → Upfront compute cost

WHAT FINE-TUNING CAN DO? — Behavior Change / Improve consistency / Style, Tone, format
— New Knowledge

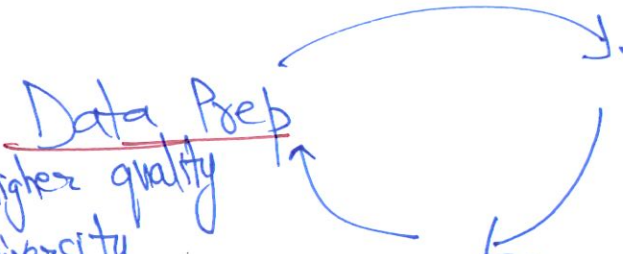
⇓

General Purpose LLM → Specialized LLM

INSTRUCTION FINE TUNING: Data Prep: Instruction – Response Pair (optional I/P)

e.g Ins: Add 2 numbers
Input: 2, 7
Response: 9

FINE-TUNING



Data Prep

1) Higher quality
2) Diversity
3) Real (not synthetic)

Training

1) Collect Prompt-Response Pairs
2) Tokenize with the same tokenizer used in pre-training (+ padding/Truncation)
3) Train-Test split
4) Hyperparams: epochs, learning rate

Evaluation

1. Evaluation Metric (Task specific)
2. Error-Analysis (categorize errors, augment samples to rectify those errors and re-train)
3. Vary amt of data & measure performance

PEFT → LoRA

(Performance-Efficient fine-tuning) (Low-Rank Adaptation)

→ Fewer trainable params (i.e. less memory requirement)
→ Slightly lower accuracy than full fine-tuning
→ Same inference latency as full fine-tuning

→ Train new wts in some layers, freeze main wts
And at inference, merge with main wts
(These new wts are rank decomposition matrices of original wts change)

→ Issue in Sequential fine-tuning: Catastrophic Forgetting

DIRECT PREFERENCE OPTIMIZATION

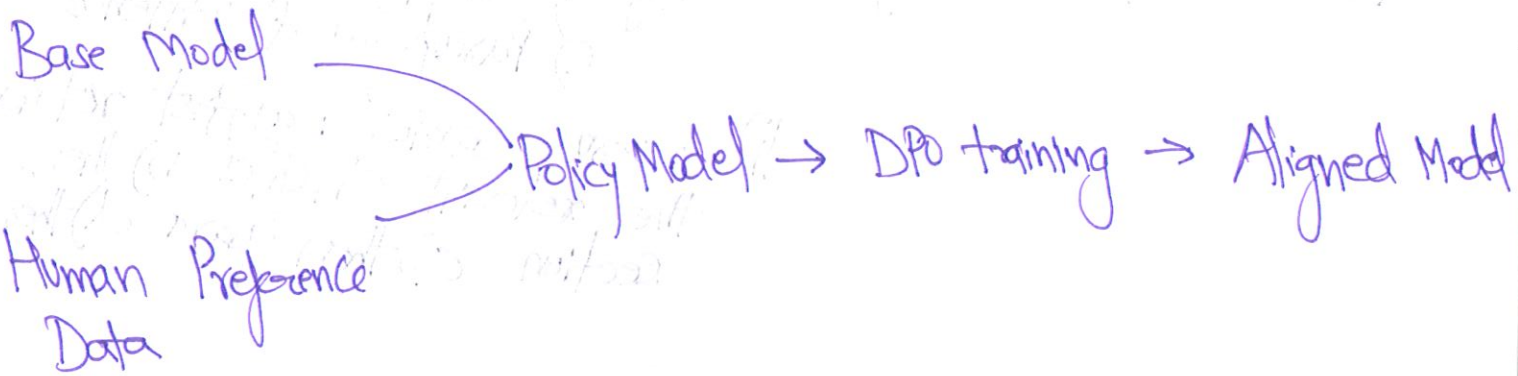
→ DPO is an alternative to RLHF

→ Training data has good & bad responses (preference pairs) for the same prompt

→ DPO reformulates the RLHF objective into a classification problem

→ DPO makes the model learn to maximize the probability gap b/w preferred & non-preferred responses (e.g. cross-entropy loss) while the regularization term prevents it from deviating too much from the base model

→ DPO implicitly learns a reward function within its training process



CHUNKING STRATEGIES IN RAG

- Fixed length : Cons → Same topic content can be separated in diff. chunks (ideally should be in one chunk)
- Sliding window / overlapping window : Tries to solve above issue
- Paragraph / section based
- Semantic chunking : i) Divide into sentences / paragraph / section
  ii) Find similarity based on embedding of these sent. / para / section to all other sent. / para / section and iii) cluster them together and label the cluster ~~with~~ the topic of the cluster
- Hierarchical chunking : i) Multiple ~~levels~~ of chunking
  e.g. a) Chapter b) section
  c) Paragraph d) sentence
  ii) Query comes : a) first retrieve the relevant chapter b) then section c) then para d) then sentence