aveni

FinLLM Case Study

Robust LLM Evaluations for Financial Fact Finds

aveni

Table of contents

Introduction

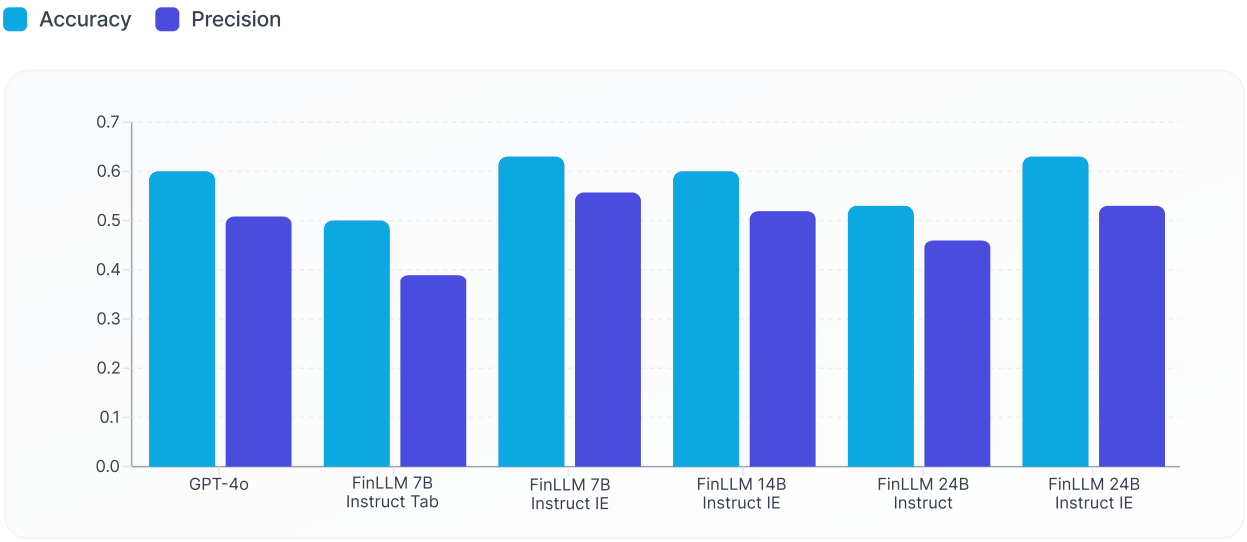
Overview

Financial advice lives or dies on the accuracy of what gets captured from client conversations. Advisers discuss incomes, assets, liabilities, and spending in noisy, real-world dialogue, and those details must be recorded correctly for both regulatory compliance and good advice. Aveni Assist Fact Find uses an external, API-based LLM to turn call transcripts into structured tables. But confidence in those tables still depended on manual checking, which is slow, expensive, and hard to scale, especially in a domain where even small hallucinations can have consequences.

This case study focuses on a shift in how we validate and improve LLM performance for financial workflows: building robust, automated evaluations that measure real production accuracy and directly surface hallucination risk. By moving from ad-hoc human QA to repeatable, metrics-driven evaluation, we can track performance over time, compare models quickly, and detect failure modes before they affect advisers or clients. Just as importantly, we can quantify whether extracted facts are genuinely grounded in the underlying summary and transcript, rather than being plausible but invented.

Our results show that FinLLM, fine-tuned with synthetic Assist data and finance-specific information extraction datasets, performs strongly on this task and often surpasses GPT-4o at the field level. Fine-tuning not only improved raw extraction accuracy but also reduced formatting errors and made model behaviour more consistent, which is critical for downstream automation. The automated evaluation framework now lets us iterate rapidly, while a structured manual evaluation provides a final safeguard against judge bias and alignment artefacts. Together, these evaluation layers give us a clearer path to deploying models that are not just more accurate, but more trustworthy: models that minimise hallucination and can be relied on in real financial advice workflows.

Fact Find Overall Evaluation Results



Existing Set-up

The pipeline currently used in production for Assist Fact Find consists of transcribing the call, constructing a narrative and summary using an external LLM, extracting the hard facts and verifying with a human for accuracy.

Fact extraction entities

The following facts are automatically extracted from the LLM-generated summary for each of the following categories: Income, Expenditure, Asset. These facts are extracted as a table, where each item is listed as a separate row.

For example, given the following summary:

Income

Joe Bloggs has a yearly income of £50,450 from his job as teacher at Sanderson High. He also receives a passive income from his flat in London that he rents out for £2,100 per month.

Expenditures

Joe spends £1,500 per month on the mortgage for his flat in Brighton and pays £200 a month to run his car, a Ford Fiesta.

The table with gold values may look like:

PERSON	ITEM	CATEGORY	VALUE	FREQUENCY	NOTES
Joe Bloggs	Salary	Income	£50,450	Annually	Job as a teacher at Sanderson High
Joe Bloggs	Flat	Income	£2,100	Monthly	Flat in London
Joe Bloggs	Mortgage	Expenditure	£1,500	Monthly	Flat in Brighton
Joe Bloggs	Expenses - car	Expenditure	£200	Monthly	Cost of running car, Ford Fiesta

The key challenges with this approach:

- Manual QA was too slow, expensive, and inconsistent to support rapid model iteration or ongoing production monitoring.
- External LLM usage also drove high inference costs due to multi-step API calls.
- We needed a reliable automated evaluation framework to test FactFind accuracy repeatedly without human bottlenecks.

FinLLM Approach

We're replacing the external FactFind LLM with a FinLLM model tuned for Assist, and the key enabler is a stronger evaluation stack. We fine-tune using task-aligned information extraction and instruction following data, boosted by synthetic Fact Find examples, then validate performance in two stages:

1. **Automated evaluation with an LLM-judge** to rapidly compare many models
2. **Structured manual evaluation** of the top FinLLM candidates against GPT-4o to confirm results and ensure production-level trust.

Fine-tuning & Synthetic Data

Our synthetic data creation and fine-tuning approach in brief:

Synthetic data generation

We first create realistic client/adviser "characters" (names, jobs, family status, etc.). Then a prompt generates both (1) a markdown FactFind-style table of incomes/expenditures/assets and (2) a matching Assist-style summary grounded in that table.

Fine-tuning data mix

FinLLM is trained on a blend of:

- Synthetic Assist FactFind examples (core task training)
- Synthetic Detect examples (similar IE task to improve generalisation)
- General Information Extraction datasets
- Instruction-following data to preserve correct table formatting
- Broader finance datasets (QA, summarisation, numerical/tabular reasoning)

Evaluation

Automated and manual evaluation play different roles in the validation stack. Automated metrics let us benchmark many models quickly, but they depend on two imperfect components: (1) row alignment, and (2) LLM-as-a-judge correctness calls. Because those steps can introduce noise, automated scores alone aren't sufficient to sign off production readiness [1].

Manual evaluation removes these sources of error and gives a higher-fidelity estimate of real-world performance, but it's expensive and slow, so we reserve it for only the strongest candidates. In practice, automated evaluation acts as a screening/triage layer, and manual evaluation as the final verification layer.

[1] For example where the judge marks empty fields as "correct" when the gold data contains a value for the field, or where the judge is overly generous in terms of assessing the equivalence of numerical values

Automated Evaluations

Table Row Alignment

Automated evaluation for Assist Fact Find is designed to be robust to real-world LLM behaviour and to directly measure trustworthiness. Because fact extraction is an Information Extraction task, we report standard metrics (precision, recall, F1), but replace the usual string matching with an LLM-judge so semantically equivalent answers (e.g., “yearly” vs “annually”) are scored correctly. Since models often produce different numbers of rows through duplication, splitting, or hallucination, we first align model rows to gold rows using embedding similarity and cosine matching, This enables fair comparison even with over/under-generation.

LLM Judge

Once aligned, the LLM-judge scores each field in each row as correct or incorrect. Scoring at field level gives a clearer picture of where models succeed or fail, rather than hiding issues inside a single aggregate score. We report accuracy/precision per field and per category to support fast model iteration.

Measuring Hallucinations

Because hallucinations are especially risky in finance, we evaluate grounding separately from CRM accuracy. CRM tables often include details advisers added later from other sources, so they aren’t a perfect reference for hallucination. Instead, we compare extracted rows back to the generated summary and classify them as Grounded (fully supported), Partial (some support), or Ungrounded (mostly unsupported). We also run this grounding check on gold CRM rows to estimate the practical upper bound on extraction performance from summaries alone.

Together these pieces, semantic row alignment, field-level LLM judging, and grounding analysis, give us a robust and repeatable evaluation stack that captures both extraction accuracy and hallucination risk in conditions that mirror production use.

Manual Evaluations

We ran a structured manual evaluation comparing GPT-4o with three fine-tuned FinLLM variants (24B M quantised, 14B M, 7B Q). Annotators were shown the GPT-4o summary, a single extracted row from a model output table, and the full gold CRM table. They label row correctness to quantify over/under-generation (row-level precision/recall) and score individual fields for field-level precision/recall.

This setup gives a clean, model-agnostic view of whether FinLLM improves on the production baseline and where remaining failure modes sit.

Performance

The table with gold values may look like:

Table 1: Fact Find Overall evaluation results

Model	Acc	Pr	Rc	F1	Over Gen (%)[2]	Markdown Issues[3]
GPT-4o	0.6025	0.5102	0.9971	0.6728	-17.13	0
FinLLM-7B-Instruct-Tab-Q	0.4986	0.3932	0.9945	0.5610	+69.83	58
FinLLM-24B-Instruct-M	0.5367	0.4762	0.9099	0.6248	-27.19	2
FinLLM-24B-Instruct-IE-M	0.6251	0.5437	0.9874	0.6996	-20.86	5
FinLLM-14B-Instruct-IE-M [4]	0.6084	0.5178	0.9866	0.6777	-20.11	3
FinLLM-7B-Instruct-IE-Q [5]	0.6371	0.5519	0.9878	0.7060	-22.16	2

The results in Table 1 support the replacement of GPT-4o with a FinLLM model. Although these results suggest that the Q model would be the best choice, the numbers in this table are averages that obscure the full picture. When drilling down to the category and field levels (Table 2), FinLLM-24B-Instruct-IE-M model outperforms GPT-4o on almost every field, for each of the three categories, highlighting its suitability for use in production.

Extracting values is still challenging with the current approaches, e.g. we observed particularly low scores for the “value” field in the Income and Expenditure tables for all models. We found that this is due to the value associated with an item often not being mentioned in the summary text, highlighting a key limitation of evaluating against the gold data from the CRM.

...

[2] Over/Under generation: all of the LLMs typically generate tables with more or fewer rows than the gold table contains. This column captures how much the models “over or under generate” table rows. Values are negative when the model generates fewer rows than the number of rows in the gold table.

[3] Markdown issues: the number of summaries for which the models output a badly-formatted markdown table (out of 92 summaries). In these cases the information extracted may be (partially) correct, but the output is not usable downstream due to formatting issues.

[4] (Including Information Extraction and Synthetic Data)

[5] (Including Information Extraction and Synthetic Data)

Table 2: Fact Find Fine-grained evaluation results for

FinLLM-24B-Instruct-IE-M (Including IE and Synthetic Data) vs. GPT-4o

Category	Field	GPT-4o				FinLLM 24B Instruct IE M			
		Acc	Pr	Rc	F1	Acc	Pr	Rc	F1
Income	Person	0.7629	0.7629	1.0000	0.8655	0.8533	0.8525	1.0000	0.9204
	Item	0.4639	0.4639	1.0000	0.6338	0.5054	0.5054	1.0000	0.6715
	Value	0.2113	0.2113	1.0000	0.3489	0.1902	0.1902	1.0000	0.3196
	Frequency	0.6598	0.6598	1.0000	0.7950	0.6685	0.6721	0.9919	0.8013
	Notes	0.6443	0.0000	0.0000	0.0000	0.8533	0.0000	0.0000	0.0000
	All fields (combined)	0.5485	0.4817	1.0000	0.6502	0.6141	0.5348	0.9975	0.6963
Expenditure	Person	0.7271	0.7219	1.0000	0.8385	0.8330	0.8294	1.0000	0.9068
	Item	0.1751	0.1751	1.0000	0.2980	0.2129	0.2147	0.9623	0.3511
	Value	0.1545	0.1545	1.0000	0.2676	0.1357	0.1357	1.0000	0.2390
	Frequency	0.8435	0.8576	0.9808	0.9151	0.8633	0.8733	0.9869	0.9266
	Notes	0.7796	0.0000	0.0000	0.0000	0.8027	0.0000	0.0000	0.0000
	All fields (combined)	0.5359	0.4496	0.9913	0.6186	0.5695	0.4869	0.9903	0.6529
Asset	Person	0.7242	0.7235	1.0000	0.8395	0.7735	0.7722	1.0000	0.8715
	Item	0.5850	0.5850	1.0000	0.7381	0.7294	0.7294	1.0000	0.8435
	Value	0.5599	0.5599	1.0000	0.7179	0.5618	0.5618	1.0000	0.7194
	Frequency	0.9805	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
	Notes	0.7660	0.4595	1.0000	0.6296	0.3941	0.1317	0.5366	0.2115
	All fields (combined)	0.7231	0.5995	1.0000	0.7496	0.6918	0.6093	0.9744	0.7497

Hallucination analysis

Results for Grounded, Partial and a sum of both categories are shown in Table 3. We report only the results for GPT-4o, FinLLM-24B-Instruct-IE-M, and FinLLM-7B-Instruct-IE-Q as well as the gold data.

The hallucination evaluation showed that several “gold” rows did not originate from the summaries themselves.

For instance, the maximum proportion of grounded rows was only 0.2070 for Asset, suggesting that many such rows either came from the transcripts or from external sources. Among the models, GPT-4o achieved the strongest performance in identifying grounded rows and, although FinLLM-7B-Instruct-IE-Q obtained higher scores in cell-based evaluation (Tables 1 and 2), its advantage diminished when assessed at the row level.

When grounded and partially grounded rows were considered jointly, FinLLM-24B-Instruct-IE-M demonstrated a slight improvement over GPT-4o, indicating a sensitivity to partial alignment.

Table 3: Hallucination Results

Category	Grounded	Partial	Sum
Gold			
Income	0.1217	0.6825	0.8042
Expenditure	0.0763	0.4140	0.4903
Asset	0.2070	0.6531	0.8601
FinLLM-24B-Instruct-IE-M			
Income	0.5158	0.4325	0.9481
Expenditure	0.5390	0.2846	0.8237
Asset	0.5123	0.3868	0.8992
FinLLM-7B-Instruct-IE-Q			
Income	0.4073	0.4764	0.8836
Expenditure	0.4081	0.3270	0.7351
Asset	0.4592	0.4393	0.8985
GPT-4o			
Income	0.6570	0.2852	0.9422
Expenditure	0.6529	0.2112	0.8641
Asset	0.5685	0.2944	0.8630

Key Takeaways

We set out to demonstrate how through specialised evaluation and fine-tuning, FinLLM can compete and outperform leading external API-based LLMs on a core financial task. We validated this through real-world application in Aveni Assist Fact Find calls, and streamlined the evaluation process for a more robust and reliable validation.

1. FinLLM outperforms GPT-4o for the majority of fact find fields thanks to fine-tuning on specific synthetic data and finance-specific information extraction data. This improved performance on the FactFind task should reduce the Advisor effort involved in fact verification

2. We developed an automated evaluation method to rapidly evaluate the performance of a diverse set of models. The manual evaluation can now be reserved to a small set of models to promote confidence regarding their expected performance in production.

3. Successful correction of unreliable markdown table output and overgeneration of table rows by FinLLM-7B-Instruct-Tab-Q models

4. Investigated reasons for continued difficulty in extracting values due to the value associated with an item often not being mentioned in the summary text, highlighting a key limitation of evaluating against the gold data from the CRM.

aveni

Get started with Aveni Assist today

aveni.ai/aveni-assist